Machine Learning Methods for Robust Quantum Optimal Control

Muhammad Irtaza Khalid

School of Computer Science and Informatics

Cardiff University

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

October 2023

Abstract

Quantum technologies have the potential to revolutionize many classical tasks, particularly including sensing and simulation applications. Yet their full potential is limited by the presence of noise, amongst other issues. This thesis addresses the problem of quantum optimal control of a controllable system with noisy dynamics and an uncertain theoretical description.

Towards this goal, this thesis makes two contributions. Firstly, it develops a novel robustness measure called the Robustness Infidelity Measure (RIM) for certification of robustness of optimal control schemes, agnostic of the acquisition method. The RIM is a statistical measure and it can be used to compare the robustness of different schemes. Secondly, this thesis develops novel optimization techniques based on Reinforcement Learning (RL) for robust optimization of noisy quantum dynamics with model uncertainties. In particular, a model-based RL algorithm is proposed that is able to improve over direct applications of model-free RL algorithms in terms of experimental resource consumption. This is done via incorporation of partial knowledge of the uncertain model whilst the rest is learned using experimental data. Our approach highlights the potential of extending pure model-free methods towards model-based approaches, with a learnable model, for noisy optimization tasks and brings RL algorithms closer to deployment on near-term quantum devices. We evaluate the RIM and various model-free RL algorithms on a number of benchmark problems. Our results show that the RIM is a valuable tool for assessing the robustness of quantum control schemes. Moreover, we demonstrate that RL algorithms are able to generate robust control schemes which outperform schemes generated using other methods. We also show how learned models of noisy quantum dynamics can be leveraged to increase the optimality of quantum control schemes found by RL algorithms whilst retaining their robustness performance.

Contents

Originality

Statement

The writing in this thesis is my original work except where relevant others' work is included for reference and appropriately acknowledged.

Papers

The material in this thesis has been published in the following papers:

Reinforcement Learning vs. Gradient-Based Optimisation for Robust Energy Landscape Control of Spin-1/2 Quantum Networks
Irtaza Khalid, Carrie A. Weidner, Edmond A. Jonckheere, Sophie G. Shermer, Frank C. Langbein
IEEE Conference on Decision and Control, pp. 4133-4139, 2021

Statistically Characterising Robustness and Fidelity of Quantum Controls and Quantum Control Algorithms
Irtaza Khalid, Carrie A. Weidner, Edmond A. Jonckheere, Sophie G. Shermer, Frank C. Langbein
Phys. Rev. A 107, 032606, 2023

Analyzing and Unifying Robustness Measures for Excitation Transfer Control in Spin Networks
Irtaza Khalid*, Sean P. O'Neil*, Carrie A. Weidner, Frank C. Langbein, Sophie G. Shermer, Edmond A. Jonckheere
IEEE Control Systems Letters, vol 7, pp. 1783-1788, 2023

Sample-efficient Model-based Reinforcement Learning for Quantum Control
Irtaza Khalid, Carrie A. Weidner, Edmond A. Jonckheere, Sophie G. Shermer, Frank C. Langbein
arxiv:2304.09718, accepted by Phys. Rev. Research, 2023

* denotes equal contribution

Open source software

A number of software projects were developed during the course of my doctoral journey. Some of the important ones include:

robchar
An algorithmic framework for ROBustness CHARacterization of quantum control and control algorithms
`https://github.com/qyber-black/code-robchar`
DOI:`10.5281/zenodo.6891381`

transmon
The algorithm for the Learnable Hamiltonian model-based Soft Actor Critic for closed and open quantum gate control
`https://github.com/erg0dic/transmon_public`

For my parents, Nabeela and Khalid,

Chapter 1

Introduction

1.1 Motivation

Quantum technologies and computing hold immense promise for delivering on many scientific and technological fronts. There has been steady progress over the past years on many fruitful frontiers including more accurate sensing and imaging devices in the field of quantum metrology [GLM11], higher precision atomic clocks [Lud+15], and secure communication networks powered by quantum key distribution protocols [Wan+19; SP00].

Recently, the milestone of a modest experimentally scalable quantum error correcting surface code was achieved [Ach+23] – only a few years after the first quantum advantage milestone involving Gaussian Boson Sampling [Aru+19]. The path to fully fault tolerant quantum computers is expected to be long [Pre18] but the promise of practical quantum advantage over existing classical computation based approaches in a variety of problems like learning from quantum physical experiments [Hua+22b], prime number factorization [Sho94], optimization [FGG14] and quantum simulation [Fey18] beckons us.

We are currently in the Noisy Intermediate Scale Quantum (NISQ) era [Pre18] of quantum computing. Current NISQ quantum devices, though being readily accessible on the cloud [Ani+21], are limited and prone to significant sources of errors which limits the capabilities of quantum technologies especially in regard to their computational applications [Che+22; Lyk+22]. Unlocking the full potential of quantum computers in terms of their computational complexity will require the design and manipulation of fully fault tolerant logical qubits, comprised of many physical qubits, using techniques from quantum error correction (QEC) [Got97]. Being similar to classical error correction, these techniques essentially encode all possible errors into redundancies, per qubit of information, to allow detection and correction. QEC is a significant challenge and will likely take many years. Moreover, the path to QEC, e.g. the milestone of creating the first logical qubit, will require lower operational errors of standard operations in quantum computation to reduce the size of the QEC codes needed for fault-tolerance.

Quantum errors or noise can roughly be characterized into three forms:

1. **Environmental noise:** interactions with the environment that lead to non-unitary dynamics, further categorizable into:
   - **Markovian noise:** environmental noise that is memoryless and can be described by a Lindblad master equation;
   - **Non-Markovian noise:** environmental noise that has memory and needs to be accounted for by something more general;
2. **Parametric noise:** inaccuracies in the control model and/or signal representing a specific physical implementation which leads to unitary evolution errors, which are:
   - **Time-dependent changes:** model parameters fluctuate or drift as a function of time
   - **Static changes:** model parameters are fixed in time but are not known exactly
3. **External noise:** measurement errors as a result of finite sampling statistics, due to interaction with an external probe, a manifestation of the probabilistic nature of observables.

Broadly speaking, three ways have been proposed to deal with errors in order to realize various quantum technologies including fault-tolerance:

1. via QEC protocols [CS96; Ste96a; Ste96b; Got09];
2. using error mitigation schemes, e.g. reversing noisy dynamics [BK02; TBG17; Bea+18; EBL18; LB17; Cet+20; Ega21], active variational noise minimization [LB17], or parametric modelling of architecture defects in trapped qubits [Cet+20; Ega21];

3. Robust Quantum Optimal Control (RQOC) engineering, e.g. landscape shaping of the quantum control optimization problem in search of noise-free regions [Hoc+14; VKL99; SZZ04], decoherence-free subspaces [LCW98; JSR14], or noise spectral density based filter functions [Kab+14; Gre+13].

We focus on the challenges and opportunities presented by the relatively nascent field of RQOC in this thesis. In a nutshell, RQOC is concerned with the design and development of controls necessary to realize a broad range of quantum dynamics accurately with emphasis on the controls being immune to variations and fluctuations introduced by various noise sources in the physical environment. We can approach the RQOC control problem in two ways: (1) leverage our knowledge of the nature of noise directly in the design of robust controllers (2) obtain controllers in ideal settings and then study their performance under experimental/theoretical noise to post-select a robust controller. Both methods are explored in this thesis.

Typically, standard Quantum Optimal Control (QOC) protocols assume that a theoretical model of the quantum system is available e.g. via spectroscopic characterization and focus solely on optimizing quantum dynamics in ideal no-noise conditions [Kha+05a; RNK12; Mac+11a; Koc+22]. Thus, these *model-based* methods[1] which use an analytical model can be employed and have have been the focus of over half a century of fruitful contribution to QOC, including algorithms such as GRAPE [Kha+05b] and Krotov [RNK12] which utilise gradient-based optimisation of a model-based target functional.

As mentioned before, noise unsurprisingly reduces the performance of standard control protocols on NISQ devices. Also, theoretical models of quantum devices are not always available at large scales and are also expensive to obtain and/or verify [EHF19]. This means that errors in the model are more likely to arise which also reduce the performance of model-based control protocols. RQOC attempts to address these problems faced by standard QOC where the final goal is to increase the reliability of control protocols *in the wild*, in experiments, outside the theoretical/computational settings where they are created. For example, a model-based RQOC method is non-adiabatic inverse eigenvalue engineering that derives time-dependent control Hamiltonians from an undetermined propagator that generates a transitionless evolution of an eigenstate of a dynamical invariant of the Schrödinger equation. Protocols for

[1] some based on Pontyragin's maximum principle [Pon87] for optimal control amongst many others

robust state preparation using this approach where the invariant is a pure generally parametrized density matrix have been proposed for closed and open systems for some theoretical noise models such as decoherence and amplitude or systematic errors in the Hamiltonian [Jin+13; Rus+12] as well robust gate control [DLG20].

As an alternative to model-based control, model-free RQOC involves looking at optimal control in the absence of a model i.e. the *model-free* setting. The modern interpretation of model-free control is usually synonymous with machine learning or black-box optimization for control, particularly Reinforcement Learning (RL) for optimal control [Ber19]. However, the idea is not new and goes back to dual control theory initiated by A.A. Feldbaum in the 1960s [Fel60]. Both coalesce the control problem to approximate dynamic programming solved using Bellman's principle of optimality [Bel52]. Solvers follow the principle of initially exploring and learning the unknown model by probing the system, and, later, exploiting this information for control. Initially the control actions taken by the controlling agent are sub-optimal as it works with a highly uncertain model although they can still be seen as optimal in the sense of solving the Bellman equation step-wise based on the acquired information. Iterated composition of the solutions achieves near optimal solutions, eventually. Prior work has demonstrated the usefulness of RL for quantum optimal control [Che+13] in its application to synthesis of transmon gates [Dal+20a], coherent transport by adiabatic passage through semi-conductor quantum dots [Por+19], and robust two-qubit gmon gate synthesis [Niu+19a].

Finally, we note that the theoretical potential, existence and cost of robust control solutions is unclear for the general case [Koc+22]. This is partly linked to the absence of general analytical solutions to all types of quantum dynamics governed by the Schrödinger equation [Ahm+18]. The questions underlying existence and cost of robustness are explored numerically and analytically in the literature. For example, for state preparation, robustness is akin to ensemble control with Bloch vector control equations. Here, the algebra of the polynomials of the non-commuting vector fields that generate the system dynamics provide requirements for controllability of the ensemble system and therefore robustness [LK09]. In this thesis, we explore why certain algorithms can create robust controllers and show that this is likeley due to optimization of certain robustness metrics indirectly.

1.2 Thesis contributions

The main contributions of this thesis are to advance RQOC on two frontiers: robustness *certification* and robust *optimization* of quantum dynamics. These are expanded upon in detail below:

1. Towards the first aim, in Chapter 5 that is derived from publications [Kha+23b; ONe+23], a measure of reliability of a control scheme that extends the traditional measure of accuracy, i.e. fidelity of the realized scheme, is developed. In the presence of noise, high fidelity itself is insufficient to gauge reliability of a control scheme, and extra effort is required to systematically search for solutions that are, both, robust against noise and have high fidelity [AGS21; JSL18]. This requires a notion of robustness and ideally a single measure that can capture robustness and fidelity, enabling the identification and construction of more efficient methods to find controls that satisfy both properties. The measure is general with respect to assumptions about the quantum noise and can be applied for a variety of QOC problems. More specifically, we make the following theoretical contributions:

    - We propose a novel statistical robustness measure called the Robustness Infidelity Measure (RIM) based on the pth-order Wasserstein distance to quantify robustness and fidelity of a control scheme.
    - We develop bounds on the RIM in terms of RIMs at higher order ps to motivate the practicality of using $p = 1$ for the RIM.
    - We provide some connections between our measure and other measures of robustness in quantum and classical control, namely the log-sensitivity and the differential sensitivity for the latter, and highlight its ability to be extended as a measure to compare control acquisition algorithms in the form of the algorithmic RIM (ARIM).
    - We demonstrate that a reason why RL control schemes are robust (and good RIM-wise) is because RL effectively optimizes the RIM as its objective function.

    And the following computational contributions:

   - We develop a systematic method and a consistency statistic to compare the robustness and fidelity performance of different quantum control schemes for the same control problem.
   - We conduct a computational analysis of the performance of different popular control acquision algorithms and the control schemes that they produce in terms of the RIM and ARIM.
   - We demonstrate the cost utility, in terms of optimization objective calls, of using RL to find robust control schemes for noisy control problems.

2. Towards the second aim, we consider the problem of RQOC in a model-free setting with the optimization of a stochastic/noisy target function – a particularly challenging setting where most standard QOC methods break down. We reformulate existing model-free RL methods [Lil+15; FHM18; Sch+17; Sch+15; Haa+18] and demonstrate their ability to find robust and high fidelity solutions in this setting. However, most machine learning methods [MRT18], including RL, require a lot of data or samples in the form of quantum measurements and are relatively difficult to deploy on NISQ devices where these might be prohibitively expensive to obtain. To reduce this cost, inspired by model predictive control [Gol+22], we develop a novel model-based RL algorithm that incorporates partial information of the controllable system and learns the rest of the model during the RL loop. The correct choice of the model ansatz places a strong prior on the space of all possible models that can be learned and reduces the number of samples needed for system identification considerably. More specifically, in Chapter 4 derived from the publication [Kha+21], we make the following contributions:

   - The problem of quantum state control is reformulated into a novel and scalable partially observed Markov decision problem for an RL agent to address.
   - Policy gradient RL algorithms are benchmarked against each other and gradient based control methods on noisy state control problems where it is shown that RL is able to produce high quality (robust and high fidelity) control schemes when gradient-based methods break in the presence of large amounts of noise.

- We study the effect of noise on the quantum control landscape and the associated problems like traps or volatile peaks. These findings become the fuel and the backdrop for the later findings in the thesis.

And in Chapter 6 based on the paper [Kha+23a], we make the following contributions:

- We develop a novel model-based RL algorithm LH-MBSAC (Learnable Hamiltonian Model-Based Soft Actor-Critic) that incorporates partial information of the controllable system and learns the rest of the model in depolyment. This is a step towards addressing the high sample complexity of model-free RL methods.
- We computationally demonstrate the sample efficieny of LH-MBSAC over model-free RL for noisy quantum gate control problems.
- We theoretically and computationally analyse the relationship between propagator or dynamics' error and Hamiltonian/model error and motivate the idea that successful quantum control is possible even with the wrong model, provided that the model's dynamical predictions are accurate.
- We demonstrate that the model learned by LH-MBSAC can be leveraged using gradient based control methods to improve the fidelity of control schemes found by RL methods.

1.3 Breakdown of thesis contributions

The author is the primary contributor to the idea development and implementation (theoretical, computational and writing) of Refs. [Kha+21; Kha+23b; Kha+23a] with the rest of the co-authors providing feedback and guidance.

Ref. [ONe+23] is joint work between the author and Sean O'Neil as the primary contributors with the rest of the co-authors providing supervisory guidance. Part of this work that is presented in Chapter 5, i.e., the proof connecting the RIM and the log-sensitivity, is the author's contribution.

1.4 Thesis outline

Chapters 2 and 3 presents the necessary background material for the thesis. We present the problem of QOC and RQOC and the general methods to solve these problems via variational methods or Bellman's principle of optimality in RL in Chapter 2. The idea of robustness of quantum control schemes is concretized. A review of different approaches for RQOC and QOC is presented including gradient-based optimal control methods leveraging a theoretical model, at one end, and model-free RL or black-box optimal control methods at the other. Due to the expansive nature of the techniques, we focus deeply only on RL for the latter category and build it up from a rudimentary level. Chapter 3 dives into robustness certification methods of quantum control schemes including techniques from classical control theory to motivate the development of statistical robustness quantification methods presented later in the thesis.

In Chapter 4, we compare the performance of different model-free policy gradient methods in RL for RQOC. We focus on the problem of energy landscape shaping of XX-Heisenberg spin chains with model noise and coarse-graining of fidelity measurements. RL performance for finding controllers is compared to a standard second order gradient-based QOC method (L-BFGS), with full access to an analytical model. We demonstrate that RL is able to tackle challenging, noisy quantum control problems where L-BFGS optimization algorithms struggle to perform well. We further perform a Monte Carlo robustness analysis under different levels of model noise to conduct a qualitative distributional comparison of fidelities for all controllers found by a particular control algorithm. We find that the controllers found by RL appear to be less affected by noise than those found with the standard gradient-based QOC method.

In Chapter 5, we present the Robustness Infidelity Measure (RIM) to quantify the statistical robustness and fidelity of quantum control schemes. The idea stems from a need to summarize and quantify the qualitative robustness performance of controllers in Chapter 4 into a single scalar measure. This could either be at an individual level thereby capturing the randomness in the performance of a single controller under model noise or a family of controllers belonging to a particular control algorithm class.

Treating quantum noise as a source of randomness, we transform the fidelity into a random variable that has a probability distribution. We then use a probabilistic distance from the ideal fidelity probability distribution to obtain the RIM. We use the RIM to demonstrate that not all high fidelity individual control schemes are robust w.r.t. RIM for a number of spin network transfer control problems. Further generalizing the RIM to the algorithmic RIM (ARIM), we show how it can be used to quantify the robustness of control acquision algorithms. Here we compare various QOC and RQOC techniques including RL and show that RL roughly optimizes the RIM which explains the comparatively high robustness of the control schemes found via RL. More generally, these measures should prove more useful in certifying the real-world performance of control schemes compared to just using the fidelity measure. We further connect the RIM to the time-domain log-sensitivity – a robustness measure from classical control.

In Chapter 6, we address the problem of high data requirements of model-free RL techniques for RQOC. Towards this end, we propose a novel physics-inspired model-based RL algorithm built on top of the standard soft actor-critic method that incorporates some knowledge of the controllable system (time-dependent components) into a learnable model. Further data then allows the rest of the system that is unknown to be characterized. The model is a simple ordinary differentiable equation (ODE) written using automatic-differentiation software so that stochastic gradient descent can update the generic ansatz of the system towards the true model. The model learning scheme is naturally constrained to produce physical predictions without any extra modifications and we highlight its ability to characterize completely unkown quantum systems. We demonstrate its RQOC potential by applying it on noisy open and closed system settings.

To summarize, in this thesis, we present novel ideas about quantifying robustness of quantum control schemes and model-free learning based methods to control and characterize quantum systems using realistic noise simulations. Due to the nature of the algorithms and problem we studied, it was out of scope for this thesis to apply the newly developed algorithms on a real NISQ device. However, at every step, we were very careful in making sure our simulations took the various forms of NISQ noise into account. Our ideas and methods were developed with the intention of being general without extra emphasis on the particular settings in which they are showcased. Where possible, we try to push them to probe their limitations on larger systems and more noise. We hope that they can prove fruitful in a variety of problems faced by the NISQ era quantum technologies' practitioner. Currently, the quantum industry landscape has matured to the level that commercial companies are also offering bespoke lower (pulse) level optimisations of quantum operations for specific architectures [Bal+21] which are being utilised in various settings by practitioners to engineer robust control schemes [Kud+22]. The techniques proposed in this thesis complement these commercial methods and are applicable at a higher level and for more general systems and therefore relevant in the same settings.

Part I

Background

The first part of this thesis reviews literature on topics in robust quantum optimal control at a high level to provide a better perspective on the contributions of this thesis. We also aim to equip the reader with an elementary background on concepts and techniques that will be used in subsequent chapters. We will expatiate more on topics that are closely related to the thesis contributions: specifically, the quantification of robustness for figures of merit used in quantum control problems and reinforcement learning based methods for control. For subsequent chapters, further mini-reviews will be provided on specialist topics that are more aligned with the core ideas in those chapters.

Chapter 2

Formulation and Control Methods

In this chapter, we formally introduce quantum control and highlight the various applications in quantum technologies and beyond that are enabled by it. Furthermore, we review various techniques for quantum control and, in particular, methods that enable control when the theoretical model of the quantum system is uncertain.

2.1 What is Robust Quantum Optimal Control?

The field of quantum optimal control (QOC) originated in the 1980s as a means to enable realization of various chemical reactions at the molecular level. On one hand, improvements in laser pulse shaping and pulse modulation enabled the construction of ultrashort minimal-time width pulses which made fine tunability of laser amplitudes possible [RRZ88; Zew88]. On the other hand, advances in non-linear laser spectroscopy allowed relative phases[1] between the frequency components of laser pulses to be modified to realize arbitrarily complex pulse shapes [Wei+86; SHW87]. One of the very first instances of QOC was made using a variational formulation of a two-photon state-transfer problem to optimally steer the reaction pathway towards the creation of a particular chemical species [TR85]. This was achieved by varying the laser pulse waveform and numerically solving for the ensuing dynamics until the waveform optimized a mathematical cost function encapsulating the desired final state. However, specifically for large molecules these methods did not translate well to the experimental setting due to inaccuracies in the dynamical simulation and real-world

[1] aimed at manipulating *coherence* or inherently non-classical properties of the atom/molecule and coherent laser control manipulates non-classical interference properties of wavefunctions of sub-atomic particles to drive desired behaviors

imperfections [Ass+98]. To bridge the gap between theory and experiment, an adaptive learning strategy to iteratively tune the laser pulse parameters was introduced in Ref. [JR92] that only required access to experimental data without any numerical simulation of the underlying dynamics. It was later used successfully for the control of chemical reactions [Ass+98; Bar+98].

A significant amount of development of the QOC field took place in the 1990s, first chiefly in terms of theoretical principles, but later increasingly also taking into account some unique problems with QOC when applied to the experimental setting, namely: loss of quantum properties over time due to environmental interactions; statistical noise due to estimation of inherently probabilistic quantum observables; and errors in the dynamics' model [Gla+15; Koc+22]. This led to the consideration of robustness in QOC strategies to these various noise sources. And, thus, began the field of robust quantum optimal control (RQOC).

For instance, standard QOC assumes that the quantum system essentially exists in an ideal vacuum with zero environmental interaction – which is often invalid in a real-world setting. To that end, an RQOC technique was developed where external controllable interactions can be used to reduce degradation dynamics due to environmental interaction and decouple the system from it [VKL99].

On the whole, both QOC and RQOC techniques are instrumental to enable quantum technologies in the fields of sensors, computation, simulation and optimization [Aci+18] but RQOC is more aligned to address the potential challenges posed by noisy quantum devices of our present time. RQOC is designed to extend or completely reformulate QOC to more directly address performance issues under various noise sources. In the literature and in this thesis, this distinction is an extra qualification (that is sometimes omitted) to highlight the overarching motivation behind the construction of a specific method or its operational performance in noisy settings. This distinction also more clearly earmarks QOC methods that are not designed to be robust and consequently break down in noisy settings.

2.1.1 Motivating Applications

In the following, as motivation, we focus specifically on some of these interesting applications that are enabled by RQOC and QOC. This section serves as a backdrop to contextualize the methods and techniques proposed in this thesis.

2.1.1.1 Quantum Metrology

Quantum sensing or metrology is the utilization of quantum systems to estimate unknown parameters with more precision than what is classically possible [GLM11]. Fundamental limitations on the uncertainty of the estimate are imposed by physical laws, e.g., Heisenberg uncertainty relations. Quantum sensing aims to measure physical observables with precision that reaches these limits. More concretely, given an unknown estimatable parameter θ and n single-shot probes[2], the central limit theorem necessitates that the estimation error scales as $|\hat{\theta} - \theta| \sim n^{-\frac{1}{2}}$ if the probe is classical. Quantum probes, enhanced via quantum entanglement, reach the more elusive Heisenberg scaling of $\sim n^{-1}$ in the estimation error – the ultimate precision limit dictated by the laws of quantum mechanics [Cav81; Bra92; LKD02; BC94]. The many technological applications include sensors for various scientific or commercial settings: biological, gravitational, clocks, plasma, magnetic [TB16; Col11; Lud+15; Lee+18; Jon+09]. However, recently, it was shown in Ref. [Len+22] that measurement errors can cause the Heisenberg scaling to be washed out and be constrained to the classical asymptotic limit in the number of probes. But using QOC methods, specifically the variational constuction of global unitary control operators that allow better distinguishibility of states after measurement, the optimum scaling can be recovered. QOC is also useful to prepare entangled squeezed states that can serve as probe states in the presence of state preparation errors [Kau+19].

2.1.1.2 Quantum Simulation

Quantum simulation can be described as using a quantum computer or some customizable quantum system to mimick or compute the dynamics of some target physical many-body quantum system [Dal+22] and is the most promising application within all the NISQ era quantum technologies to demonstrate a practical quantum advantage.

Promising applications include simulating: topological phases of matter [Sem+21] and quantum many-body scars [Blu+21b]; the physics that might result in high-temperature superconductors [Chi+19]; long/short-range dynamics of spin systems [Blu+21c; Zha+17]. Informally, for classical computers, such a task is expected to be exponentially more difficult in memory and compute resources when the number of

[2] single-shot implies that the probe can only be used once

particles in the physical system increases due to the exponentially growing number of the superposed system configurations that need to be tracked.

Note that there is no complexity proof for the non-existence of classical methods achieving the same efficiency [Dal+22]. QOC techniques have been also applied for robust preparation of Bose-Einstein condensate motional states and robust dynamical driving/simulation of a phase transition in a Mott-insulator made up of ultracold atoms in optical lattices [Fra+16]. The produced control schema are robust w.r.t. fluctuations of the system parameters whilst being realistically fast[3]. QOC has also been used to automatically calibrate single qubit operations for simulation and to simulate neutron-neutron dynamics on qudit (multilevel) systems [Fra+17a; Hol+20].

2.1.1.3 Quantum Computation

Quantum computation, which is more digital than quantum simulation, has the goal of manipulating a collection of quantum bits (qubits) using a universal set of quantum logic gates to perform novel computations – similar to what is currently possible to do using a classical computer. A universal set of gates is composed of three single-qubit gates and a two qubit controlled-NOT (CNOT) gate [Deu85].

Loosely speaking, quantum computation enables new types of algorithms for solving certain problems to be implemented that are superior to their classical counterparts. These algorithms essentially harness the constructive and destructive interference of qubits due to superposition to achieve their respective quantum advantage. Notable examples include: (a) Shor's algorithm for prime number factorization of some integer N that can run in $\mathcal{O}(\text{poly}(\log N))$ time versus its classical equivalent that runs with $\mathcal{O}(\exp(\log N))$ time complexity. This is essentially possible by speeding up the period finding problem using quantum Fourier transform enabled efficient discrete logarithm function [Sho94]; and (b) Grover's algorithm for search through an unstructured database with a promise of at least a quadratic time-complexity advantage over its classical counterpart. Although these algorithms are out of the reach of current NISQ era quantum devices, steady progress is being made to make them a reality. To that end, many QOC and RQOC techniques have been utilized for accurate and robust realization of single-qubit and two-qubit quantum gates (in the presence of state

[3] respecting the quantum speed limit – imposed by the Heisenberg energy-time uncertainty relation

preparation and measurement errors) on a variety of near-term noisy quantum computing architectures including nitrogen-vacancy centers [HZS20; Fra+17b], trapped ions [CZ95], Rydberg atoms [Blu+21c] and superconducting qubits [MG20]. QOC and RQOC methods have also been used to prepare initial or special non-classical or entangled states that are resources for various quantum computation subroutines [Siv+22a; Por+19; Chr+04].

Moreover, QOC techniques have also been used to optimize noisy quantum circuits composed of many quantum gates [Per+14]. Current noisy quantum computers can be greatly enhanced in their compute ability through the provision of Quantum Error Correction Codes (QECC) that enable fault-tolerant logical computing – just like classical error correction. Recently, with the help of RQOC and QOC methods, improvements in base single-qubit and two-qubit gate operations and suppression of environmental degradation effects allowed the very first practical demonstration of QECC on superconducting qubits [Ach+23].

2.1.1.4 Quantum Communications

Provably secure communication can be enabled by non-fallible Quantum Key Distribution (QKD) protocols [SP00; Wan+19] that are powered by the principles of quantum mechanics. QKD protocols can detect eavesdropping attempts by potential adversaries as errors in the transmitted signal. Secure communication can take place once the secret key has been successfully transmitted via QKD. It is predicted that Shor's algorithm will be capable of breaking cryptographic workhorses like the Rivest-Shamir-Adleman (RSA) protocol [RSA78] that power our present day communication networks using fault-tolerant quantum computers beyond the NISQ era. Thus, much work is underway to create truly secure post-quantum networks based on QKD, amongst other (theoretically less secure but practical) classical ideas [Pir+20].

However, there exist tangible challenges before QKD based communication can be deployed to replace the current global communications infrastructure. Most importantly, amongst other technical issues, QKD suffers from lossy transmission with distance. The error in signal transmission grows exponentially with distance of the fiber along which it is transmitted. To tackle this loss, a repeater that allows the creation of entanglement and its passage through intermediate particles over arbitrary distances (called quantum memories) is used to boost the transmission signal need to be employed [Bri+98; ATL15]. Phase control techniques have been applied to improve the contrast between a reference and a quantum signal in the optical fiber [Wan+19].

This concludes our survey of promising near and far term applications of quantum technologies. Quantum control is essential to many if not all of these. We now formalize the quantum control problem and review various techniques to solve it.

2.1.2 Problem formulation

We formally (and generally) introduce the QOC problem and a few related flavours that are studied for the remainder of this thesis. In a nutshell, QOC deals with the construction of some optimizable objective or cost function $\mathcal{F}_{\mathbf{u}}$, a functional that maps to $[0, 1]$, w.r.t. some control parameters $\mathbf{u}$ that numerically represents the goal or objective of the control problem which is then maximized (minimized) by an optimization strategy w.r.t. $\mathbf{u}$ subject to some constraints.

We represent the unitary dynamics of the n-qubit quantum system we wish to control by its Hamiltonian H that exists in the space of complex Hermitian $2^n \times 2^n$ matrices

$$H(\mathbf{u}(t), t) = H_0 + H_c(\mathbf{u}(t), t), \tag{2.1}$$

where H_0 is the time-independent system Hamiltonian and H_c is the control Hamiltonian parametrized by time-dependent controls $\mathbf{u}(t)$. The qubit is a two energy level system and can be generalised to d energy levels as the qu$d$it that is represented by $d^n \times d^n$ matrices. The system Hamiltonian represents the physical dynamics that we do not control and is usually assumed to be time-independent. We also adopt this modelling assumption throughout the course of this thesis unless stated otherwise.

There are fundamentally two types of quantum control problems: state preparation and gate preparation problems. Gate problems can be thought of as a generalization of state preparation problems in that we consider the task of finding a unitary process that maps all possible states in a given basis to some target states. In other words, the initial state which was fixed for the state preparation case is now made arbitrary. We deal with both cases in this thesis and now formalize them separately.

2.1.2.1 State preparation

Consider a closed n-qubit quantum system represented by the state $|\psi(t)\rangle$ in a Hilbert space $\mathcal{H}$ with dimension 2^n. Its dynamical generator H is given by Eq. (2.1) and its time evolution is governed by the linear ordinary differential equation (ODE) known as the Schrödinger equation,

$$\frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = -\frac{i}{\hbar} H(\mathbf{u}(t), t) |\psi(t)\rangle \tag{2.2}$$

where $\hbar$ is the reduced Planck's constant.

Informally, the control problem is generally the following: *starting from an initial state $|\psi_0\rangle$ at time $t = t_0$, we wish to reach some target state $|\psi_{target}\rangle$ at some final time $t = t_1$ by varying $|\psi(t)\rangle$ through some controllable parameters in its dynamics.*

An equivalent but mathematically more compact representation of the ODE dynamics is given by the unitary propagator U that solves Eq. (2.2),

$$U(t_0, t_1, \mathbf{u}(t)) = \mathcal{T} \exp\left(-\frac{i}{\hbar} \int_{t_0}^{t_1} H(\mathbf{u}(t), t)\, dt \right) \tag{2.3}$$

where $\mathcal{T}$ denotes time ordering operator. Note that the closed system dynamics governed by the Schrödinger equation are time-reversible. This is represented by the unitary property of the propagator: $U^\dagger U = U U^\dagger = \mathbb{1}$ where $\mathbb{1}$ is the identity operator.

In general, the goal of the control problem is quantified by a metric called *fidelity*, $\mathcal{F}$, over some physically admissible controllable parameters. In this case, the fidelity between the propagated controlled state $|\psi(\mathbf{u}(t), t)\rangle = U(t_0, t_1, \mathbf{u}(t)) |\psi_0\rangle$ and the target state is given by

$$\mathcal{F}(|\psi(\mathbf{u}(t), t)\rangle, |\psi_{\text{target}}\rangle) := |\langle \psi_{\text{target}} | U(t_0, t_1, \mathbf{u}(t)) |\psi_0\rangle\rangle|^2 . \tag{2.4}$$

This measures the similarity between the two states. In general, the fidelity is bounded, and without loss of generality we assume it lies in $[0, 1]$, where $\mathcal{F} = 1$ if and only if we have $|\psi(\mathbf{u}(t), t)\rangle = e^{i\phi} |\psi_{\text{target}}\rangle$, up to a global phase ϕ.

More precisely, the state preparation control problem can be formulated as the following fidelity optimization problem,

$$t_{\text{opt}}, \mathbf{u}_{\text{opt}}(t) = \underset{(t_1, \mathbf{u}(t)) \in \mathbb{X}}{\arg\max}\, \mathcal{F}(|\psi(\mathbf{u}(t), t)\rangle, |\psi_{\text{target}}\rangle). \tag{2.5}$$

In this problem, the final time is also considered a control parameter and can be optimized, where $\mathbb{X}$ represents the set of physical constraints or bounds on the control function $\mathbf{u}(t)$ and final time values, e.g., maximum or minimum t_{opt}.

Similarly, the open system control optimization problem is analogous to Eq. (2.5). To model the system's dynamics when it is no longer closed off or isolated from the surrounding environment we need to consider the state in the density matrix representation ρ. The open system dynamics, which are no longer unitary or reversible, evolve according to the master equation [BP+02; FDS12]

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(t) = -\frac{i}{\hbar}[H(\mathbf{u}(t), t), \rho(t)] + \mathfrak{L}(\rho(t)), \tag{2.6}$$

where $\mathfrak{L}(\cdot)$ describes the Markovian decoherence and dephasing dynamics (i.e., the environment),

$$\mathfrak{L}(\rho(t)) = \sum_d \gamma_d \left(l_d \rho l_d^\dagger - \frac{1}{2}\{l_d l_d^\dagger, \rho\} \right) \qquad \rho(t = t_0) = \rho_0, \tag{2.7}$$

l_d is a decoherence operator that crucially is not necessarily a unitary and $[\cdot, \cdot]$ and $\{\cdot, \cdot\}$ are the commutator and anti-commutator respectively. The density matrix fidelity is analogously given by $\mathcal{F}(\rho(t), \rho_{\text{target}}) = \text{Tr}\left[\rho(t)^\dagger \rho_{\text{target}}\right]$.

2.1.2.2 Gate preparation

A quantum logic gate is essentially a unitary evolution propagator U and its acquisition is the gate preparation control problem. Since the Schrödinger equation is a linear ODE, the propagator U also obeys an identical closed-system ODE given by

$$\frac{\mathrm{d}}{\mathrm{d}t}U(\mathbf{u}(t), t) = -\frac{i}{\hbar}H(\mathbf{u}(t), t)U(\mathbf{u}(t), t), \quad U(t = t_0) = \mathbb{1}, \tag{2.8}$$

where $U(\mathbf{u}(t), t)$ is the unitary propagator representing the arbitrary state evolution starting from an initial time t_0. The evolved propagator at $t = t_1$ is given by $U(t_0, t_1, \mathbf{u}(t))$ in Eq. (2.3). Its fidelity to realize a target gate U_{target} is

$$\mathcal{F}(U_{\text{target}}, U(t_0, t_1, \mathbf{u}(t))) = \frac{1}{2^{2n}} \left| \text{Tr}\left[U_{\text{target}}^\dagger U(t_0, t_1, \mathbf{u}(t)) \right] \right|^2. \tag{2.9}$$

The closed-system control optimization problem to implement U_{target} is

$$\mathbf{u}_{\text{opt}}(t), t_{\text{opt}} = \underset{(t_1, \mathbf{u}(t)) \in \mathbb{X}}{\arg\max} \, \mathcal{F}(U_{\text{target}}, U(t_0, t_1, \mathbf{u}(t))), \tag{2.10}$$

where $\mathbf{u}_{\text{opt}}(t)$ is the optimal control function for an optimal final time t_{opt} as before.

For the open system dynamics, we follow the master equation prescription for the density matrix in Eq. (2.6) and obtain an analogous propagator prescription. To characterize the gate implemented by $\mathbf{u}(t)$ in the density matrix prescription, we need to consider the evolution of a complete basis of states, $\{\rho_k\}_{k=1}^{2^{2n}}$. For this we introduce the Liouville superoperator matrix $\mathbf{X}$ that acts on an arbitrary vectorized state $\boldsymbol{\rho}$ (e.g., we stack the matrix columns but row stacking also works) to produce the evolution

$$\boldsymbol{\rho}(t) = \mathbf{X}(t)\boldsymbol{\rho}(t = 0). \tag{2.11}$$

This is equivalent to the tensor-matrix evolution [WBC11]

$$\rho(t)_{mn} = \sum_{\mu,\nu} X_{nm,\nu\mu}(t)\rho_{\mu\nu}(t = 0). \tag{2.12}$$

$X_{nm,\nu\mu}(t)$ is a fourth order tensor form of $\mathbf{X}(t)$ that encodes the evolution of the state element $\rho_{\mu\nu}$.

Thus, similar to Eq. (2.8), we define a superoperator $X(\mathbf{u}(t), t)$ which encodes the evolution of $\{\rho_k\}_{k=1}^{2^{2n}}$ and follows the linear ODE

$$\frac{\mathrm{d}\mathbf{X}(\mathbf{u}(t), t)}{\mathrm{d}t} = -\frac{i}{\hbar}(\mathbf{L}_0 + i\mathbf{L}_1)\mathbf{X}(\mathbf{u}(t), t), \quad \mathbf{X}(t = 0) = \mathbb{1} \tag{2.13}$$

where $\mathbf{L}_0, \mathbf{L}_1$ represent the superoperator version of the commutator map $[H(\mathbf{u}(t), t), \cdot]$ and $\mathfrak{L}(\cdot)$ the Markovian decoherence and dephasing dynamics, respectively.

Note that we factorize out an imaginary prefactor i to the left in Eq. (2.13) to unify the ODE for open and closed system dynamics. For $\mathfrak{L} \equiv \mathbf{0}$, the above reduces to the closed system dynamics of Eq. (2.10). Finally, we can transform the superoperator $X_{nm,\nu\mu}$ to the Choi matrix $\Phi/\operatorname{Tr}[\Phi]$ that is given by index reshuffling or partial transpose (and more formally a contravariant-covariant change of coordinates) [WBC11; Lic16],

$$\Phi_{nm,\mu\nu} = X_{\nu m,\mu n}. \tag{2.14}$$

Using $\boldsymbol{\Phi}$, the matrix version of $\Phi_{nm,\mu\nu}$ with trace normalisation, we obtain the generalized trace fidelity [FL11a] between the open system target and evolved propagator in Choi operator representation,

$$\mathcal{F}(\boldsymbol{\Phi}_{\text{target}}, \boldsymbol{\Phi}(t_0, t_1, \mathbf{u}(t))) = \operatorname{Tr}\left[\boldsymbol{\Phi}_{\text{target}}\boldsymbol{\Phi}(t_0, t_1, \mathbf{u}(t))\right]. \tag{2.15}$$

It can now be seen that the open system control optimization problem is analogous to Eq. (2.10) using the generalized trace fidelity and can be solved in a similar manner.

In Chapter 6, we use this for open and closed dynamics. In reality, we need to estimate the gate operation by Φ since the unitary is not observable. Estimating Φ is possible using ancilla-assisted quantum process tomography (AAPT) and the

Choi-Jamiolkowski isomorphism [Cho75; Jam72; Alt+03] for $2\log_d n$-qudit states and $\log_d n$-qudit gates.

Analogously to the above, Φ has a matrix version $\mathbf{\Phi}$. We decompose $\mathbf{\Phi}$ over a generalised $\mathfrak{su}(n^2)$'s basis $\{P_k\}_{k=1}^{n^4}$, e.g., Gell-Mann matrices [BK08],

$$\frac{\mathbf{\Phi}}{\mathrm{Tr}[\mathbf{\Phi}]} = \frac{\mathbb{1}}{n^2} + \sum_{k=2}^{n^4-1} q_k P_k \tag{2.16}$$

whose coefficients are

$$q_k = \frac{\mathrm{Tr}[P_k \mathbf{\Phi}]}{\mathrm{Tr}[\mathbf{\Phi}]} \in [-1, 1]. \tag{2.17}$$

q_k can be modelled as a binomial random variable $\mathrm{Bin}(M, p_k)$ with probability $p_k = \frac{1}{2}(1 + q_k)$ where M is the number of single-shot (Bernoulli) measurements [SM20].

We measure the faithfulness of the implemented gate $\mathbf{\Phi}(\mathbf{u}(t), t)$ w.r.t. the target gate (as another Choi state) $\mathbf{\Phi}_{\text{target}}$ using the generalised state-fidelity [FL11a],

$$\begin{aligned} F(\mathbf{\Phi}(\mathbf{u}(t), t), \mathbf{\Phi}_{\text{target}}) &= \mathrm{Tr}[\mathbf{\Phi}(\mathbf{u}(t), t)\mathbf{\Phi}_{\text{target}}] \\ &= \frac{1}{n^4} + \sum_{k=2}^{n^4-1} q_k^{\text{target}} q_k. \end{aligned} \tag{2.18}$$

Analogously to the closed case, the open control problem is to find an optimal control $\mathbf{u}^*(t^*)$ for an optimal final time $t^* \leqslant T$ (with T being the fixed upper bound), such that

$$\mathbf{u}^*(t^*) = \underset{\mathbf{u}(t),\ t \leqslant T}{\arg\max}\, F(\mathbf{\Phi}(\mathbf{u}(t), t), \mathbf{\Phi}_{\text{target}}). \tag{2.19}$$

2.1.2.3 Discretization

The exact solution of the time-dependent general dynamics discussed in Eq. (2.19) is given by the time-ordered operator

$$\mathbf{E}(t^*, \mathbf{u}^*(t^*)) = \mathcal{T} \exp\left(-\frac{i}{\hbar}\int_0^{t^*} dt'\ \mathbf{G}(t', \mathbf{u}^*(t'))\right)$$

for a unitary/Lindbladian generator $\mathbf{G}$. In practice, we solve for a piece-wise constant version of the dynamics represented by N fixed steps of $\Delta t = T/N$ of the fixed final time T.

Thus, $\mathbf{E}(\mathbf{u}(t), t)$ is discretized, which amounts to fixing $\mathbf{u}(t) = \mathbf{u}_m$ to be constant for each timestep such that $\mathbf{u}_m \in \mathbb{C}^{m \times C}$ is a finite dimensional array where C is the number of controls per timestep in the vector u_l parametrizing $H_c(u_l, t_l)$ and m is the number of total timesteps in the pulse, with $m \leqslant N$ for a maximum number of pulse segments N. The propagator is

$$\mathbf{E}(t, \mathbf{u}(t)) := \mathbf{E}(\mathbf{u}_m) = \prod_{l=1}^{m} E_l = \prod_{l=1}^{m} \exp\Big(-\frac{i}{\hbar}\Delta t \mathbf{G}(t_l, \mathbf{u}(t_l))\Big) \tag{2.20}$$

and the control problems in Eq. (2.10) and Eq. (2.19) are equivalent to

$$\mathbf{u}_m^* = \underset{\mathbf{u}_m=[u_1,\ldots,u_m]\in\mathbb{X}, m\leqslant N}{\arg\max} \mathcal{F}(\mathbf{\Phi}(\mathbf{E}(\mathbf{u}_m)), \mathbf{\Phi}(\mathbf{E}_{\text{target}})) \tag{2.21}$$

for a fidelity $\mathcal{F}$ and the time. Note that $\mathbf{u}_m$ is constrained to some maximum and minimum values given by $\mathbb{X} = \{\mathbf{u}_m : \forall c, l\ u_{\min} \leqslant u_{cl} \leqslant u_{\max} \in \mathbb{C}\}$. The constraints are applied separately to the real and and imaginary parts of the components of $\mathbf{u}_m$.

To summarize, we have formulated two types of quantum control problems pertaining to gate and state preparation as optimization problems where the objective is to realize some target state or gates on a quantum system by steering its dynamical evolution using some control functions. The objective is quantified by a fidelity metric which can be maximized by some optimization algorithm. For each problem, we have shown how to model the ideal unitary and environmental interaction versions of the dynamics. Next, we conduct a review of various algorithms that can be used to solve these optimization problems.

2.2 Control methods

We now review various RQOC and QOC techniques and highlight their various strengths and weaknesses. On a high level, they can be partitioned into two groups: (a) *model-based* methods that assume a theoretical model of the system $H(\mathbf{u}(t), t)$ one wishes to control which can therefore expedite some computational effort in maximizing the fidelity via the use of an analytical gradient or achieving robustness w.r.t. model uncertainties or environmental interactions via analytical Hamiltonian engineering; (b) *model-free* methods that do not assume any model and solve the control problem either completely independently of any model of the system or learn an effective model of the system during the control protocol.

Both types of methods are important in their own regimes. When equipped with a theoretical model that has very high probability of being correct, methods of type

(a) should be naturally preferrable. However, that is not always the case and a central theme of this thesis is to tackle the problem of control when confidence in the correctness of a theoretial model is low. This calls for the problem to be tackled using methods of type (b) which is done through the application of model-free methods to noisy control problems in Chapters 4 and 5 respectively. However, this too has its own disadvantages in that the model-free methods might not converge, could be unstable or consume too many resources such as physical queries of a quantum system that we wish to control or could be prohibitive in the nature of the queries such as full tomography that is likely not scalable with the system's size. This issue can be addressed by leveraging partial knowledge of the true model with the rest being learned using data acquired from sampling the controllable system and is explored in Chapter 6.

2.2.1 Model-based methods

2.2.1.1 Analytical methods

One of the earliest examples of analytical pulse shaping is the STImulated Rapid adiabatic passage (STIRAP) technique for efficient state transfer between two levels $|0\rangle \rightarrow |2\rangle$ in an atom or molecule via an intermediary connecting state $|1\rangle$ that remains unoccupied after the transfer is complete [Kuk+89]. In other words, during the transfer, no loss in the state population due to spontaneous emssion in $|1\rangle$ occurs. The method was demonstrated for exciting a multilevel system in sodium using a Stokes' laser to drive the $|1\rangle \rightarrow |2\rangle$ transition and the pump laser to drive the $|0\rangle \rightarrow |1\rangle$ transition [Gau+88]. It was found to be robust to laser frequency modulation, pulse shape and intensity errors compared to previous methods [CH88]. STIRAP seeks to realize the zero energy steady eigenstate of the Hamiltonian given by $|\mathbf{u}(t)\rangle = \cos\theta(t)\,|0\rangle + \sin\theta(t)\,|2\rangle$ by counterintuitively applying a Stokes' pulse first and then a bigger pump pulse. Since its discovery, STIRAP has been used to realize coherent superpositions of states, beam splitters in atomic interferometers, manipulation of laser-cooled trapped atoms and single photon generations [Vit+17].

Another popular analytical technique in trapped ions makes use of the Mølmer-Sørenson interaction [SM99; SM00]. It couples ion quantum states with vibrational modes or phonons to achieve many-body entanglement states. The typical state is given by $|g, n\rangle$ where the g denotes the two-level ion ground state and n denotes the vibrational mode. Typical transitions are driven by red-sideband, carrier and blue-sideband lasers driving $|g, n\rangle \xrightarrow{\text{red}} |g, n-1\rangle$, $|g, n\rangle \xrightarrow{\text{blue}} |g, n+1\rangle$, $|g, n\rangle \xrightarrow{\text{carrier}} |e, n\rangle$. The transitions (in the Lamb-Dicke regime[4]) are robust to changes in the other vibrational modes due to constructive and destructive interference effects of the transition pathways taken to achieve the many-body entangled states. Methods built on top of this interaction essentially power current trapped ion quantum computing architectures with a scalable number of qubits and quantum sensors that require preparation of nonclassical states [Pog+21; Pez+18].

Loss of quantum coherence due to interactions with environmental degrees of freedom can be addressed by applying analytical corrections to the system via the means of dynamical decoupling [VKL99]. This method effectively eliminates the effect of the environment on quantum dynamics, projecting the latter onto a subgroup of all possible coherent dynamics, thereby allowing for *fault-tolerant control.* Suppose the Hilbert space of the system coupled to an arbitrary environmental bath is given by $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_B$. The system Hamiltonian H_0 with a control term H_1 is given by,

$$H = H_0 + H_1(t) \otimes \mathbb{1}_B = \sum_{\alpha} \mathcal{W}_\alpha \otimes \mathcal{B}_\alpha + H_1(t) \otimes \mathbb{1} \tag{2.22}$$

where $\mathcal{W}_\alpha$ and $\mathcal{B}_\alpha$ are arbitrary linearly independent system and bath operators respectively.

Now, the idea behind dynamical decoupling is to exploit the decoupling control interaction $H_1(t) \subset \mathcal{C}_S$ that generates the control algebra $\mathcal{C}_S$ to cancel out system and bath mixing contributions to dynamics of $H_0 \subset \mathcal{I}_S$ where $\mathcal{I}_S$ is the interaction subspace and $\mathcal{C}_S \neq \mathcal{I}_S$. To first order, this is effectively done by removing the mixing terms in the average Hamiltonian $\bar{H}^{(0)}$. The first Magnus expansion term in the system-bath dynamical propagator, given by,

$$\bar{H}^{(0)} = \frac{1}{T_c} \int_0^{T_c} du\ U_1(t)^\dagger H_0 U_1(t) \tag{2.23}$$

where $U_1(t)$ is the propagator induced by $H_1(t)$ and T_c is its cycle time, i.e., when $U_1(t) = U_1(t + T_c)$.

The corrections appear in $\bar{H}^{(1)}$ which is the second Magnus term. If these corrections are applied on a timescale faster than the timescale of decoherence, the higher order Magnus terms are neglible and just correcting $\bar{H}^{(0)}$ effectively stops the decoherence accumulation in the system. These corrections can be applied using a group-theoretic

[4] vibrations are small compared to the laser wavelength

decoupling operator averaging where the decoupling operators are piece-wise constant $U_1(t) = g_j$ and form a finite group $\mathcal{G} = \{g_j\}$ that generate $\mathcal{C}_S$. This effectively produces an averaging of the system Hamiltonian given by,

$$\bar{H}^{(0)} = \frac{1}{|\mathcal{G}|} \sum_{\alpha} \sum_{j} g_j^{\dagger} \mathcal{W}_{\alpha} g_j \otimes \mathcal{B}_{\alpha} \tag{2.24}$$

and requires the application of a complete basis of g_j for a time $T_c/|\mathcal{G}|$ to achieve maximal averaging – completely cancelling the dissipatory dynamics. This technique has been applied on 5-qubit IBM and 19-qubit Rigetti transmon chips to demonstrate a high gate fidelity relative to free evolution of the system [Pok+18]. It can also be extended, using a similar argument as above by incorporating some analytical pulse shaping to achieve dynamically corrected gates [KV09].

Another method to dynamically correct gates is to derive pulse constraints to account for small arbitrary first order perturbations in a two level Hamiltonian. This makes use of the analytical solution to the Schrödinger equation for simple two-level systems [BD12] to derive an analytical relation between the dynamics of time-dependent noise fluctuations in the Hamiltonian and the propagator. Then, by requiring that these fluctuations vanish at a final time, corresponding constraints are obtained on the driving pulse that is also a function of these fluctuations.

A challenge in the above approach is the nonlinear phase in the fluctuations that makes fixing the propagator at final time difficult. By treating the phase as a topological winding number, the fluctuations can be deformed as a contour on the complex plane while preserving the phase, given that the origin is not crossed. This allows the final time propagator to be fixed and makes it possible to engineer a two-level analytic driver with noise-cancellation capability [BWS15].

Furthermore, a geometric framework for generally expressing dynamical decoupling and dynamic gate correction protocols is the so-called space curve quantum control (SCQC). Applications include, accounting for perturbative unitary errors, geometric cumulative error curves related to control parameters which can allow for post-selection of time-dependent robust controllers [BDB21; Don+21]. This idea also extends the limitation of dynamical decoupling and gate correction techniques that assume that the errors are static during gate operations. In SCQC, the evolution error is modelled as a geometric space curve in the space of control Hamiltonians. The curve's displacement between its initial and final time points quantify the error and closing the curve, i.e., making sure that both initial and final positions are the same ensures ideal system evolution.

We present a simple illustration of SCQC for a single qubit Hamiltonian H_{rabi} [Bar+22]

$$H_{\text{rabi}} = \frac{\Omega(t)}{2}\sigma_x + \epsilon\sigma_z \tag{2.25}$$

with a Rabi drive $\Omega(t)$ and a single axis quasistatic noise term σ_z with strength ϵ. To remove the contribution of noise in the propagator dynamics, we start by expanding the unitary propagator, U, in powers of ϵ. We obtain $U(t) = \sum_n \epsilon^n U_n(t)$ where the term U_n depends only on a term,

$$g_n(t) = \int_0^t dt' \ \exp\left\{ i \int dt''\Omega(t'')g^*_{n-1} \right\} \tag{2.26}$$

with $g_0(t) = 1$. The error correction constraint on the Rabi drive $\Omega(t)$ is $U_n(T) = 0 \leftrightarrow g_n(T) = 0$ for some final time T. The condition $g_n(T) = 0$ can be recast geometrically as a space curve. Specifically, to first order, g_1 induces a space curve $\mathbf{r}(t) = \text{Re}(g_1)\hat{\text{Re}(g_1)} + \text{Im}(g_1)\hat{\text{Im}(g_1)}$ where the hats denote unit vectors. The curvature of $\mathbf{r}(t)$ is the Rabi pulse shape,

$$\Omega(t) = \frac{\text{dRe}(g_1)}{\text{d}t}\frac{\text{d}^2\text{Im}(g_1)}{\text{d}t^2} + \frac{\text{d}^2\text{Re}(g_1)}{\text{d}t^2}\frac{\text{dIm}(g_1)}{\text{d}t} \tag{2.27}$$

and the length of the space curve is the evolution. The curvature and the length of the space curve can be optimized to realize robust time-optimal short pulses. The extension to larger number of qubits is possible by identifying space curves of the system's Schrödinger equation and casting them in an orthonormal Fresnet-Serret basis $\{\mathbf{e}_n\}$ that satisfy the relations $\mathbf{e}_n = -\kappa_{n-1}\mathbf{e}_{n-1} + \kappa_{n+1}\mathbf{e}_{n+1}$ where $\kappa_n = \frac{\text{d}\mathbf{e}_n}{\text{d}t} \cdot \mathbf{e}_{n+1}$ are the generalized curvature coefficients. These generalized curvatures then provide constraints on the control Hamiltonian terms that allow the realization of robust dynamically correcting pulses.

In addition to decoherence decoupling, SCQC can be used to correct for cross-talk or coherent errors due to a residual $\sigma_z \otimes \sigma_z$ type interaction in multi-qubit transmon systems [BDB21]. Also for a similar setting, a Pontryagin maximum principle for optimizing generalized leading order Taylor moments of control objectives subject to parametric uncertainties has been also proposed in Ref. [KBC21].

2.2.1.2 Gradient-based methods

It is often not possible to analytically solve for the general QOC problem for gate or state preparation. Instead, a numerical treatment is necessary. This is essentially the first relaxation of the control problem from the analytical case presented in the previous section where some control vector $\mathbf{u}$ as a solution to the QOC problem is updated iteratively via a numerical computation to maximize the target fidelity objective function $\mathcal{F}$. This is usually done by performing a gradient ascent update[5],

$$\mathbf{u} \leftarrow \mathbf{u} + \alpha \nabla_{\mathbf{u}} \mathcal{F} \tag{2.28}$$

where $\nabla_{\mathbf{u}} \mathcal{F} = \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathbf{u}} = \left[\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}u_1}, \ldots, \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}u_k}\right]$ is the gradient vector of the fidelity w.r.t. the control parameters u_k and α is an appropriate step-size parameter. Note that each control parameter u_k is a vector of length C which we further index as u_{kc} for the arguments that follow below.

A second-order[6] variant of the gradient ascent update is,

$$\mathbf{u} \leftarrow \mathbf{u} + \alpha \left(\mathcal{H}\right)^{-1} \nabla_{\mathbf{u}} \mathcal{F} \tag{2.29}$$

where $\mathcal{H} = \nabla^2_{\mathbf{u}} \mathcal{F}$ is the Hessian matrix where $\mathcal{H}_{ij} = \frac{\mathrm{d}^2\mathcal{H}}{\mathrm{d}u_i \mathrm{d}u_j}$. The Hessian incorporates nonlocal information into the optimization procedure by preconditioning the gradient with curvature information thereby accelerating the procedure. In practice, since the inverse Hessian is expensive to compute and requires $\mathcal{O}(k^3)$ operations, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) relation and its limited memory low-rank adaptation (L-BFGS) allow a faster iterative approximation of the inverse Hessian without computing any inverse that only requires $\mathcal{O}(k^2)$ operations [Zhu+97]. Let $\delta_k(.) := (.)_{k+1} - (.)_k$, we get the inverse Hessian $\mathcal{H}_{k+1}$ at the $k+1$ iteration via the BFGS recursion relation,

$$\begin{aligned}
Y_k &= (\delta_k(\nabla_{\mathbf{u}}\mathcal{F})^T \delta_k(\mathbf{u})) \\
Z_k &= \delta_k(\mathbf{u})\delta_k(\nabla_{\mathbf{u}}\mathcal{F})^T \\
\mathcal{H}_{k+1} &= (I - Y_k^{-1} Z_k)\mathcal{H}_k(I - Y_k^{-1} Z_k^T) + Y_k^{-1}\delta_k(\mathbf{u})\delta_k(\mathbf{u})^T.
\end{aligned} \tag{2.30}$$

[5] derived using a linear Taylor approximation to the fidelity: $\mathcal{F}(\mathbf{u} + \Delta\mathbf{u}) \approx \mathcal{F}(\mathbf{u}) + \Delta\mathbf{u}^T \nabla_{\mathbf{u}}\mathcal{F}$.

[6] derived using a quadratic taylor approximation to the fidelity: $\mathcal{F}(\mathbf{u} + \Delta\mathbf{u}) \approx \mathcal{F}(\mathbf{u}) + \Delta\mathbf{u}^T \nabla_{\mathbf{u}}\mathcal{F} + \Delta\mathbf{u}^T \mathcal{H} \Delta\mathbf{u}$ and then requiring that the gradient of the approximation $\nabla_{\mathbf{u}}\mathcal{F}(\mathbf{u} + \Delta\mathbf{u})$ be 0.

Moreover, another degree of freedom is the choice of either: (a) updating all control vector parameters concurrently or (b) updating them sequentially, e.g., either in the time dimension if k indexes time slices or across different control Hamiltonian terms [RNK12]. In the case of (b), only first order ascent updates are possible due to a large overhead imposed by sequential Hessian evaluations, however, it can be more computationally cheaper with similar guarantees[7] as the second-order method with some relevant application of step-size α control [Mac+11a].

L-BFGS is a very popular second-order gradient ascent algorithm that becomes a subroutine in many gradient-based quantum control algorithms *after* an accurate method to compute the gradient $\nabla_{\mathbf{u}}\mathcal{F}$ is known. Although L-BFGS has not been designed for noisy optimization there exist smoothing modifications that attempt to address this [AO20; Mac+11b; Shi+21] problem with limited success, the chief problem being that the noise scale is too large compared to the gradient.

We discuss different ways to obtain the gradient function $\nabla_{\mathbf{u}}\mathcal{F}$ which is possible when the Hamiltonian of the system is assumed to be fully known. We start with the popular GRadient Ascent Pulse Engineering (GRAPE) algorithm [Kha+05a; Fou+11; Mac+11a] which allows for an efficient computation of the gradient function using the Piece Wise Constant (PWC) ansatz of the control pulse discussed in Sec. 2.1.2.3. To illustrate an example, let us consider the gradient of the generalized state fidelity function in Eq. (2.15)

$$\begin{aligned}\nabla_{\mathbf{u}}\mathcal{F}(\mathbf{\Phi}_{\text{target}}, \mathbf{\Phi}(t_0, t_1, \mathbf{u}(t))) &= \operatorname{Tr}\left[\mathbf{\Phi}^{\dagger}_{\text{target}} \nabla_{\mathbf{u}} \mathbf{\Phi}(t_0, t_1, \mathbf{u}(t))\right] \\ &= \operatorname{Tr}\left[\mathbf{\Phi}^{\dagger}_{\text{target}} \left(\nabla_{\mathbf{u}} \mathbf{X}(t_0, t_1, \mathbf{u}(t))\right)^{\text{pT}}\right]\end{aligned} \tag{2.31}$$

where pT is the partial transpose operation to convert from Choi to superoperator representation. Now consider the closed system gate control problem where we can compute the gradient of the gate fidelity given in Eq. (2.9),

$$\nabla_{\mathbf{u}}\mathcal{F}(U_{\text{target}}, U(\mathbf{u}(t))) = \frac{2}{2^{2n}} \operatorname{Tr}\left[U^{\dagger}_{\text{target}} \nabla_{\mathbf{u}} U(\mathbf{u}(t))\right]. \tag{2.32}$$

All that remains is to find the gradient of the propagators. We illustrate how this gradient is built iteratively for the superoperator $\mathbf{X}$ by computing the gradient of each PWC term in the propagator. Firstly, recall that the superoperator in time-discretized form is composed of a train of time-independent superoperators,

$$\mathbf{X}(t_0, t_1, \mathbf{u}) = \prod_{l=1}^{k} \mathbf{X}_l = \prod_{l=1}^{k} \exp\left(-\frac{i}{\hbar}\Delta t(\mathbf{L}_0(t_l, u_l) + \mathbf{L}_1)\right). \tag{2.33}$$

[7] or better due to the variational derivation [RNK12]

So the gradient of the individual PWC term is well approximated[8] by

$$\frac{\mathrm{d}\mathbf{X}_l}{\mathrm{d}u_{lc}} \approx -\frac{i}{\hbar}\Delta t \left((\mathbf{L}_0(t_l, u_l))_{lc} + \frac{\mathrm{d}\mathbf{L}_1}{\mathrm{d}u_{lc}} \right) \mathbf{X}_l. \tag{2.34}$$

which when plugged into the gradient equation in Eq. (2.31) lets the gradient $\nabla_{\mathbf{u}}\mathcal{F}$ be built as a chain of computations. Here, each PWC gradient is independent of the head (forward) and tail (backward) propagator terms that can be precomputed and efficiently stored after matrix multiplication.

The case for the unitary propagator is the same but can be further improved using some further analysis [9] [Mac+11a].

Note that this version of GRAPE involves a lot of manual hardcoding of the gradients for different types of target objective functions but with a potentially restricting assumption that the controls are piece-wise constant. This can be relaxed next using a technique to compute the gradients accurately or analytically correctly via the adjoint method [Mac+18] without any restriction on the functional form of the control pulses. Consider the unitary gate control problem (again the argument generalizes to open systems). The idea is to notice that the difficult step in the computation of $\nabla_{\mathbf{u}}\mathcal{F}$ is the computation of $\nabla_{\mathbf{u}}\mathcal{U}$ which can be sidestepped by augmenting the ODE state with its gradient w.r.t. the control vector $U \to [U, \nabla_{\mathbf{u}}U]$ and solving for the augmented dynamics of the coupled system. The following augmented coupled dynamics ODE can be derived [KR09],

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} U(\mathbf{u}(t), t) \\ \nabla_{\mathbf{u}}U(\mathbf{u}(t), t) \end{pmatrix} = -\frac{i}{\hbar}\begin{pmatrix} H(\mathbf{u}(t), t) & 0 \\ \nabla_{\mathbf{u}}H(\mathbf{u}(t), t) & H(\mathbf{u}(t), t) \end{pmatrix}\begin{pmatrix} U(\mathbf{u}(t), t) \\ \nabla_{\mathbf{u}}U(\mathbf{u}(t), t) \end{pmatrix} \tag{2.35}$$

and can be fed into L-BFGS which calls the gradient function and the state that are obtained using numerical forward integration. The gradients obtained using this trick are more accurate compared to other strategies like brute-force finite differencing. We use this idea to obtain the gradient function that forms part of the L-BFGS quasi-Newton algorithm that is used in Chapters 4 and 5.

Increasingly, it is possible to automate even more gradient computation steps using efficient computational techniques to perform automatic or auto-differentiation. In a nutshell, auto-differentiation allows highly precise automatic computation of

[8] if Δt and the operator norms are small

[9] The new gradient expression is in the eigenbasis of the control Hamiltonian. Careful analysis yields more refined gradient expressions in terms of eigenvalue difference relations of the control Hamiltonian.

derivatives of complicated functions by composing known derivatives of elementary operations that make up these functions via the chain rule [Pas+19]. For example, suppose we have two functions f, g whose gradients are separately known to be f', g'. Then an elementary application of the chain rule yields the gradient of the composite function $f(g(\cdot))$ as $f' \cdot g'$. This simple idea is scaled up arbitrarily to compute gradients of complicated functions, e.g., those represented by neural networks, as long as they are compositions of smaller building-block differentiable functions.

Many methods involving automatic differentiation for model-based control have also been proposed [Sch+20; Sch+21a; GCM22; SM22]. It is possible to combine GRAPE with auto-differentiation methods to improve the efficiency of auto-differentiable gradient functions, allow computation of gradients with arbitrary control functionals and get gradients of non-analytic figures-of-merits such as the open system Fisher information in quantum metrology [GCM22]. Optimizations to the automatic gradient calculation are possible by analytically evaluating the chain rule using a GRAPE ansatz.

Also, a neural network controller that is optimized by making the propagator dynamics differentiable for robust state preparation has been proposed by utilizing auto-differentiable infrastructure to define the quantum control problem [Sch+20]. A differentiable stochastic differential evolution of the state during state preparation and stabilization subject to homodyne detection has also been proposed in a similar vein [Sch+21a]. Differentiable neural ordinary differential equations have been used to optimize neural control functions that map many gate parameters such as rotation angles to many high fidelity control parameters [SM22; PCM22].

The problem with GRAPE or gradient based methods is that stochastic noise in the Hamiltonian of the system in the real-world or in the *a priori* model used in GRAPE or measurement errors in estimation of the objective function can cause the method to fail to produce any high fidelity solution or cause the produced solution to significantly underperform at test-time since the model and system uncertainties were not taken into account. This can be addressed by batch optimization strategies of the average fidelity either uniformly [Wu+19a] or with respect to a utility function, i.e., risk function [GW21], which improves performance of the GRAPE control schemes.

We highlight a mathematical reason why this makes sense in Chapter 5 by showing that the average fidelity is actually a probabilistic robustness measure. We also use a variant of GRAPE or L-BFGS presented here as a benchmark method or baseline for comparison for other model-free algorithms on the time-independent state transfer problem in Chapter 4 and for leveraging the learned Hamiltonian model in Chapter 6 for the time-dependent gate control problem. Furthermore, since L-BFGS has performed well on finding high-fidelity energy landscape controllers [LSJ15a], its controllers are used in Chapter 5 as a relevant benchmark for individual controller comparisons.

Still, the inability to procure a model fundamentally limits the gradient based approaches discussed above. This is addressed by model-free methods, which in some sense are a further relaxation of the methods presented in this section.

2.2.2 Model-free methods

Model-free control methods do not require any Hamiltonian model of the system that needs to be controlled. Instead, either an approximate model is learned *ab initio* in the case of reinforcement learning and Bayesian Optimization or no model is learned and gradients are estimated using heuristics or sampling techniques in the case of genetic algorithms [Yan+20] or direct search methods like Nelder-Mead [NM65]. The usual price to pay with these techniques as opposed to the ones presented in prior sections is that the number of queries of the controllable system need to be necessarily larger since more information is now required to infer the direction of ascent in the abscence of a gradient field.

2.2.2.1 Conventional Gradient free methods

Nelder-Mead is a popular simplex-based control algorithm using direct search. Essentially, for a D-dimensional optimization problem, it creates and updates a $D+1$ dimensional polytope (simplex) $\{\mathcal{F}(\mathbf{u}_i)\}_i^{D+1}$ whose vertices are function evaluations that are updated towards an optimum direction. The main idea behind Nelder-Mead's heuristic update is to replace the worst vertex $\mathcal{F}(\mathbf{u}_{D+1})$ with a new point based on how good the reflection point $\mathcal{F}(\mathbf{u}_r)$ through the centroid of the remaining D points is compared to the rest of the points. The reflection point is computed as follows,

$$\mathbf{u}_r = \mathbf{u}_o + \alpha(\mathbf{u}_o - \mathbf{u}_{n+1}) \tag{2.36}$$

where $\mathbf{u}_o$ is the centroid of the top D points and α is a step size parameter in $[0, 1]$. The simplex is either expanded or contracted based on whether the reflection point is the best, worst or middle of the rank order of the $D+2$ points. Because Nelder-Mead is a model-free method and does not compute the gradients (recall that a Hamiltonian model is absent in model-free control), it is more robust to noise in the objective function evaluations. Moreover, Nelder-Mead performs a heuristic gradient descent ("the simplex rolls down the hill") via the direct search method outlined above which converges quickly without many function evaluations per descent iteration (i.e. 1-2 evaluations per simplex contraction or expansion).

It has successfully been used in noisy experimental settings [WA18] due to its non-reliance on gradient information [Kel+14a; DCM11], especially when obtaining such information is resource-intensive. A popular augmentation of this approach is the *Chopped RAndom Basis* (CRAB) method to parametrize the control function $\mathbf{u}(t)$ in a random Fourier basis (sampling the harmonics from a uniform distribution) and then use Nelder-Mead to optimize the coefficients of the basis using black-box fidelity function evaluations without extra computational overhead of gradient construction in model-based optimization [DCM11]. Due to the simplicity and parsimoniousness of the CRAB method, it has been used in many experimental quantum control settings including state and preparation for quantum computation, single-photon generation and manipulation of quantum dynamics in quantum simulation [Mue+22]. It is also used in the adaptive hybrid quantum optimal control protocol which combines Nelder-Mead with model-based methods as a post-refiner of pulses obtained after gradient-based control is performed using the model learned through system characterization [EW14]. Hence, we use Nelder-Mead as a benchmark RQOC method to measure its ability to produce robust controllers in Chapter 5 in contrast with other comparatively sophisticated model-free strategies.

Stable Noisy Optimization by Branch and Fit (SNOBFit) has been designed to filter out quite large scale noise in objective functionals [HN08] and is designed for optimization of noisy and expensive objective functions. The idea behind SNOBFit is to iteratively select new evaluation points of the objective function while maintaining a balance between the search for global and local optima. Each optimization call involves SNOBFit consuming input evaluation points $\{\mathcal{F}_i\}$ and then proposing new points $\{\mathbf{u}_i\}$ to be evaluated. It fits local models using objective function evaluations and implements a branching algorithm to partition the parameter space into smaller boxes with one function evaluation per box. The latter is a non-local search scheme that orders promising sub-boxes by the number of bisections required to get from the base box to that box. Sub-boxes with smaller bisections are worth exploring more. It does not rely on explicit gradient information and builds models of the optimization landscape. Thus, it should be able to cope with large amounts of noise in the form of controller and model uncertainties, environmental effects and singularities during optimization. SNOBFit, however, differs importantly from RL in the assumption that those models are linear. Moreover, its non-local optimization landscape exploration is not random and thus has comparatively a lot less variance in performance (that may or may not be poor).

Due to the simplicity of the models fit by SNOBFit for its acquisition of new evaluation points, we use it as a benchmark against methods that fit non-linear models like RL in Chapter 5.

2.2.2.2 Genetic Algorithms

Genetic or evolutionary algorithms are stochastic optimization techniques inspired by the biological idea of natural selection and are among the first numerical techniques employed for QOC purposes [JR92]. Here, the objective function is treated as a fitness function and a set of candidate parameters $\{\mathbf{u}_i\}$, called chromosomes, are *evolved* in a three-step iterative protocol:

1. **Selection:** A set of parent candidates are selected probabilistically w.r.t. their fitness function values $\mathcal{F}_i$ with probability,
$$p_i = \frac{\mathcal{F}_i}{\sum_i \mathcal{F}_i} \tag{2.37}$$
2. **Crossover:** Offspring candidates are generated by mixing parts of the chromosomal candidates at random locations.
3. **Mutation:** Finally, a small portion of the offspring candidate's parameters are replaced with random entries drawn from a uniform distribution to introduce mutations in the candidate population and encourage more fitness.

The steps are repeated, until convergence, with the children becoming the parents of the next generation. The appeal behind genetic algorithms is that, empirically, their probability of success in reaching a global optimum, in a control landscape with a lot of local sub-optima or traps, is higher since there is more exploration of the parameter space by the candidate set of parameters and the algorithm converges slowly to an optimum [ZSS14]. It has been shown that evolutionary algorithms and specifically differential evolution, a type of genetic algorithm that follows the same three-step protocol for iterative refinement of the candidate solution set[10], avoid local optima or traps in the landscape (discussed in more detail later in Sec. 3.1) better than quasi-Newton gradient-based method and Nelder-Mead [ZSS14]. They are able to find global optima in the presence of noise for molecular state prepartion [JR92] and adaptive enhanced phase estimation for quantum metrology [Lov+13].

Thus, these algorithms could be suitable for control problems with a lot of traps. In this thesis, since we do not study these types of hard quantum control problems, these methods are out of scope and will not be discussed further.

2.2.2.3 Bayesian Optimization

Bayesian optimization is a popular technique for optimizing black-box functions whose queries are expensive and noisy and the parameter dimension is less than 20 [Fra18].

Despite not directly exploring Bayesian optimization in this thesis, we provide an overview to draw out the similarities between its components and those of reinforcement learning (discussed next). Most of the terminology regarding the components can be well-understood in a Bayesian context and translates to the reinforcement learning setting. The fact that it is a lot simpler in principle than RL could also be advantageous, in particular few parameter settings.

Given a target objective function $\mathcal{F}$ to optimize, the technique proceeds by constructing its surrogate function $\hat{\mathcal{F}}$ that is maximized and also quantifies its uncertainty using process regression, that can be Gaussian or any parametric distribution, by assuming that the evaluations of the function $\mathcal{F}$ follow this distribution drawn from the corresponding process. Henceforth, we focus our description on the Gaussian process for simplicity but we also point out that the technique is not restricted to Gaussian processes. In probability theory, a Gaussian process is an infinite collection of random variables whose every finite subset (here, observations of the objective function $\{\mathcal{F}_i\}$) follows a multivariate Gaussian distribution. There are two main components in Bayesian optimization: a Bayesian surrogate $\hat{\mathcal{F}}$ for modelling the objective function $\mathcal{F}$ (exploitation of data) and an acquisition function for deciding which control parameters $\mathbf{u}_{D+1}$ to sample next (exploration to acquire new data). The essence of Bayesian optimization is to solve a decision problem ("what is the best control parameter based on data I have acquired thus far?') in the face of uncertainty ("spread in the surrogate probability distribution of the values the objective function will take").

[10] each candidate solution follows a trajectory in control parameter space

We describe the exploitation part first. The protocol starts by points $\{\mathbf{u}_o\}$ chosen uniformly at random in an initial space-filling experimental design. By assuming a prior Gaussian process distribution on the space of surrogate functions $\mathbf{P}(\hat{\mathcal{F}})$, the protocol involves updating this prior using Bayesian inference by connecting observed evalutions of the objective function $\{\mathcal{F}(\mathbf{u}_i)\}$ at the points $\{\mathbf{u}_i\}$ with that of the surrogate function $\{\hat{\mathcal{F}}(\mathbf{u}_i)\}$. Suppose that we have made D evaluations and denote the sequence as a vector for the objective function as $\mathcal{F}(\mathbf{u}_{:D}) = [\mathcal{F}(\mathbf{u}_1), \ldots, \mathcal{F}(\mathbf{u}_D)]$. Following Bayes' rule, the posterior distribution $\mathbf{P}(\mathcal{F}(\hat{\mathbf{u}}_{:D})|\mathbf{u}_{:D})$ is given by,

$$\mathbf{P}(\hat{\mathcal{F}}(\mathbf{u}_{:D})|\mathbf{u}_{:D}) = \frac{\mathbf{P}(\mathbf{u}_{:D}|\hat{\mathcal{F}}(\mathbf{u}_{:D}))\mathbf{P}(\hat{\mathcal{F}}(\mathbf{u}_{:D}))}{\mathbf{P}(\mathbf{u}_{:D})} \tag{2.38}$$

where $\mathbf{P}(\mathbf{u}_{:D}|\hat{\mathcal{F}}(\mathbf{u}_{:D}))$ is the multivariate likelihood, and

$$\mathbf{P}(\mathbf{u}_{:D}|\hat{\mathcal{F}}(\mathbf{u}_{:D})) = \frac{1}{(2\pi)^{D/2}\mathrm{det}(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\hat{\mathcal{F}}(\mathbf{u}_{:D}) - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\hat{\mathcal{F}}(\mathbf{u}_{:D}) - \boldsymbol{\mu})\right\}, \tag{2.39}$$

where $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\mathbf{u}_{:D})$ is the prior covariance matrix and $\boldsymbol{\mu} = \boldsymbol{\mu}(\mathbf{u}_{:D})$ is the prior mean vector. The power of the Gaussian assumption now pays off in that the posterior (predictive) distribution in Eq. (2.38) is also Gaussian and is simply

$$\mathbf{P}(\hat{\mathcal{F}}(\mathbf{u}_{:D})|\mathbf{u}_{:D}) = \mathcal{N}(\boldsymbol{\mu}', \sigma_D^2) \tag{2.40}$$

for any arbitrary control parameter $\mathbf{u}'$ with the posterior mean vector and variance given by

$$\begin{aligned} \boldsymbol{\mu}'(\mathbf{u}') &= \boldsymbol{\Sigma}(\mathbf{u}', \mathbf{u}_{:D})\boldsymbol{\Sigma}(\mathbf{u}_{:D}, \mathbf{u}_{:D})^{-1}\left(\hat{\mathcal{F}}(\mathbf{u}_{:D}) - \boldsymbol{\mu}\right) + \boldsymbol{\mu} \\ \sigma_D^2((u)') &= \boldsymbol{\Sigma}(\mathbf{u}', \mathbf{u}') - \boldsymbol{\Sigma}(\mathbf{u}', \mathbf{u}_{:D})\boldsymbol{\Sigma}(\mathbf{u}_{:D}, \mathbf{u}_{:D})^{-1}\boldsymbol{\Sigma}(\mathbf{u}_{:D}, \mathbf{u}') \end{aligned} \tag{2.41}$$

where $\boldsymbol{\Sigma}(\mathbf{u}', \mathbf{u}_{:D})$ is a D dimensional row vector indicating the covariance between $\mathbf{u}'$ and $\mathbf{u}_i$ for D evaluation points. Likewise, $\boldsymbol{\Sigma}(\mathbf{u}_{:D}, \mathbf{u}_{:D})$ is the covariance matrix. Usually, the mean function $\boldsymbol{\mu}$ is standardized to be a constant or is some lower order polynomial and the covariance function $\boldsymbol{\Sigma}$ is chosen from a breadth of kernel choices with some hyperparameters η that are optimized via maximizing the likelihood function.

The second part in the Bayesian optimization protocol is exploration: choosing or sampling the next control parameter $\mathbf{u}_{D+1}$ which is similar to the exploration problem for RL discussed in the next section. To that end, there are many acquisition functions and we list one simple acquisition function, called the Upper Confidence Bound (UCB) α_{UCB},

$$\alpha_{\mathrm{UCB}}(\mathbf{u}') = \boldsymbol{\mu}'(\mathbf{u}') + \lambda\sigma_D(\mathbf{u}') \tag{2.42}$$

where λ is a hyperparameter controlling the strength of exploration. The UCB captures the principle of *optimism in the face of uncertainty* that has theoretical connections with RL interpretations as optimistic dynamic programming [NP20]. The next point to sample is then simply

$$\mathbf{u}_{D+1} = \arg\max_{\mathbf{u}'} \alpha_{\mathrm{UCB}}(\mathbf{u}'). \tag{2.43}$$

Bayesian optimization has been used in quantum control settings with single shot fidelity estimation and Gaussian noise for state preparation problems including the realization of GHZ states and the Mott-insulating phase transition from a superfluid ground state [SM20]. The performance is improved if instead of a Gaussian process a binomial process is used to model single shot measurement data obtained from the fidelity which can thus be leveraged as an inductive bias by the algorithm to work in the limit when the number of shots tends to just one [SM20]. It has also been benchmarked for ultra-cold ordered state preparation tasks to be competetively better against Nelder-Mead and genetic algorithms [Muk+20]. The strength of Bayesian optimization is that the simplifying modelling assumptions in the exploitation part allow it to require much less data compared to more expressive function approximation based approaches used in RL. Furthermore, for the exploration part, the probabilistic modelling of the surrogate function allows one to incorporate planning in the sampling methodology which is usually absent in RL and further allows one to reduce the number of evaluation points needed for convergence. A weakness of the approach is that it is not scalable when the control parameter dimension is very large and solving more demanding decision problems is required with larger combinatorial loads where RL usually excels, including mastering gameplay [Sil+18] and problems including decision making with partial observations/information. Improving the ability of Bayesian optimization methods to work well in high-dimensional settings is an area of active research [Wan+16; Eri+19].

2.2.2.4 Reinforcement Learning

Reinforcement learning (RL) is another framework for blackbox decision making in the presence of uncertainty that is similar to Bayesian optimization. RL models the interaction of an agent with an environment with the goal of finding the optimal behavior policy of the agent that maximizes a reward signal the agent receives from the environment. At its heart, RL is comprised of two processes involving a policy function that models the agent's behavior and the value function that models the agent's quantification of the reward conditional on any of its possible behavioral actions. RL proceeds then by solving two related problems in a loop: the *prediction problem* where the value function is learned given some behavioral policy and an *optimal control problem* where the optimal behavioral policy is learned given some value function. Like Bayesian optimization, RL also tackles the joint problem of exploration (what's the best action to try to gather as much information about the system as possible?) for the prediction stage with exploitation (what's the best action to choose to maximize total reward?). The major difference between the two approaches is that RL treats the problem using dynamic programming that is described shortly as opposed to the Bayes' rule update described previously. RL has been successfully used for tackling quantum control in challenging noisy environments, resulting in similar or better performances compared to standard control methods. Promising results include the stabilization of a particle via feedback in an unstable potential [WAU20], optimizing circuit-QED, two-qubit unitary operators under physical realization constraints [Dal+20b], and optimizing multi-qubit control landscapes suffering from control leakage and stochastic model errors [Niu+19b], among many others.

Recall from Chapter 1 that one aim of the thesis is to develop RQOC methods for robust optimization of quantum dynamics. We will develop novel RL methods to achieve this goal by modifying existing model-free RL methods in the quantum domain and later propose a novel model-based RL method that improves upon the former methods. Given that the theoretical model of the quantum system in noisy settings is uncertain, RL is well suited for this domain which we demonstrate by comparing it to other benchmark algorithms discussed in this chapter. RL successfully addresses challenging, noisy quantum control problems with the promise of inherent robustness [Niu+19c; Kha+21; Kha+23b; Dal+20c; Siv+22b; Buk+18].

The RL problem [SB18a] is formulated as a Markov Decision Process (MDP) represented by the four-tuple: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P)$ signifying the state, action, reward and Markov[11] transition spaces with $\gamma \in [0, 1]$ being the discount factor. The MDP problem involves discrete transitions. At each timestep j in the MDP framework,

[11] the Markov property simply imposes the constraint on the transition probability: $P(s_{j+1}, r_j | s_j, a_j) = P(s_{j+1}, r_j | s_j, a_j, s_{j-1}, a_{j-1}, \ldots, s_1, a_1)$

given an initial state $s_j \in \mathcal{S}$, a next state $s_{j+1} \in \mathcal{S}$ can be achieved that carries with it some reward $r_j \in \mathcal{R}$ that is obtained by performing an action $a_j \in \mathcal{A}$. State transitions are assumed to be probabilistic and captured by the Markov transition dynamics model $P(s_{j+1}, r_j | s_j, a_j)$, indicating the probability of going from s_j to s_{j+1} with the action a_j, getting a reward r_j. A trainable policy function $\pi(a_j|s_j)$ that represents the *agent*'s behavior policy is generally a probability distribution of executing action a_j given state s_j. The agent follows π by sampling an action a_j, interacts with an environment $\mathcal{E}$ and associates a state transition $Y : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ with a reward function $\mathcal{R}(s_j, a_j)$.

The agent's (π) goal is to maximize the discounted cumulative rewards called the returns,

$$\eta(\pi) := \mathbb{E}_{\pi,P}\left[\sum_{k=0}^{\infty} \gamma^k \, \mathrm{r}_k\right], \tag{2.44}$$

where the sum is over all timesteps[12] k. Here, the discounting factor γ prioritizes immediate rewards over future rewards. More generally, returns for timestep j are given by,

$$R_j = \sum_{k=0}^{\infty} \gamma^k r_{j+k}. \tag{2.45}$$

The agent tries to maximize $\mathbb{E}_{\pi,P}[R_j]$, the expected returns given some initial state s, where we have the discrete expectation operator $\mathbb{E}_{\pi,P}[(\cdot)] = \sum_{s',r,a} \pi(a|s)P(s',r|s,a)(\cdot)$.

The value function $V_\pi(s)$ is the expected returns following π starting from some state s. Mathematically,

$$V_\pi(s) = \mathbb{E}_{\pi,P}[R_j | s_j = s] = \sum_{s',r,a} \pi(a|s)P(s',r|s,a)\left(r + \gamma V_\pi(s')\right). \tag{2.46}$$

And, likewise, the state-action value function $Q_\pi(s,a)$ is the expected returns given some initial state s and action a following policy π,

$$Q_\pi(s) = \mathbb{E}_{\pi,P}[R_j | s_j = s, a_j = a] = \sum_{s',r,a'} P(s',r|s,a)(r + \pi(a'|s')Q_\pi(s',a')). \tag{2.47}$$

Finally, we note the connection between Q and V: $V(s) = \max_{a'} Q(s,a')$.

In a nutshell, the RL objective of maximizing expected returns is achieved in two steps as mentioned before which form the core principles of the overarching RL strategy. Namely,

[12] a timestep is simply an iteration of the MDP

1. *Policy evaluation* (Prediction): Given some policy π, we compute the value function (Q or V). These form our prediction of the returns and inform the decision-making of the behavior policy.

2. *Policy improvement* (Control): Given a value function, we compute the optimal policy π, e.g., by acting greedily w.r.t. the value function, where the agent takes the action with the highest predicted value at each state.

The first step is achieved using the principle of dynamic programming. This is usually done approximately via sampling in RL since an exhaustive search over all possible states, actions and rewards is generally not computationally tractable when their corresponding spaces are high-dimensional. By noting that the right hand side expansion of value functions Q, V is recursive and is exactly the application of a Bellman operation that is commonly used in dynamic programming [Bel52], we apply iterative Bellman updates of the following form,

$$V_\pi(s)^{(k+1)} = \sum_{s',r,a} \pi(a|s)P(s',r|s,a)\left(r + \gamma V_\pi^{(k)}(s')\right) \tag{2.48}$$

to obtain the updated value function at iteration k. The Bellman update is a contraction in the space of value functions [BT96] and thus in the limit $k \to \infty$, we arrive at the fixed point, the optimal value function $V^*(s)$ which is independent of π,

$$V^*(s) = \max_\pi V_\pi(s) = \max_a \sum_{s',r} P(s',r|s,a)\left(r + \gamma V^*(s')\right). \tag{2.49}$$

A similar argument holds for Q. Learning Q involves approximately solving the Bellman optimality equation iteratively at every timestep k,

$$Q_\pi^{(k+1)}(s,a) = \sum_{s',r} P\left(s',r|s,a\right)\left[r + \gamma \max_{a'} Q_\pi^{(k)}(s',a')\right]. \tag{2.50}$$

Step two, the policy improvement step, involves updating the policy using the value function. The policy improvement theorem [SB18a] guarantees that the (new) greedy policy w.r.t. the value function $\pi'(s) = \arg\max_{a'} Q_\pi(s,a')$ is better than π. That is, for two deterministic policies π, π', if $\forall s \in \mathcal{S}$ we have that $Q_\pi(s,\pi'(s)) \geqslant V_\pi(s)$, then $V_{\pi'}(s) \geqslant V_\pi(s)$. Alternating between policy evaluation and policy improvement ad infinitum yields the fixed point where the final updated policy $\pi^{(k+1)} = \pi^{(k)} = \pi^*$ is optimal and likewise the value functions are optimal and any further iteration will yield the same functions. General theorems for policy and value functions guarantee iterated policy improvement [SB18a]. We define the number of agent-environment interactions needed to find an approximately optimal policy π^* as the *sample complexity.* The RL control problem becomes a problem of finding the optimal control policy π^* given by

$$\pi^* = \arg\max_{\pi} \eta(\pi). \tag{2.51}$$

The Markov transition model P can be difficult to learn without completely exploring the state and action spaces so the aforementioned approach only captures RL in principle which in practice can look very different. Likewise, the Bellman update can only really be approximate or a sample/Monte Carlo type update since the full update requires P and computing the best value function for a given state or action that requires entirely exploring the corresponding spaces. In reality, P is not needed for approximately solving the Bellman equation. The updates to the value and polciy function are made using samples from the four-tuple MDP space $\{s_j, a_j, s_{j+1}, r_j\}$ stored in the replay buffer or dataset object $\mathcal{D}$ (also called experience) using some exploration strategy (e.g., agent takes random actions or ϵ-greedy random actions[13]). This procedure is also sometimes called bootstrapping. One of the simplest ways to update the state-action value function is the SARSA (State Action Reward State Action) update,

$$Q_\pi^{(k+1)}(s_j, a_j) = Q_\pi^{(k)}(s_j, a_j) + \alpha\left(r_j + \gamma Q_\pi^{(k)}(s_{j+1}, a_{j+1}) - Q_\pi^{(k)}(s_j, a_j)\right) \tag{2.52}$$

which is a sample-based version of the dynamic programming update shown in Eq. (2.50) and does not use the transition model. Here, α is the learning rate hyperparameter. The celebrated Q-learning algorithm [WD92] involves the following policy evaulation update approximation,

$$Q_\pi^{(k+1)}(s_j, a_j) = Q_\pi^{(k)}(s_j, a_j) + \alpha\left(r_j + \gamma \max_{a'} Q_\pi^{(k)}(s_{j+1}, a') - Q_\pi^{(k)}(s_j, a_j)\right). \tag{2.53}$$

Furthermore, the SARSA update for the value function V also has an analogous form,

$$V_\pi^{(k+1)}(s_j) = V_\pi^{(k)}(s_j) + \alpha\left(r_j + \gamma V_\pi^{(k)}(s_{j+1}) - V_\pi^{(k)}(s_j)\right) \tag{2.54}$$

and is called the *temporal difference* (TD) update [Sut88].

An alternative to the above value-function based policy evaluation update is to directly optimize the policy function using a gradient signal. The gradient signal in question, $\nabla_\pi \eta(\pi)$, is obtained by differentiating the returns $\eta(\pi)$ which is possible to

[13] switch from determinisitc to random action with probability ϵ

Algorithm 1: Reinforcement learning loop

```
Initialize empty dataset $\mathcal{D}$, parametrized random policy $\pi_\theta$, $k \leftarrow 0$
Observe initial state $s_0$
while $k < T/\Delta t$ do
    Execute $\mathbf{a}_k \leftarrow \pi_\theta\left(\cdot|\, \mathbf{s}_k\right)$
    Observe $\mathbf{s}_{k+1}$, $\mathrm{r}_k \leftarrow \mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$
    Store $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k)\}$
    $k \leftarrow k + 1$
 // if require update: perform model-free update of parameters
 (e.g. policy $\pi_\theta$ )
```

do due to the *policy gradient* theorem [SB18a]. There exists a class of RL algorithms where the gradient $\nabla_\pi \eta(\pi)$ form varies; these are called policy gradient algorithms. One example of the policy gradient is

$$\nabla_\pi \eta(\pi) = \mathbb{E}_P \left[\nabla_\pi \log \pi(s|a)(Q_\pi(s, a) - b(s))\right] \tag{2.55}$$

which forms the core of the family of actor-critic algorithms [SB18a] where b is some variance reducing baseline function that can be estimated along with the rest of the functions within the expectation operator. The critic is synonymous with the value function that critiques the actor or agent, whose behavior is represented by the policy function. Note that policy gradient algorithms are not restricted by requiring a value function in $\nabla_\pi \eta(\pi)$. In essence, the critic or the value function is used to reduce the high variance in the reward function due to the non-stationary nature of the MDP. The adjusted value function in the policy gradient informs a more realistic estimation of the reward function which could instead be directly used in the former's place albeit paying the price of high variance rewards – this encapsulates the policy gradient update in the algorithm called REINFORCE [SB18a].

The policy gradient approach is particularly well-suited for continuous state and action spaces where the previous value-based procedures fall short in terms of effectively exploring the space. For such high-dimensional spaces, policy gradient methods are quite effective in optimizing the returns $\eta(\pi)$ using direct updates to the policy. Since the quantum control problem is a continuous control problem, we are primarily concerned with policy gradient algorithms – those, directly used in this thesis, is summarized next.

One final thing to note is that in the practical quantum control setting, in order to be able to represent the continous value and policy functions as statistically learned functions, we need to make use of an expressive function approximation that is able to capture their complex behavior in a generalizable manner. This step is necessary for the value and policy functions to generalize to larger spaces [Lil+15] and brings us in the realm of deep reinforcement learning where we use neural networks for function approximation which do the job well and function approximation scales favourably with dataset size. We denote the neural network functions representing pieces of the RL algorithm using a single greek letter subscript. For example, the neural network policy function becomes π_θ with θ nonparametrically denoting its trainable weights and biases, following RL literature [SB18a]. Likewise, we can also assume a similar nonparametric neural network form for the value functions Q_ϕ and V_ϕ. Then, using backpropagation [RR96], i.e., the chain rule, we can update θ in the direction of the policy gradient

$$\theta \leftarrow \theta + \nabla_\theta \eta(\pi_\theta). \tag{2.56}$$

The policy outputs usually parametrize the mean and covariance $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ of a learnable multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ from which the action is drawn. Both policy and the value functions (π and Q or V) are simple multilayer perceptrons throughout the thesis. A schematic of the general QOC RL loop is illustrated in Algorithm 1 for some fixed final time T and timestep size Δt.

The specifics of how to transform the QOC optimal control problem to the MDP formalism are technical and are covered in later chapters where the details are relevant: specifically in Chapter 4 for the state control problem and in Chapter 6 for the gate control problem.

We next cover policy gradient RL algorithms in more detail.

Policy Gradient RL Algorithms

Here we review a number of popular policy gradient algorithms that have been successful in classical RL literature. We empirically evaluate these for benchmarking reasons in Chapter 4 on some standard static quantum control problems. This empirical comparison helps motivate the choice of whether one should just use one of these algorithms for generally applying RL techniques on a quantum control problem or whether different policy gradient algorithms provide stratified advantages for different subsets of QOC problems.

Due to the wide breadth of policy gradient RL algorithms that can be found in the current classical RL literature, we only consider a few representative algorithms in this thesis: proximal policy optimization (PPO) [Sch+17], deep deterministic policy gradient optimization (DDPG) [Lil+15], twin delayed DDPG (TD3) [FHM18] and the soft actor-critic (SAC). They were chosen because of their popularity, performance and stability in the classical RL literature.

Here, we summarize the core ideas in each algorithm and draw out their connections. Refer to the original papers for the fully fleshed out details.

DDPG

DDPG [Lil+15] is an actor-critic algorithm that makes use of the deterministic policy gradient theorem [Sil+14] to iteratively improve a deterministic neural network policy function $\mu_\theta(s)$. The idea behind DDPG is to stably generalize Q-learning to high-dimensional continuous state/action spaces using neural network function approximation. A deterministic policy helps in simplyfing the form of Q and the policy gradient reducing the number of stochastic moving parts in the algorithm thereby reducing the variance in performance and improving the algorithm's stability. The deterministic policy gradient is given by

$$\nabla_\theta \eta(\mu_\theta) = \mathbb{E}_{\rho^\beta(s)}\left[\nabla_{\mu_\theta} Q_\phi(s, \mu_\theta(s)) \nabla_\theta \mu_\theta(s)\right], \tag{2.57}$$

where $\rho^\beta(s)$ represents the probability of visiting some state $s \in \mathcal{S}$. This gradient is used to update the policy parameters using gradient ascent, cf. Eq. (2.56). The state-action value function Q_ϕ is updated via fitting the neural network Q_ϕ using supervised learning to the bootstrapped SARSA update target in Eq. (2.52). Moreover, DDPG is an off-policy algorithm in that the deterministic policy function μ_θ can be updated using replay experience of other, usually older, policy functions. This means that the replay buffer dataset $\mathcal{D}$ can be very large and each update step can be made, with high probability, using samples with a significant proportion of uncorrelated MDP transitions. Exploration of the state and action spaces is done by adding temporally correlated noise sampled from an Ornstein-Uhlenbeck process $\epsilon \sim \text{OUP}$ for improved exploration efficiency [UO30],

$$\mu'_\theta(s) \leftarrow \mu_\theta(s) + \epsilon. \tag{2.58}$$

Another key idea, that is used by most if not all of the deep policy gradient algorithms, is the target function for both the policy and the value function for improving algorithmic stability during training. The target function is simply another function

$f_{\theta'}$ whose parameters θ' are related to the original function f_θ parameters θ using the exponential average update $\theta \leftarrow (1-\alpha)\theta + \alpha\theta'$. Intuitively, we can understand why a target function can crucially improve algorithmic stability by taking the policy function target, $\pi_{\theta'}$ that is used actively in the moving parts of the algorithm, e.g., taking actions in the environment but only its counterpart π_θ undergoes the policy gradient update. This smoothes out the behavior of the acting policy $\pi_{\theta'}$ whose parameters are only gradually moved towards the updated policy's parameters. The same applies to value functions whose predictions also undergo more smooth updates and improve convergence properties of the algorithm.

TD3

TD3 [FHM18] is an improved version of DDPG that corrects for the overestimation bias found in Q_ϕ due to SARSA type bootstrapping updates, a problem that is a well-known property of temporal difference learning [Sut88] since the estimate of the value function for a given state s_j is updated using the value estimate of the next state s_{j+1}. To address this consistent overestimation propensity, TD3 uses two Q different functions which form the supervised learning SARSA targets for either Q function of the form,

$$y_i = r_i + \gamma \min_{i=1,2} Q^{(i)}_{\phi'}(s_j, \mu_{\theta'}(s_j) + \epsilon). \tag{2.59}$$

This idea coupled with delayed policy and value function updates – ensuring updates happen only when the immediate accumulated error in the learning signal is small enough – is shown to reduce the overall accumulation of temporal difference errors due to the consistent overestimation bias of the value function.

PPO

PPO is not an actor-critic algorithm. It is also on-policy, implying that the current policy can only be updated using replay experience of recent past policies so the replay buffer is emptied after every few iterations and is typically smaller than for an off-policy algorithm. PPO is a first order approximation to the Trust Region Policy Optimization (TRPO) method [Sch+15]. Both methods sample the environment using the policy equipped with some exploration strategy and then do the policy improvement step by optimizing a surrogate objective function. The surrogate is based on the idea of conservative policy iteration [KL02] which provides explicit lower bounds on the improvement of returns under some new policy π_{new} over some old policy π_{old}. This is encapsulated by the update,

$$\pi_{i+1} = \arg\max_{\pi} \left[\eta(\pi_i) + \sum_{s,a} \rho_{\pi_i}(s)\pi(a|s)A_{\pi_i}(s,a) + 2\frac{\Omega\gamma}{(1-\gamma)^2} \max_{s} D_{\text{KL}}(\pi_i, \pi) \right] \tag{2.60}$$

with the advantage function $A_{\pi_i}(s,a) = (Q_{\pi_i}(s,a) - V_{\pi_i}(s))$; $\Omega = \max_s \max_a |A_{\pi_i}(s,a)|$ and ρ_π is the state visitation probability distribution of the policy π. TRPO effectively converts the analytical update Eq. (2.60) into a practical second-order constrained optimization problem, which we write in the unconstrained manner below,

$$\theta_{i+1} = \arg\max_{\theta_i} \mathbb{E}_j \left[\frac{\pi_{\theta_i}(a_j|s_j)}{\pi_{\theta_{\text{old}}(a_j|s_j)}} A_{\pi_{\text{old}}}(s_j, a_j) - \beta D_{\text{KL}}(\pi_{\text{old}}, \pi_{\theta_i}) \right], \tag{2.61}$$

for some constraining hyperparameter β. TRPO then uses a trust region method [BGN00] to compute the Hessian of the KL-divergence with a backtracking line search [NY98] to update the parameters of the policy. Note that in earlier methods, there is no objective constraint on the policy that makes sure it does not vary wildly during parameter updates for different episodes – an episode is one full RL loop shown in Algorithm 1. TRPO improves upon this by using the KL-divergence constraint D_{KL} between the new π_{θ_i} and old policy π_{old} to make sure its variation is constrained during each update.

The constrained optimization problem in Eq. (2.61) is unnecessarily complicated and precludes noisy policy and value function architectures or parameter sharing between value and policy functions. PPO simplifies this objective by proposing simpler clipped variation bounds on the KL-divergence that can instead be used directly in the parameter updates of the policy. The first order approximation to Eq. (2.61) is given by

$$\theta^{i+1} = \arg\max_{\theta_i} \mathbb{E}_j \left[A_{\pi_{\text{old}}}(s_j, a_j) \min\left(\frac{\pi_{\theta_i}(a_j|s_j)}{\pi_{\theta_{\text{old}}}(a_j|s_j)}, \text{clip}\left(\frac{\pi_{\theta_i}(a_j|s_j)}{\pi_{\theta_{\text{old}}}(a_j|s_j)}, 1 \pm \delta \right) \right) \right] \tag{2.62}$$

where the clip function truncates the values to stay within $[1-\delta, 1+\delta]$ for some $\delta \in [0,1]$ which is the hyperparameter constraint on the KL-divergence of the new and old policy.

SAC

Soft Actor-Critic (SAC) is an off-policy actor-critic algorithm. It is designed to address two issues with previous policy gradient algorithms: large sample complexity and non-robustness to changes of various hyperparameters. SAC uses the reparametrization trick to extend the deterministic policy gradient to a version that can be used by the stochastic policy function π_θ. SAC addresses the sample complexity problem of RL algorithms by utilizing the maximum entropy RL framework where the behavior policy aims to optimize the returns while maximizing the entropy of its actions. The latter is done due to the observation [Haa+18; Zie+08] that adding an entropy maximizing term for the policy $\pi(\mathbf{a}_j \,|\, \mathbf{s}_j)$ to the optimization objective encourages exploration of the state space $\mathcal{S}$, improves the learning rate of the agent, and reduces the relative number of samples that are needed, compared to other standard RL algorithms. In other words, the behavior policy tries to attain optimal behavior by acting as randomly as possible. The maximum entropy objective for N steps is

$$J(\pi) = \sum_{j=0}^{N} \gamma^j \mathbb{E}_{\mathcal{E}_\pi(\mathbf{s}_j, \mathrm{r}_j)} \left[r_j + \alpha J_1(\mathbf{s}_j) \right] \tag{2.63}$$

where $\mathcal{E}_\pi(\mathbf{s}_j, \mathrm{r}_j)$ represents the environment's state-action probability distribution induced by the policy π, α is an optimizable temperature parameter (signifying the importance of exploration in the objective), and $J_1(\mathbf{s}_j)$ is the entropy of the policy function $\pi(\cdot|\,\mathbf{s}_j)$ conditional on the kth state $\mathbf{s}_j$,

$$J_1(\mathbf{s}_j) = -\mathbb{E}_{\pi(x|\,\mathbf{s}_j)} \left[\log(\pi(x|\,\mathbf{s}_j)) \right]. \tag{2.64}$$

The state-action value function also becomes modified to predict the new entropy augmented discounted rewards:

$$Q_\phi(\mathbf{s}_j, \mathbf{a}_j) = \mathbb{E}_{(\mathbf{s}_j, \mathbf{a}_j) \sim \mathcal{E}_\pi} \left[\sum_{j=0}^{\infty} \gamma^j (\mathrm{r}(\mathbf{s}_j, \mathbf{a}_j) + \alpha J_1(\mathbf{s}_j)) \right] \tag{2.65}$$

with the trainable parameters ϕ. It is trained by having its predictions match the estimated $\hat{Q}$ values obtained for sampled replay experience data from $\mathcal{D}$ obtained from a b-length rollout (number of interactions) with the environment $\mathcal{E}$. The actor is trained by minimizing the loss function

$$J'(\pi_\theta) = \mathbb{E}_{\mathcal{E}_{\pi_\theta}(\mathbf{s}_j, \mathbf{a}_j)} \left[\alpha \log \pi_\theta(\mathbf{a}_j \,|\, \mathbf{s}_j) - Q_\phi(\mathbf{s}_j, \mathbf{a}_j) \right], \tag{2.66}$$

which is equivalent to maximizing J in Eq. (2.63). Finally, the temperature parameter α is also automatically updated by maximizing the following objective function:

$$J(\alpha) = \mathbb{E}_{\pi(\mathbf{a}_j \,|\, \mathbf{s}_j)} \left[-\alpha \log \pi(\mathbf{a}_j \,|\, \mathbf{s}_j) - \alpha J_1(\mathbf{s}_j) \right]. \tag{2.67}$$

Again, all the trainable parameters are updated using an exponential moving average w.r.t. target counterparts that partipate in the active part of the algorithm.

Model-based RL

So far, the RL methods that we have described are all *model-free*, i.e., they do not involve the incorporation of any premeditation or planning components on the behalf of the agent. *Planning* in RL refers to the idea of a model of the environment $\mathcal{E}$ that the agent can use to make predictions of the environment in response to any potential actions it might take, including the rewards incurred by taking them. In classical RL literature [SB18a], models for stochastic noisy MDPs are mostly statistical and allow augmented search (w.r.t. model-free RL) through the state-action space as opposed to searching through some plan or function space of models. The latter is often ignored due to their abstract or non-general nature. However, both approaches need not be treated separately. Indeed, in Chapter 6, we show how both approaches can be combined in the QOC setting.

State space model-based reinforcement learning is centrally based around the idea of learning the environment $\mathcal{E}$ using collected experience $\mathcal{D}$ in the form of MDP transitions – in addition to learning the value and policy functions that can be done using the same methods described earlier. The motivation is that the number of environmental interactions or sample complexity to solve the policy optimization problem Eq. (2.51) can be significantly reduced by planning. The procedure is completely statistical. The model that we learn is assumed to be Markovian like the environment following the MDP definition. We denote the estimator of the the environmental Markov transition model P as a non-parametric function given by $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k+1} \,|\, \mathbf{s}_k, \mathbf{a}_k) \equiv \hat{\mathcal{E}}_{\boldsymbol{\zeta}}$. This estimated model $\mathbf{M}_{\boldsymbol{\zeta}}$ can generate probabilities of state outcomes conditional on an initial state and action in order to generate a single sample of the next state.

The description below is kept general to discuss the fundamental issues and advantages of model-based RL.

One of the earliest methods that captures the heart of the statistical model-based RL strategy is Dyna [Sut91]. Dyna samples a state $\mathbf{s}_k$ randomly from the collected experience D of the policy after some exploration. It then takes any number of MDP step using the learned model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k+1} \,|\, \mathbf{s}_k, \mathbf{a}_k)$ and generates model data $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$ that is used to train the policy π_θ. The model is estimated by using supervised learning by function fitting state transitions and/or rewards if a continuous function approximator is used. Instead, in the olden days, just a lookup table also works that caches the transitions $\mathbf{s}_j, \mathbf{a}_j \to \mathbf{s}_{j+1}, r_j$, which can then be used instead of querying the environment if the same $\mathbf{s}_j, \mathbf{a}_j$ pair is encountered.

There are two further meta-strategies when it comes to learning the model based on the question of whether it is better to learn a better model now or optimize the current policy using environment data: (1) *Backwards*: Bellman updates of the value function using simulated experience $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$ where the policy just explores the environment with the original model-free goal; (2) *Forwards*: decision-time planning, e.g., Monte-Carlo Tree Search (MCTS) [SB18a], where selective exploration of the environment is additonally taken by a planning policy in order to improve the existing model through the construction and traversal of a simulation experience tree. This is in addition to the behavior or rollout policy that interacts with the environment after a state not in the model tree is encountered.

Dyna takes the first approach. Experience of the policy collected with the original model-free goal is used to train the model. And some M times for every step taken by π_θ in the real environment $\mathcal{E}$, the policy also simulates its experience using the model. These model interactions are also added to the total experience $\mathcal{D}$ that is used to train the policy.

The biggest problem with model-based RL is that the faithfulness and quality of the learned model degrades as the simulated experience $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$ generated by it is fed into the value and policy updates during policy iteration. Model-based RL methods might be unable to capture a faithful prescription of all environments and thus produce sub-optimal policies. This can become difficult to diagnose if a strong universal function approximator is used. The central issue one faces when using function approximators as opposed to tables for the model is overfitting during the data-limiting stage and underfitting in the data-abundance stage.

Steps towards addressing this problem were made by Ref. [Chu+18] who propose bootstrapping multiple models $\{\mathbf{M}_{\boldsymbol{\zeta}}{}^{(i)}(\mathbf{s}_{k+1} \,|\, \mathbf{s}_k, \mathbf{a}_k)\}_{i=1}^{B}$ and keeping track of epistemic (systematic bias) and aleatoric (stochasticity) uncertainties using $\mathrm{Var}\left(\mathbb{E}_{\hat{\mathbf{P}}(\mathbf{M}_{\boldsymbol{\zeta}})}\right)$

and $\mathbb{E}_{\hat{\mathbf{P}}(\mathbf{M}_{\boldsymbol{\zeta}})}\left[\mathrm{Var}\left(\mathbf{M}_{\boldsymbol{\zeta}}{}^{(i)}\right)\right]$. This approach is further refined using ensembling methods [Buc+18]. Furthermore, Ref. [Jan+19] goes even further by proposing a novel model-based augmentation of SAC called model-based policy optimization (MBPO). It uses PAC (Probably Approximately Correct) generalization bounds on the model and an error bound on the policy's distributional shift during training to propose a generalized k-step Dyna method incorporating the above probabilistic approach in the model construction to make sure the model remains quantifiably faithful to the environment and the quality of the sample it generates are adequate for policy optimization. This approach improves sample complexity compared to past model-based and model-free methods, e.g., PPO [Sch+17; Kha+21; Kha+23b], for certain benchmark problems.

In Chapter 6, we present a modification of MBPO to address the sample complexity problem faced by RL algorithms when specifically solving the QOC problem.

Another problem with model-based RL is the planning resource allocation problem: How much of the policy's exploration budget should be dedicated purely to improve the model? Again, the quantity of interest that needs to be minimized in this case is the model uncertainty and it is theoretically possible to do so optimally using Bayesian adaptive exploration [Duf02; GSD12]. Moreover, the price paid by such methods is that the decision making has high computational complexity and cannot scale to high-dimensional settings.

Model-based RL also has poorer asymptotic performance than model-free RL in the setting where the model is learned from scratch [Moe+23]. This problem can be addressed by incorporating a known model or partially known model which, in contrast, gives model-based RL the asymptotic upper hand. The issue stems from the problem of learned model uncertainty which can only be addressed by consuming more samples from the real environment. Ideally, combining the model-based approach with a partial model and model-free RL can yield the best of both approaches. Indeed, we explore and develop this idea in Chapter 6.

Applications

We now highlight some applications of RL algorithms for quantum control problems.

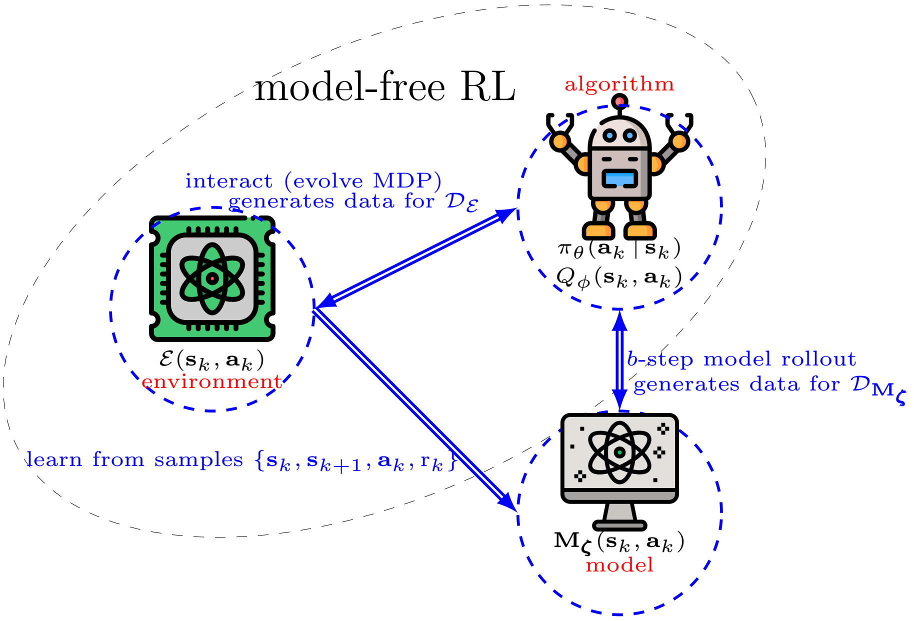

model-free RL
algorithm
interact (evolve MDP)
generates data for $\mathcal{D}_{\mathcal{E}}$
$\pi_\theta(\mathbf{a}_k \mid \mathbf{s}_k)$
$Q_\phi(\mathbf{s}_k, \mathbf{a}_k)$
$\mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$
environment
b-step model rollout
generates data for $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$
learn from samples $\{\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k\}$
$\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_k, \mathbf{a}_k)$
model

Figure 2.1: A schematic of model-based RL. The arrow-head implies direction of affect of the edge between a source and sink node. The agent or policy function π_θ interacts with the RL environment modelled as an MDP to collect data $\{\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k\}$. This encompasses model-free RL. The data is then used to train the model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_k, \mathbf{a}_k)$. The model is trained until some quality measure like the validation prediction error on some untrained-upon data from the environment plateaus indicating that the training is complete. Then, it is used to generate synthetic data through a b-step rollout in which the policy interacts with the model b times. The policy parameters θ (and the state-action value function parameters ϕ) are optimised using the real and model generated data.

Optimal Control with partial or no observations

Optimal control with partial observability in the absence of an accurate model is a regime that is particularly challenging for the dominant model-based, open-loop control approaches. The control problem in this setting is addressed by: (1) dual control theory initiated by Feldbaum in the 1960s [Fel60], where the idea is to balance *controlling* the system with *learning* its dynamics; (2) and RL for optimal control [Ber19]. (2) is the modern interpretation of (1). As mentioned in Chapter 1, in both interpretations, the control problem is reduced to approximate dynamic programming that is solved using Bellman's principle of optimality [Bel52]. We focus on the latter in this thesis since (1) is a lot more theoretical/constrained and is well-studied for specific classical control problems whereas (2) is more generic and data-driven.

The motivation for RL is to find adaptive model-agnostic ways of performing optimization to solve quantum control problems. These methods, in principle, promise to have less overhead compared with functional variation or Pontyragin-variation-based methods for optimal control which use an analytical model and have been the focus of over half a century of fruitful contribution to quantum control, including algorithms such as GRAPE [Kha+05b] and Krotov [RNK12] that utilise gradient-based optimisation of a model-based target functional. Limited knowledge about the system and control Hamiltonians, and interactions with the environment, however, have a strong effect on the performance of such controls. Sampling over uncertain parameters combined with gradient-based optimisation can find robust controls [Don+13].

RL methods are either model-based or model-free, but all methods can in principle be fully model-agnostic. Model-based methods involve creation of a model from scratch, whereas model-free methods skip this step. RL aims to tackle and optimize the trade-off between exploitation and exploration that is the hallmark of dual control. Prior work demonstrated the usefulness of deep RL for quantum optimal control [Che+13] in its application to synthesis of transmon gates [Dal+20a], coherent transport by adiabatic passage through semi-conductor quantum dots [Por+19], and robust two-qubit gmon gate synthesis [Niu+19a]. We show in Chapter 4 that policy gradient RL can successfully solve state preparation QOC problems that are formulated using an MDP with no state observations and noisy reward signals where gradient-based methods fail or consume too many samples.

Towards sample efficient Model-free RL

Recently, Ref. [Siv+22a] proposes making the single call to the control objective function binary for an RL circuit optimization problem of Fock state preparation. The idea is that, under expectation w.r.t. control actions, the full RL reward objective is the true value of the control objective function. This is shown to significantly reduce the sample complexity for the state preparation problem but could be potentially restrictive if the objective function cannot be binarized. Moreover, unlike the spirit of Ref. [Wit+21], knowledge of the physical system in the form of a partially correct model is not leveraged and the RL algorithm solves the optimization problem completely from scratch without the help of this prior knowledge. Instead, in this thesis, in Chapter 6, we approach the RL sample complexity problem from this conventional angle by assuming some partial knowledge about the controllable system where we learn a more succinct description (as far as control is concerned). In other words, an algorithm that lies in between completely model-free RL and complete model-based methods is provided.

2.2.2.5 Model learning for control

One way of reducing sample complexity or the number of interactions with the controllable system in the control problem is to learn an effective model. Ref. [Wit+21] proposes an open-loop differentiable optimization model with model-free calibration to leverage both the physics knowledge in the form of a model and the robustness against noise afforded by being model-free and then updating the model using new data generated by applying the changes. This is motivated by the inherent limitation of pure model based methods in settings where the model is wrong. In Chapter 6, we introduce a similar approach with respect to model learning using automatic differentiation [Pas+19] for time-independent Hamiltonian characterization. However, we learn the complete structure of the system Hamiltonian H_0 instead of scalar coefficients parametrizing fixed physical terms in the Hamiltonian. We also perform a more standard control protocol using complete measurements while their approach uses randomized benchmarking [Kel+14b] that is discussed in Sec. 3.3.

Moreover, gradient-based (automatic differentiation) optimization can be used close to an optimum. In the same spirit, Ref. [Gol+22] proposes a lean model predictive control method using the idea of quantum trajectory optimization using sequential quadratic programming to alleviate the sample complexity problem. This requires a collection of reference quantum states and controls to be attained sequentially which might be problematic in practice if there are too many imposing constraints[14] but the idea is related to classical safe adaptive switching control, where under the overarching goal of stabilizing a plant, one switches off the current controller after acquiring experimental evidence that suggests that desired stability or performance objectives are not being met [SS08]. The relation is in the assumption of feasibility of the next switching controller.

The latter approach does, however, lead to nice bounds on the stability of the control scheme. This idea is taken further in Ref. [Pro+22; HJM19], who propose an efficient solver that optimizes over states and controls while constraining the trajectories to conform to the dynamical evolution equation and various other physical

[14] This is a choice and there can equally be as few constraints as desired, coinciding with a base case where there is only an initial and final time constraint on the control problem.

constraints by projecting the intermediate solutions back on the constraint manifold. Moreover, inspired by model-free RL, GRAPE with measurement feedback [PPM22] allows robust state preparation for the Jaynes-Cummings model with the gradient signal comprised of an ensemble of POVM measurements.

2.3 Summary

In this chapter we have introduced problem of quantum control and the various applications enabled by it in the realm of computation, communication, simulation and metrology. We then presented a glimpse of the range of QOC and RQOC methods used for solving variations of the quantum control problem. We classified these into two broad categories: model-based methods that rely on a theoretical model of the controllable system and model-free methods that do not. For model-based methods, at one end, we have analytical methods such as STIRAP [Kuk+89] and dynamical decoupling [VKL99] that do not require any numerical treatment and intuitively provide the control pulse shape that can be directly translated in an experimental setting. The strength of these techniques is also their weakness, i.e., they are designed to be specific and address only a particular type of control problem: state-transfer in the former and decoherence protection in the latter. Yet, the strong physical motivation behind the construction of these techniques makes their application comparatively robust, at least in principle, in current NISQ devices in contrast to the other methods [Vit+17; Pog+21; Pez+18]. However, as we noted earlier, not all RQOC and QOC problems can be attacked using these methods which require a lot of physical insight and manual effort to glean and it is still an open problem of how the strengths of intuitive approaches can be imparted to the rest of the control methods that we have discussed.

Moving the arrow of governing the creation of the control methods towards more automation, we introduced gradient-based methods that manipulate some parameters of the theoretical model using gradient descent on some optimizable figure-of-merit. This can either be done by manually deriving the derivatives, using the adjoint method or auto-differentiation. The central benefit of this strategy is that the automation is fairly easy to generalize to many RQOC and QOC problems without any dependence on the underlying specificities of the controllable system as opposed to analytical methods. Again, the drawback is the fact that these methods are only as scalable as what classical computation allows and quantum systems of larger Hilbert space sizes remain intractable without further reductions on the modelling requirements asked of the classical machines that are effectively required to simulate quantum dynamics before they can be numerically solved.

An important problem with model-based control methods is their unflinching reliance on the theoretical model. As mentioned in the introductory chapter, current quantum devices are expected to be primitive and fraught with noise of various forms that translates into palpable uncertainties in the theoretical model. Some popular noise effects that mess with the model's predictive dynamics include cross-talk [Kra+19], e.g., the unintended excitation of neighbouring particles during the excitation of the intended particle via a laser. Loss of the model's efficacy when deployed on experimental NISQ devices leads the QOC practitioner to consider model-free methods. We covered methods including simple heuristic gradient descent using Nelder-Mead or Bayesian optimization or reinforcement learning (RL). The price the practitioner must now be willing to pay is generally further compute efficiency relative to model-based methods and, in particular, analytical methods that, otherwise, *have* stability or rough convergence guarantees. Typically, these extra costs are consumed in the *ab initio* construction of a learned model of the controllable system's behavior that is then leveraged by the control algorithm via standard optimization methods. The model of the system can also just be directly learned first with the control effort only applied after a suitable model has been found [Gol+22].

Yet the issue of obtaining the sample complexity efficiency of model-based methods from model-free methods is open especially in the regime where the theoretical model is only partially wrong – that is not unrealistic given the deliberate efforts of physicists to create experimental systems in the likeness of predetermined theoretical models. and also conversely, create effective models that describe a given physical system well. Furthermore, many of the RL formulations of quantum control problems assume unrealistically that the entire quantum state is available to the agent which is a prohibitive assumption due to the associated tomographic costs that are exponential in system size. This thesis addresses both problems. Firstly, towards a realistic MDP formulation, by modelling the QOC problem as a partially observed MDP, a scalable (in system size) method of performing RL control is presented in Chapter 4 that requires no access to quantum observables other than those required to compute an optimizable figure-of-merit, thereby circumventing the need to make expensive quantum measurements. Secondly, towards better sample complexity, by proposing a novel model-based actor-critic RL algorithm in Chapter 6 that incorporate partial knowledge of the controllable system in a learned model of the system whose other parameters are updated using experimental data, we move a step closer towards bridging the gap between model-free and model-based QOC methods.

Chapter 3

Robustness certification methods

In this chapter, we review methods for assessing the general nature of optimal control schemes obtained for the control problems discussed in Sec. 2.1.2. We highlight techniques that can be used to quantify the performance of these schemes in scenarios outside the specific objective function settings in which they are found – in particular, with regards to performance under noise or perturbations from various sources in the physical environments in which these schemes are deployed which we dub *robustness*.

Quantifying the robustness of a control solution to uncertainties in the physical controllable system is important when the system is noisy as this allows for an avenue for us to be able to gauge the performance of a control scheme reliably. The usual figure-of-merit that is used in quantum control is the fidelity of the control scheme that gauges performance w.r.t. a theoretical model. But, as this thesis is concerned with controlling quantum systems that are noisy, fidelity alone is no longer a sufficient measure of performance. Towards that end, we develop a measure of performance in Chapter 5 that incorporates the stochasticity of the controllable system in the figure-of-merit of a control scheme. This chapter covers the ideas that are building material for these robustness measures. At its heart, the goal is to generalize the fidelity in the direction of robustness so that it can be a more faithful figure-of-merit for performance of a control scheme in a noisy quantum system.

3.1 Control landscape topography

In this section, we introduce some methods to roughly visualize the manifold of optimal quantum control solutions. This allows us to characterize quantum controllers based on where they lie on this manifold and further motivate the principle of exploring the local region around the solutions to understand the effect of parameter variations on the optimiality of the solutions, i.e., robustness. In Chapter 5, we use the concept of local fuzzy balls $\mathcal{B}_\sigma$ with radius σ around the solutions as regions of interest within which a robustness measure can quantify noisy performance of a quantum system.

From Sec. 2.1.2, the general form of the quantum control problem formulated as an optimization problem can be abstracted as the problem of maximizing some general fidelity cost functional $\mathcal{F}[\cdot]$ w.r.t. control functions $\mathbf{u}(t)$. Consider the abstract unconstrained optimization problem,

$$\mathcal{F}_{\text{opt}} = \max_{\mathbf{u}(t)} \mathcal{F}[\mathbf{u}(t)] \tag{3.1}$$

where $\mathcal{F}(\mathbf{u}(t))$ represents the map from the space of control functions to the space of real-valued costs or fidelities. This is called the control landscape and its topography has been studied extensively, both numerically and theoretically [RHR05]. The control landscape can be used to establish conditions for existence of optimal schemes and the complexity of finding them which also allows for the concretization of effort needed by optimization algorithms used to find these schemes. Moreover, analysis of the control landscape can be decoupled from the specificities of the underlying generating Hamiltonians [RHR05], under certain ideal assumptions, which allows for the results to be more broadly generalized across various quantum control settings. The necessary condition for an extremum (minimum or maximum) is given by the first-order functional derivative of $\mathcal{F}[\cdot]$ w.r.t. to the control function. The manifold $\mathcal{M}$ including these extrema is given by,

$$\mathcal{M} = \left\{ \mathbf{u}(t) \mid \frac{\delta F[\mathbf{u}(t)]}{\delta \mathbf{u}(t)} = 0, \forall t \in [0, T] \right\}. \tag{3.2}$$

The topology of the control landscape has interesting implications for the nature of optima: whether they are sub-optimal traps or not and if they are smooth in their vicinity, i.e., robust w.r.t. small changes in $\mathbf{u}$.

Controls or 'points' that lie on the critical manifold $\mathcal{M}$ can be sub-optimal or local optima in the sense that a point $\mathbf{u}'(t)$ can yield a fidelity $\mathcal{F}[\mathbf{u}'(t)] \leqslant \mathcal{F}_{\text{opt}}$. Sufficient conditions of optimality can be determined by examining the eigenvalues of the Hessian

$$\mathcal{H}(t, t') = \frac{\delta^2 \mathcal{F}}{\delta \mathbf{u}(t) \delta \mathbf{u}(t')}. \tag{3.3}$$

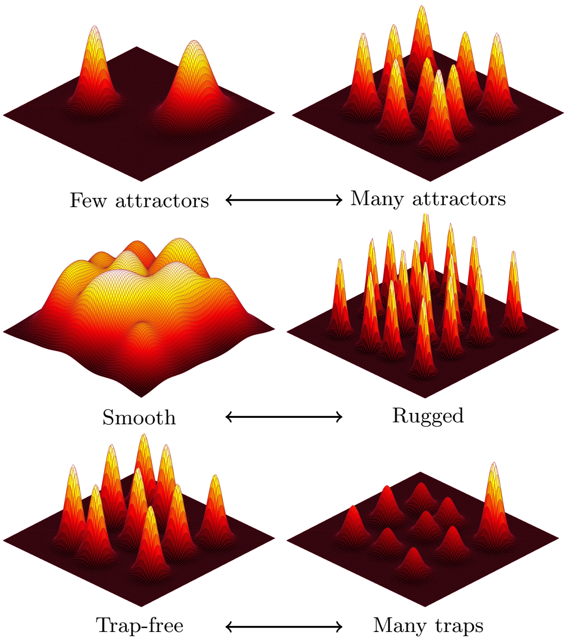


Figure 3.1: (from [DMS22]) Polar ends of different types of expected optima for a generic quantum control problem. The height represents $\mathcal{F}$ and the planar dimensions are two axes representing components of $\mathbf{u}$. The characteristics considered are: top: existence of less vs. many optima; middle: robust/smooth vs. non-robust/rugged; bottom: few vs. many traps.

For the unconstrained optimization objective considered in Eq. (3.1), if all the eigenvalues of $\mathcal{H}$ are positive, i.e., it is positive-definite, then $\mathbf{u}$ is a local minimum. Conversely, if $\mathcal{H}$ is negative-definite, then $\mathbf{u}$ is a local maximum. If $\mathcal{H}$ is indefinite, then $\mathbf{u}$ is a saddle-point.

If the controls are constrained or the system has an insufficient number of control parameters, then the minima or maxima can correspond to local minima or maxima that not optimal (including saddle-points) which in some cases are also called traps since a gradient-based optimization algorithm with constrained resources will likely terminate at this point and will not keep searching for the optimal extremum [MR12; DS13]. Reducing the controllability[1] of the system or increasing the constraints on the control parameters, leads to the reduction in the size of the critical manifold with optima until it is disconnected or isolated and finally completely disappears leading to only traps or saddle-points as the terminating points found by optimization algorithms. The fact that multiple optimal control solutions exist is important to bear in mind, since this allows for the selection of secondary characteristics within these control solutions – in particular we can select for optima that are robust to variation in the control parameters [Bel+11] and other external sources of variation in the optimization objective.

An illustration of some example types of extrema is presented in Figure 3.1 along three characteristic axes: abundance/paucity of optima; abundance/paucity of traps and smoothness/ruggedness of the landscape surface. The numerical trace approximation of the Hessian in Eq. (3.3) has been used to diagnose the smoothness of the numerical control landscape for many-body spin chains [DMS22]: a lower value implies a slower rate of change of the gradient of $\mathcal{F}$ which implies that the landscape does not vary w.r.t. the control parameters $\mathbf{u}$ [Rab+06]. Ideally, for robustness w.r.t. control parameter variation, it is desirable that for a large variation in the controls $\mathbf{u} + \delta\mathbf{u}$, we maintain a fidelity close to the optimum: $\mathcal{F}(\mathbf{u} + \delta\mathbf{u}) \approx \mathcal{F}(\mathbf{u}) \approx \mathcal{F}_{\text{opt}}$.

3.2 Measures from classical robust control theory

Classical control theory has been around for much longer than quantum control theory. The shift to classical robust control began in the 1960s [Saf12] when the inadequacy of optimal control was perceived and stability in optimality, i.e., robustness,

[1] A system is controllable iff there exists a dynamical trajectory induced by some $\mathbf{u}(t)$ between any two possible initial and final states of the quantum system.

was sought. Here, we introduce some ideas of robust control that are particularly interesting from a quantum control point of view but are unable to be directly transferable to the quantum regime. This is chiefly because the classical setting is more concerned with asymptotic stability w.r.t. time whereas quantum control is limited by the coherence time of the controllable system. Nevertheless, in this thesis, where possible, we try to connect with the classical robust control literature to ensure that RQOC benefits from the strong rigorous tradition and results discovered in its classical counterpart's backyard. In particular, we connect the novel quantum robustness measure developed in Chapter 5 with classical ideas of robustness.

In a nutshell, classical robust control is mainly concerned with ensuring that the dynamical output signal of a controllable system, called the *plant*, matches that of some desired signal, called the *reference*, using a *controller* to guide the minimimzation of the *error* signal between the output w.r.t. the reference by varying some parameters of the plant. There are usually some sources of noise in various steps of the control procedure, e.g., output noise and controller noise. There are essentially two paradigms of control: open and closed loop. The main distinction between the two is that in the open loop case, the controller does not have access to a real-time feedback signal of the error from the plant and needs to construct the entire control signal that minimizes the error a priori. Usually, this error minimization is done using a theoretical model of the plant's behavior. For the closed-loop case, the error signal minimization occurs in a feedback loop where the controller proposes some changes in the plant's input parameters and then is allowed to access the new error signal in real-time. Feedback is unnecessary if the theoretical model is correct [ZD98]. The main goal of classical robust control is the analysis and synthesis of controllers that can ensure that the plant behaves optimally and that the controller is *stable* under the presence of model uncertainties, disturbances and sources of admissible noise [ZD98].

Interestingly, most robustness certification tools in classical control have been developed for simplified linear time-invariant (LTI) systems[2] that are feedback controlled in a closed-loop. Although there has been work done to extend these ideas to more realistic settings [Dor87], e.g., those involving aircraft/structure vibration control [CS91], translating these tools, generally, to quantum control settings is still an open

[2] Linearity implies that the system output $f(x,t)$ is linear w.r.t. the input x i.e. if $x \to ax' + bx''$, then $f(x,t) = af(x',t) + bf(x'',t)$. Time-invariance is likewise a property of the system output being unaffected by time translations, i.e., $f(x, t-T) = f(x,t)$ for some time shift T

problem [Wei+22]. This is chiefly because quantum control is most likely to be formulated as an open-loop problem with a bilinear system since among other feedback sources [SJL18], general measurement of the quantum system disturbs its underlying dynamics.

Moreover, as opposed to steady-state or asymptotic (long-time) behavior, quantum control is generally more interested in transient dynamics of the system due to its finite coherence times where the system's quantumness is still operational [DP10a]. The unitary dynamics cannot be stable and so robustness in the classical sense (in the frequency domain) is hard to define; decoherence to a steady state makes it stable, but not quantum.

Nonetheless, specific quantum control problems can be recast in the LTI formalism and can facilitate the application, with some modification, of classical measures of robustness that can be used to quantify the performance of the controller w.r.t. system uncertainties [DP10a]. In short, for the time-independent quantum control setting, the density matrix ρ can be recast as a real vector $\mathbf{r}$ using the Bloch formalism [RBR16]. Likewise, the dynamical generator (for open or closed systems) can be represented as a superoperator matrix $\mathbf{A}$ in the Pauli basis $\{\sigma_k\}$ as follows,

$$r_k(t) = \mathrm{Tr}(\rho(t)\sigma_k) \tag{3.4a}$$

$$A_{k\ell} = \mathrm{Tr}\left(\frac{i}{\hbar}H[\sigma_k, \sigma_\ell]\right) \tag{3.4b}$$

with the system following the dynamics of the form

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{A}\mathbf{r}(t). \tag{3.5}$$

Using this formalism, the structured singular value μ [Doy82] has been applied to the problem of entanglement creation between two coupled qubits in a lossy cavity [Sch+22b]. Here the controller's goal is the stabilization of an entangled two-qubit steady-state. A stability margin, $\mathfrak{G}$ w.r.t. structured perturbations $\delta_k \mathbf{S}_k$ to $\mathbf{A}$ [Wei+22] is given by,

$$\mathfrak{G} = \left| \max_{\lambda_n \neq 0} [\lambda_n(\mathbf{A} + \delta_k \mathbf{S}_k)] \right| \tag{3.6}$$

where $\lambda_n(X)$ is the eigenvalue of X where δ_k is a noise scale parameter. $\mathfrak{G}$ quantifies the region of stability of the system w.r.t. $\mathbf{S}_k$. This is inspired by the classical stability margin which essentially involves the idea of upper/lower bounding the domain of uncertainties within which the plant is stable under some controller. Strong theoretical statements, e.g., bounds, such as those imposed by the small-gain theorem

[Zam63] can be made about the domain of stability of the plant w.r.t. uncertainties using μ by exploiting the LTI assumptions to make the mathematics tractable.

Another classical feedback-control robustness measure in the LTI picture is the *log-sensitivity*, the dimensionless differential sensitivity of the plant's output w.r.t. uncertainties[3] [DB11]. For the entanglement generation problem as before as well as the time-independent state-transfer problem, the log-sensitivity s can be defined in terms of the infidelity error $e = 1 - \mathcal{F}$ as follows:

$$s(\mathbf{S}_k, T) = \frac{1}{e(T)} \frac{\partial e(T; \mathbf{S}_k, \delta)}{\partial \delta}\bigg|_{\delta=0} \tag{3.7}$$

where T is the final time and $\delta \in [0, 1]$ is a noise scale parameter. This is a measure of the sensitivity of the system's performance to perturbations $\mathbf{S}_k$ and is ideally small for robust controllers but it also diverges when $e(T) \to 0$ [ONe+22a]. The log-sensitivity or the stability margin versus the steady-state entanglement measure called concurrence exhibit a trade-off due to decoherence [Sch+22b] wherein low entanglement implies higher stability and vice versa. This is expected in classical control where performance trades off with stability [DB11; SLH81].

3.3 Randomized benchmarking

Measuring the performance of a control scheme in the presence of uncertainties in the controllable system can be reformulated to the problem of performance under some randomization of the system's parameters due to noise. We utilize this idea to quantify performance or robustness of the control scheme on a noisy system whose model is uncertain in Chapter 5. However, this idea is not new and is well-captured by the technique of randomized benchmarking that is popular in the quantum technologies community as a measure of quantum circuit performance.

Randomized benchmarking [EAŻ05; MGE11; Kni+08] can be used to characterize the noise or error per gate in a quantum circuit in a manner that is robust to state preparation and measurement errors and robust to the contextual position of the gate in the circuit and only polynomial in the size of the gate with costs scaling as $\mathcal{O}(\text{poly}(n))$ for an n-qubit gate. It is a scalable stochastic way to estimate the operational error of a set of quantum gates as opposed to more demanding tomographic procedures that scale exponentially in n. It works by applying a number of randomized sequences

[3] for a plant P and the uncertain parameter S, the log sensitivity is $\frac{\partial P/P}{\partial S/S}$

of Clifford gates[4] and then modelling the resulting average fidelity obtained over the sequence as a depolarizing noise that can be fit with an exponential decay model. If the gates are perfect, these randomized circuits effectively do nothing. Otherwise, in the presence of imperfections, there is an exponential decay of the sequence fidelity with the length or depth of the circuit. We provide a brief overview of the protocol in Ref. [MGE11] and discuss some applications to QOC.

Randomized benchmarking consists of the following steps:

1. Create a sequence of m operations chosen uniformly at random from the Clifford group $\bigcirc_{j=1}^{m} C_{i_j}$. Then choose the final operation from the Clifford group that cancels this sequence of operations such that $C_{i_{m+1}} \circ \bigcirc_{j=1}^{m} C_{i_j} = \mathbb{1}$. Assume the noise in the each Clifford operation is represented by Λ_{i_j} that independently depends on the timestep j and that the noise correlation timescale is smaller than the Clifford gate action timescale. The entire randomized gate sequence is given by,
$$\mathcal{S}_{\mathbf{i}_m} = \bigcirc_{j=1}^{m+1} \Lambda_{i_j} \circ C_{i_j} \tag{3.8}$$
where we represent the i-th Clifford gate sequence by the tuple $\mathbf{i}_m = (i_1, \ldots, i_m)$.

2. For each Clifford gate sequence $\mathbf{i}_m$, compute the survival probability given by the observable $\mathrm{Tr}\left[E\mathcal{S}_{\mathbf{i}_m}(\rho)\right]$ where E is the noisy measurement observable and ρ is the initial state and ideally $E = \rho$.

3. Now average over the different $\mathbf{i}_m$ sequences to obtain the sequence fidelity,
$$\mathcal{F}_{\text{seq}}(m) = \frac{1}{|\{\mathbf{i}_m\}|} \sum_{\mathbf{i}_m}^{|\{\mathbf{i}_m\}|} \mathrm{Tr}\left[E\mathcal{S}_{\mathbf{i}_m}(\rho)\right]. \tag{3.9}$$

4. Finally, fit the sequence fidelity to a noise model such as the following,
$$\mathcal{F}_{\text{seq}}(m) = Ap^m + B + C(m-1)(q-p^2)p^{m-2} \tag{3.10}$$
where A, B, C absorb the state preparation and measurement errors, and the final term involving $q - p^2$ absorbs some weak gate dependence in the noise.

[4] elements of the Clifford group that map tensored Pauli operators to tensored Pauli operators with the generators: Hadamard, S and the CNOT [Got97]

5. Obtain the average error rate given by $r = 1 - p - (1-p)/2^n$. Note that the r is also the average gate infidelity given by $r = 1 - \int d\rho \, \mathrm{Tr}[\rho\Lambda(\rho)]$ where Λ encapsulates the noise. The action of the Clifford group essentially converts any noise into a depolarizing map such that $\Lambda(\rho) = p\rho + (1-p)/2^n\mathbb{1}$ and taking the trace recovers the average error rate stated earlier.

Randomized benchmarking is one of the most commonly used methods for experimentally characterizing gate noise [Xia+15; Bar+14; Che+16] for one-qubit and two-qubit Clifford gates and/or circuits. However, the correspondence between r and the average gate infidelity diverges when the errors in the gates are not small and gate dependent, differing sometimes by orders of magnitude [Pro+17], since the worst case error bound, independent of noise information, scales as the square root of the average gate infidelity [WF14]. Randomized benchmarking has been used alongside QOC in experimental settings to cheaply and quickly infer and thereby reduce one- and two-qubit gate infidelities, circuit fidelities and gate-bleedthrough – where error in previous gates propagates to many subsequent operations – in a five-qubit superconducting transmon [Kel+14c]. The idea is to to map the randomized benchmarking errors onto gate control pulse parameters that can be optimized using Nelder-Mead or manually engineering pulses. A variant of this technique called purity benchmarking [Wal+15] has also been used to distinguish coherent and incoherent errors in single-qubit gates which are then, using QOC methods, reduced via hardware optimization and optimal pulse preparation [Par+16]. Moreover, it has also been used for characterising leakage error rates out of a two dimensional decoherence-free computational subspace of QEC codes [WBE16].

3.4 Summary

In this chapter, we have introduced three distinct ideas for robustness certification of quantum controllers. The characterization of optima on the control landscape provides a visually appealing view of the problem of robustness ('flat peaks') that is viewed through mathematical angles with classical robustness measures such as the derivative of the fidelity error in the log-sensitivity or the concept of averaging Clifford gates in the sequence fidelity for error rate extraction in quantum circuits. All these views are essential in understanding performance and robustness of a quantum control scheme in the presence of an imperfect model or a noisy controllable system. Yet, the control landscape only offers a qualitative or analytical perspective on the robustness of an optimal control scheme without any direct or easy-to-optimize target to find robust control solutions or a cheap target to evaluate solutions that have already been found for their robustness to model uncertainties. Furthermore, classical robustness measures are not easy to translate to quantum control due to the intrinsic mismatch in the desired objectives in quantum and classical control: the former desires transient robustness while the latter desires long-time stability. Lastly, it remains to extend the averaging treatment in the sequence fidelity computation to assess individual gate robustness independent of the circuit setting where a large composition of gates needs to exist at the same time for the gate robustness to be quantified.

A contribution of this thesis is to address each of the shortcomings in the aforementioned individual procedures by proposing a novel robustness infidelity measure (RIM) for robustness certification in Chapter 5. Using theoretical foundations, we argue that one instantiation of the RIM, the average infidelity, is a robustness measure. This measure captures both the typical fidelity or performance figure-of-merit with a robustness figure-of-merit as one scalar value. Firstly, we quantify robustness with the local structure of the control landscape (i.e., topographic differences in the neighbourhood of optima) using a fuzzy ball centered at the optimal solution and by sampling various points within the ball using Monte Carlo noise. The noise strength scales with the radius of the ball. This yields a cheap numerical method to assess robustness of quantum control solutions with the noise representing or being actual physical uncertainty in the controllable system. Secondly, we connect the RIM with the log-sensitivity and comment on the utility of the classical interpretation of the measure which is again favourably viewed in a control landscape perspective. Finally, our measure, while also based on the idea of randomization and averaging over the noise terms like randomized benchmarking, is not limited to circuits. We extend the protocol to the setting where performance of individual gates, state prepartion or any other figure-of-merit for an RQOC task needs to be generalized, in the robustness sense, in a simple way.

Part II

Results and Discussion

The second part of the thesis dives into detailed technical results and expands the thesis contributions that were discussed in passing in the first part.

Chapter 4

Benchmarking control algorithms on spin chains

As mentioned in Chapter 1 and Chapter 2, although early-stage NISQ devices are expected to be error-prone and limited in size, they could pave the way to revolutionize computation and simulation at a fundamental level. They have already proven to be effective tools in physically simulating molecular networks [BDN12; GB17; Blo18]. Currently, one of the main challenges for NISQ devices is robustness to known and unknown uncertainties. Towards that end, in this Chapter, we employ policy gradient RL algorithms that were covered in Chapter 2 to find robust quantum controls with a fully model-agnostic approach using single shot measurements, which can be collected experimentally. Moreover, another challenge faced when controlling NISQ devices is that model-based control methods scale exponentially in classical computational resources as the size of the quantum system increases. We show that our approach is scalable and in that sense similar to variational quantum algorithm (VQA) approaches [Per+14; Buk+18] where the main problem lies in the exploration of a parameter space growing exponentially in the size of the system. Note that the model-based simulation requirements of the controllable system are dropped in this case.

We demonstrate that the RL agent constructs an effective *ab initio* model of the noisy unknown controllable system while incorporating inherent stochasticity in the representation of the system.

Instead of passing unitary operators or density matrices to the RL agent, as considered in previous work [Buk+18], we only give the agent access to experimentally observed data and control parameters. This is in line with real world scenarios where

RL may be deployed in an experimental setting with high levels of uncertainty, commonly seen in current setups. In Sec. 4.2.1, we provide a computational resource comparison between policy-gradient-based RL algorithms and motivate our choice of PPO (Proximal Policy Optimisation) as a representative on-policy RL algorithm for quantum control purposes that is used later in Chapter 5. In Sec. 4.2.3, we compare the cost of using PPO with that of L-BFGS in solving a spin transfer problem with increasing noise and show that the latter's cost increases considerably in contrast to its noise-free cost. This demonstrates the resilience of RL in finding optimal controllers to measurement and Hamiltonian noise, where analytical methods break down or consume too many resources. Even though analytical model optimization has an advantage over RL when the model describes the physical system well, as no exploration is required, increasing uncertainties in the model, however, require an RL or exploratory approach. Moreover, although L-BFGS is more likely to find high-fidelity controllers, preliminary robustness analysis in Sec. 4.2.4 for the controllers found by RL and L-BFGS suggests that RL controllers may be more robust to noise than those found by L-BFGS.

4.1 Preliminaries

4.1.1 Information Transfer in the Single Excitation Subspace of XX Spin Chains

We consider a network of M spins represented by the quantum Heisenberg model given by the Hamiltonian

$$\frac{H_{\text{heis}}}{\hbar} = \sum_{a\in\{x,y,z\}} \sum_{j=1}^{N} J^a \sigma_j^a \sigma_{j+1}^a + \eta \sum_{j=1}^{M} \sigma_j^z \tag{4.1}$$

where $\sigma_j^a = \mathbb{I}^{\otimes j-1} \otimes \sigma^a \otimes \mathbb{I}^{\otimes M-j}$ and σ^a are the usual Pauli matrices. We set $J^z = 0$ and $J^x = J^y = J$ for the XX model with uniform couplings. This model has been studied extensively, starting with Ref. [LSM61] in 1961, and a more recent review of the system, as it relates to quantum communication, is provided in Ref. [Bos07]. Conditions for perfect state transfer along XX chains were derived in Ref. [Chr+04] and applied to NMR systems [Zha+05]. Similar experiments have been carried out in photonic systems [BNT12; Per+13], and proposals for engineering similar systems with trapped ions [GL14] and cold atoms [BMF21] exist.

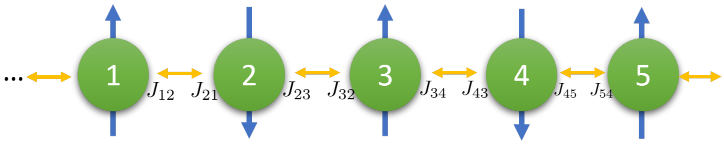


Figure 4.1: An illustration of a spin chain of length $M = 5$ prescribed by the H_{XX} Hamiltonian.

The state space naturally decomposes into non-interacting excitation subspaces as the Hamiltonian commutes with the total excitation operator. Here we consider the first excitation subspace, the smallest space that enables transfer of one bit of information between the nodes in the network. Higher excitation subspaces may be needed for other applications, but it is desirable for information transfer to limit oneself to the smallest space that is sufficient to achieve the task. This is a much smaller space and only grows as $O(M^2)$ as opposed to $O(\exp(2M))$, which comes from the exponential operator growth of the Hilbert space in the number of qubits M, which makes the control problem computationally tractable for large M. The Hamiltonian of the first excitation subspace is

$$\frac{(H_{XX})_{l,m}}{\hbar} = J\delta_{l,m\pm1} + \Delta_l \delta_{l,m} \tag{4.2}$$

where $\delta_{l,m}$ is the Kronecker delta. The static controls are local energy biases Δ_l on spin $|l\rangle$ in a diagonal matrix $H_{\boldsymbol{\Delta}} = diag(\Delta_1, \ldots, \Delta_M)$. An example of a spin chain of length $M = 5$ is shown in Fig. 4.1.

H_{XX} allows for transfer of single bit excitations from an initial spin state $|a\rangle$ to a final state $|b\rangle$.

Our control problem is of the static state preparation[1] kind defined in Section 2.1.2.1. We specifically consider transitions between one-hot encoding state vectors (canonical Euclidean basis vectors), consistent with a single bit propagating through the network by moving an excitation from one spin state to another. Recall that the solution to Eq. (2.5) is a final time t_{opt} and a single vector of M biases $\boldsymbol{\Delta}_{\text{opt}}$ for the optimal controls $\mathbf{u}_{\text{opt}}$. This is a static or time-independent version of the more general dynamic control problem where the control function $\mathbf{u}(t) = \boldsymbol{\Delta}(t)$ is time-dependent.

The most common paradigm for quantum control is dynamic [Gla+15; DP10b]. The implementation of time-dependent control functions typically requires the ability to rapidly modulate or switch controllers implemented by physical fields (e.g., lasers or

[1] or more specifically state transfer

magnetic fields). An alternative to dynamic control is time-invariant control, i.e., time-independent control parameters Δ_n [LSJ15b]. This is analogous to shaping the potential landscape to facilitate the flow of information from an initial state to the target state. For example, information encoded in electron or nuclear spins in quantum dots whose potential can be controlled by varying voltages applied to surface control electrodes, creating a potential landscape. The static control problem has fewer parameters, and so is in some sense simpler. Moreover, previous work has found evidence concerning good robustness properties of the static controls [SJL17]. They may also be simpler to implement experimentally as we do not need to modulate control fields, or could be part of a more complex dynamic control scheme. However, the optimisation landscape is challenging [LSJ15b], and there is no guarantee that the controllers found are robust with respect to uncertainties in the system and interactions with the environment.

The perturbations are given by,

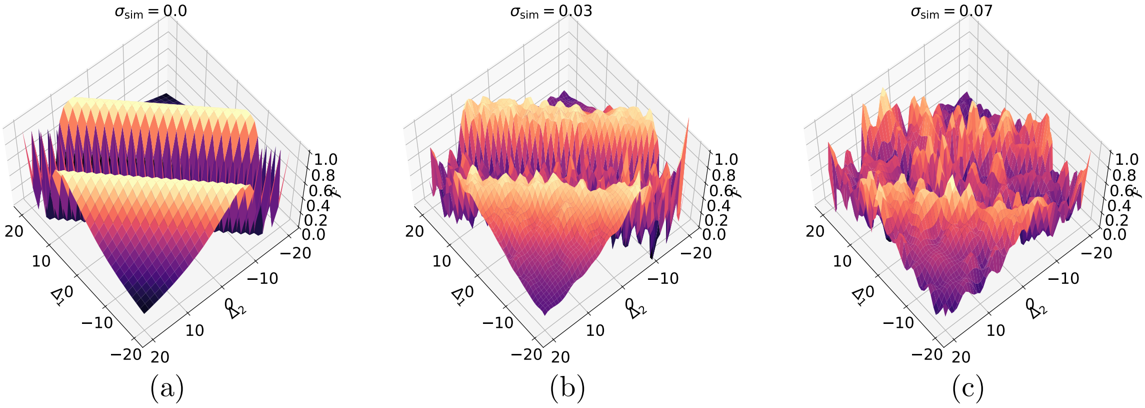
$$(S_\sigma)_{l,m} = \sum_{k=1}^{M-1} \gamma_k^J J \delta_{l,k} \delta_{l,m\pm 1} + \sum_{c=1}^{M} \gamma_c^C \Delta_c \delta_{c,l} \delta_{l,m} \tag{4.3}$$

where γ_k^J and γ_c^C are the strengths of the perturbation on the couplings and controls respectively. We draw these strengths from the same normal distribution $\mathcal{N}(0, \sigma^2)$ with mean 0 and variance σ^2. An example of a perturbation $S_{\sigma_{\text{sim}}}$ with varying noise scale parameter $\sigma_{\text{sim}} = 0, 0.03, 0.07$ and its effect on the fidelity landscape Eq. (3.1) for the end-to-end state transfer from for a spin chain with $M = 3$ is illustrated in Fig. 4.2.

Depending on the hardware platform, it is possible to consider specific practically motivated correlated noise models with correlated perturbations or a power law decaying electric-field noise ($1/s$), e.g., in trapped atomic platforms [Cet+20; Bro+15]. We have chosen to implement the simplest option of equal strength random perturbations on all non-zero entries of the Hamiltonian that is also relevant in practical settings [BMF21; BNT12; Per+13; GL14; Zha+05].

4.1.2 Existence of multiple control solutions

The quantum control problem is known to have multiple optimal control solutions [BCR10]. This means that, in practice, one has the choice between multiple potential control solutions to pick for a given control problem and, for example, deploy

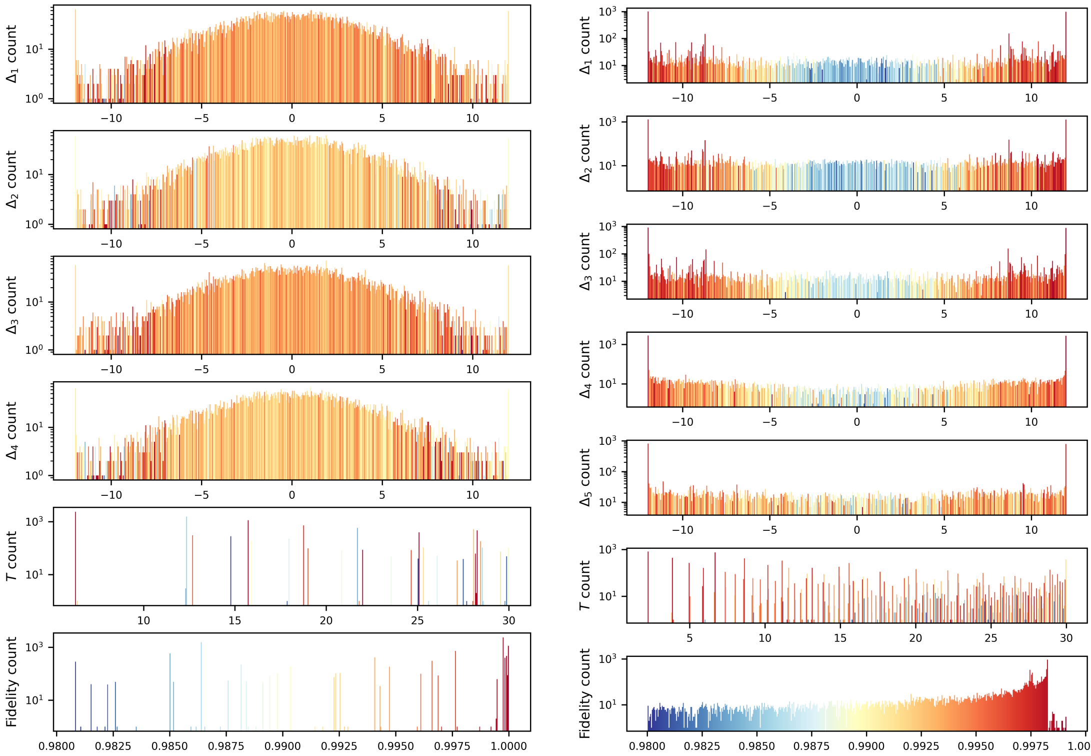


(a) (b) (c)

Figure 4.2: An illustration of the effect of perturbations of varying noise strengths on the fidelity. We use the Hamiltonian H_{XX} in Eq. (4.2) for a spin chain of length $M = 3$ with $\Delta_1 = \Delta_3$ with the state transition being end-to-end in the control problem Eq. (2.5). We show the fidelty $\mathcal{F}$ landscape (z-axis) as a function of two control parameter x,y-axes for Δ_1, Δ_2. One unstructured Hamiltonian perturbation (Eq. (4.3)) with the simulation noise strength parameter, given by σ_{sim}, is applied. As σ_{sim} increases, the landscape is more disordered and smoother in regions where the spikes of high fidelity are located. Consequently, the narrow high-fidelity peaks are washed out and no longer effective optima.

only a subset of theoretically obtained solutions on a real physical system. The existence of extra control solutions also allows us to apply a selection criterion on the multiple numerically obtained control solutions w.r.t. some other desirable attribute such as small final times to make short optimal pulses or pulses that are robust to uncertainties in the parameters of the physical system. It also means that there are extra dimensions apart from numerical optimality that can and should be explored to distinguish controllers. This is a direction that is explored throughout this thesis, especially with regards to robustness.

We illustrate the redundancies in optimal controllers for two state spin transfer problems in Fig. 4.3. We collect 10,000 optimal controllers using L-BFGS with different random seeds and bin each scalar value of the biases Δ_n separately in 200 discrete intervals w.r.t. the numerical values of their magnitude. Each bin is also colored by the average fidelity obtained by all controllers that fall within it. This highlights the structure in the control values and notably the sparsity in the time values for which optimal controls are found. Note that the bins do not preserve order of a single control parameter vector so this illustration which is an attempt to visualize the control landscape in higher dimensions is still qualitative.



(a)

(b)

Figure 4.3: (a) Histograms of 10, 000 optimal control parameters $\{\mathbf{\Delta}_{\text{opt}}, t_{\text{opt}}\}$ for the end-to-end state transfer problem for a spin chain of length $M = 4$ (b) Similar histogram for the end-to-middle state transfer problem for a spin chain of length $M = 5$. The histogram bins are colored by the corresponding average fidelity value of all the controllers that fall within an interval. The first M plots reflect the histogram of values of the control biases Δ_n and the final two plots show the histogram of final times and the fidelity values obtained. The histograms show that there are multiple optimal controllers for a given control problem. The sparsity of the time values for which optimal controllers are found is also evident.

4.1.3 Formulating the model agnostic RL control paradigm

For the state transfer control problem Eq. (2.5), we define the model agnostic MDP in the following manner. Firstly, we denote the discretization of the MDP in terms of the iteration timestep[2] k. The RL formulation of the control problem is as follows:

$$\mathbf{a}_k = \{\delta\Delta_{k-1}, \delta t_{k-1}\}, \tag{4.4a}$$

$$\mathbf{s}_k = \prod_{l=1}^{k} \exp\left(-\frac{i}{\hbar}\Delta t \mathbf{G}(t_l, u_l)\right), \tag{4.4b}$$

$$\mathrm{r}_k = \mathcal{F}(|\psi_{k-1}\rangle, |\psi^*\rangle) \tag{4.4c}$$

[2] that is not to be confused by the physical readout time parameter t

where $\mathbf{a}_k$ is the action, $\mathbf{s}_k$ is the state and r_k is the reward and δ denotes changes to the previous respective values. Note particularly that $\mathbf{a}_k$ is the change in the control parameters changing $\mathbf{s}_k$ by the given values and that $t_{k-1} = T$ is the time for which the Hamiltonian is evolved. The readout time t_{k-1} with the Δ_{k-1} are the control parameters for π to change such that the reward is improved. Here t_{k-1} is the physical readout time, a control parameter. Note that this means π is a control landscape exploration strategy with the aim to find control parameters that achieve the physical state transition from $|\psi(t_0)\rangle$ to $|\psi^*\rangle$ that maximizes $\mathcal{F}(\mathbf{\Delta}, T)$. So the goal, rather than the path to get there, is important, even if of course a shorter path makes finding the goal more efficient. We construct an environment $\mathcal{E}$ that a differentiable policy π_θ can interact with to obtain $(\mathbf{s}_k, \mathbf{a}_k, \mathrm{r}_k)$. The state vector satisfies $\mathbf{s}_k = \mathbf{s}_k \mod \mathbf{s}_{\text{limit}}$ and we set the the limit $\mathbf{s}_{\text{limit}}$ to be ±10 for Δ_{k-1} and 30 for t_{k-1} to ensure that the control parameters are physical and realisable in experiments. A reward threshold, e.g., 0.99, is set as a convergence criterion yielding a single solution vector $\mathbf{s}_k^*$, effectively reducing the problem to optimal time-independent Hamiltonian searching. The RL optimization procedure is run for some number of epochs until the reward threshold is achieved. Each epoch consists of a fixed number of timesteps of exploring the landscape from an initial random position. The policy parameters θ and the Q function are updated via backpropagation every epoch.

The utility of the fact that RL assumes nothing about the analytical form of the model should become apparent if the environment $\mathcal{E}$ is stochastic as that analytical form of the noise is unknown and is approximately learned – without a priori structural assumptions – via interaction with the noisy system. To test this hypothesis, we consider two noise models: (1) directly augmenting H_{ss} with perturbations of the form given by Eq. (4.3). This simulates noisy or tunably inaccurate physics, e.g., due to leakage of spin couplings. (2) coarse-graining the fidelity r_k to simulate single-shot or inaccurate measurements by replacing it with $\tilde{\mathrm{r}}_k \sim \mathrm{Bin}(M, \mathrm{r}_k)$, drawn from a binomial distribution where M is the number of measurements made and r_k, the true fidelity, is the binomial probability and $\tilde{\mathrm{r}}_k$ represents the average single shot measurements to estimate the fidelity probabilities. In this thesis, the choice of the noise models is motivated purely by generality and simplicity to study control in a learning framework. In the absence of a concrete physical system, we assume all parameters are equally uncertain. For both (1) and (2), correlated noise of a random functional form that actually takes into account the physical characteristics of the quantum architecture is also possible and is worth exploring in the context of a particular physical system. Dephasing and decoherence errors that are characteristic of quantum processes are possible to explore under the Sudarshan-Lindbladian evolution of the density matrix [BP+02] are considered in Chapter 6.

We only consider leakage within the nearest neighbour spins. Another possible source of noise could be leakage to the next nearest neighbours due to cross-couplings between spins in transmon systems or finite laser beam sizes in cold atom or ion systems. For the purposes of this thesis, however, we neglect next-nearest neighbor coupling as it is negligible or can typically be mitigated in practical systems. We have also made the implementation of actions $\mathbf{a}_k$ noisy by perturbing the diagonal of H_{ss}, but we could have also coarse-grained the actions to account for the finite resolution of the magnetic or laser field that actually implements the controls in a real experiment. Both are equivalent in terms of their final effect.

4.2 Benchmarking experiments

In this section, we attempt to quantify the sample complexity cost or environment calls $\mathcal{E}$ of different RL algorithms in comparison with L-BFGS. We also conduct a qualitative comparison of robustness of controllers found by RL and L-BFGS w.r.t. Hamiltonian perturbations defined in Eq. (4.3) via a Monte Carlo Robustness Analysis (MCRA). The scheme is outlined in Fig. 4.4. We collect $C = 100$ controllers with a fidelity of at least 0.99 and consider 100 Monte Carlo perturbations per controller.

4.2.1 Cost of Reinforcement Learning Algorithms

We first analyse the cost of the policy gradient algorithms from Chapter 2. The costs are expressed as the number of environment $\mathcal{E}$ calls, corresponding to estimating the fidelity via single-shot measurements, for a run that successfully terminates at a fidelity threshold. This links performance to experimental costs and makes different algorithms comparable without resorting to timing or iteration counts.

We choose to study a noisy transition $|0\rangle \rightarrow |2\rangle$ for chains of length $M = 3, \ldots, 7$. We use 100 single-shot fidelity measurements to estimate the fidelity of a controller and a Hamiltonian perturbation noise of $\sigma_{\text{noise}} = 0.05$. The *"perceived"* fidelity is the stochastically produced by the noisy environment, as observed from noisy measurements. We compare it to the *"true"* fidelity of the controller without noise. A perceived fidelity threshold of 0.99 is set as termination criterion. Fig. 4.5 shows the

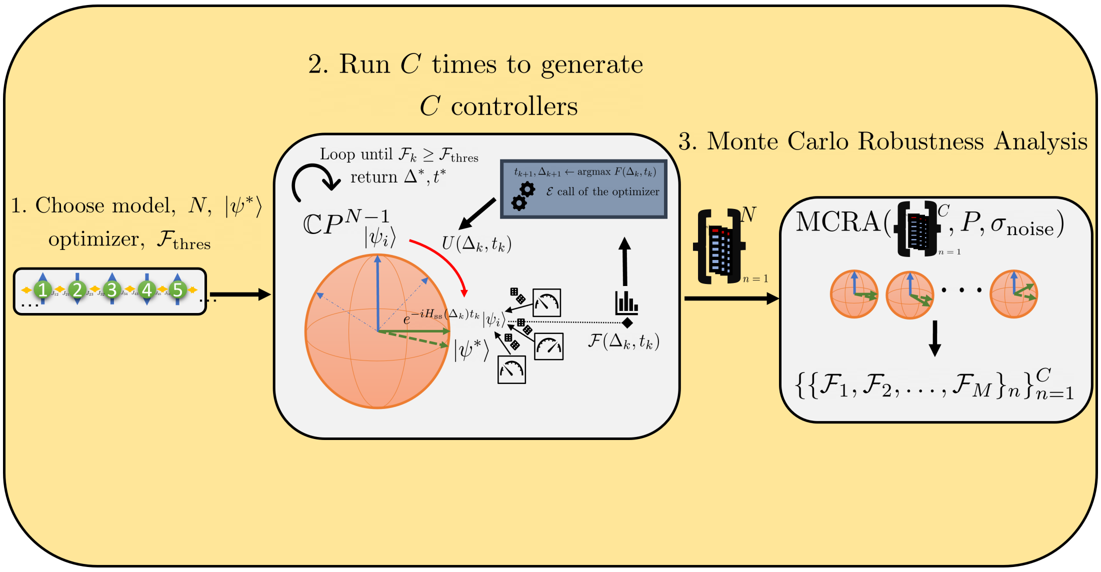

2. Run C times to generate C controllers
1. Choose model, N, $|\psi^*\rangle$ optimizer, $\mathcal{F}_{\text{thres}}$
1
2
3
4
5
Loop until $\mathcal{F}_k \geq \mathcal{F}_{\text{thres}}$ return Δ^*, t^*
$t_{k+1}, \Delta_{k+1} \leftarrow \text{argmax } F(\Delta_k, t_k)$
$\mathcal{E}$ call of the optimizer
$\mathbb{C}P^{N-1}$
$|\psi_i\rangle$
$U(\Delta_k, t_k)$
$e^{-iH_{\text{ss}}(\Delta_k)t_k}|\psi_i\rangle$
$|\psi^*\rangle$
$\mathcal{F}(\Delta_k, t_k)$
3. Monte Carlo Robustness Analysis
N
$n=1$
MCRA(C $n=1$, P, σ_{noise})
$\{\{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_M\}_n\}_{n=1}^C$

Figure 4.4: A general outline of the numerical experiment to collect C controllers and conduct an MCRA to qualitatively evaluate comparative control algorithm robustness.

median performance of DDPG, PPO and TD3 over 50 runs. In terms of environment calls, DDPG performs significantly worse compared to PPO and TD3, but it is more difficult to decide between the latter two.

TRPO and REINFORCE were excluded from the study as sufficient statistics could not be obtained. Their behaviour was highly variable and inconsistent due to a lack of successful termination which prevented further analysis. For REINFORCE, we suspect that this was because of the absence of a replay buffer to sample a sufficient variation of transitions and a value/Q function that maps actions to expected rewards to ground policy parameter updates. Similarly, TRPO, while successful in achieving fidelities > 0.99 on complicated transitions such as $|0\rangle \to |3\rangle$ for $M = 7$, was algorithmically complex (e.g., the Hessian computation for the KL constraint) and took much longer than the rest.

4.2.2 Robustness of Reinforcement Learning Controllers

The robustness of the controllers found by RL in Section 4.2.1 remains unclear and serves as a further criterion to choose a suitable RL algorithm. We conduct a qualitative Monte Carlo robustness analysis (MCRA) using variable Hamiltonian perturbation noise σ_{noise} of the 50 controllers computed for each chain length for all three algorithms. For each controller $\mathbf{s}_k$ found, we perturb the Hamiltonian $\mathcal{H}_{ss}$ using noise

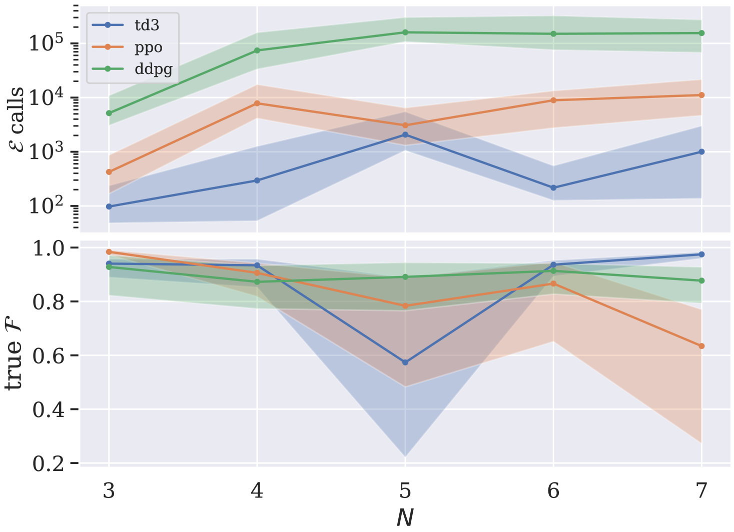


Figure 4.5: Top: Cost comparison between PPO, TD3 and DDPG for $|0\rangle$ to $|2\rangle$ for chains of length $N = 3, \ldots, 7$ with 100 single shot measurements and $\sigma_{\text{noise}} = 0.05$. The algorithms were run 50 times and the median $\mathcal{E}$ calls are plotted with the interquartile range. A perceived fidelity threshold of 0.99 was set as the termination criterion. Bottom plot shows true fidelities.

of the same triagonal form with mean 0 and the variance $\sigma^{(i)}_{\text{noise}} = 0.1k/9$, $k = 0, \ldots, 9$. We then evaluate the true fidelities $\mathcal{F}$ of the controller $\mathbf{s}_k$ for each level of perturbation without any additional noise. We repeat this ten times for all 50 controllers and combine the results into a single fidelity distribution. We then compare the distributions visually to ascertain the robustness of a respective control algorithm visually. This allows us to judge the expected fidelity of the controllers found by the algorithm.

The distributions are represented non-parametrically as 1D box-plots as shown in Fig. 4.6 for the spin transfer problem of $|0\rangle$ to $|2\rangle$ for chains of length 4 and 5. This is a representatitve example with the other cases being similar. This figure highlights that some fidelity distributions are heavy tailed with many outliers, meaning there is significant variation of fidelity between some controllers under perturbation. DDPG controllers, despite making more function calls, were the least robust when it came to preserving the interquartile width of the performance distribution. For PPO vs. TD3, there are cases where TD3 is better than PPO's and vice versa. However, PPO's performance was more consistent compared with TD3's. TD3, similar to REINFORCE and TRPO, showed a high variation in successful termination, getting stuck indefinitely at local minima for some problems, and there were gaps in the collected statistics due to timeouts. So we were only able to collect statistics for

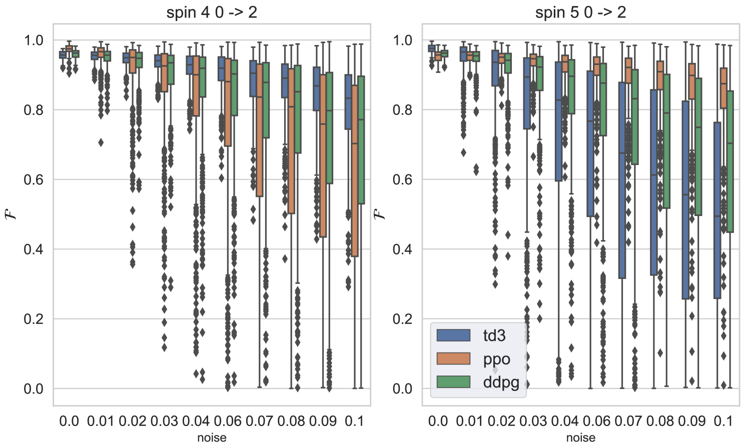


Figure 4.6: Robustness analysis for PPO, TD3 and DDPG for $|0\rangle$ to $|2\rangle$ for the 50 controllers found in Section 4.2.1 for chains of length $M = 4$ (left) and 5 (right). Ten levels of perturbation noise $\sigma_{\text{noise}} = 0, \ldots, 0.1$ are considered for each controller which is evaluated ten times to yield 500 points per box-plotted fidelity distribution.

some M for some of the cases in Section 4.2.1 without rerunning multiple times. On balance, we find that PPO performs most consistently compared to the other RL algorithms for multiple repetitions for different spin transitions. Its algorithmic performance therefore was more stable compared to the rest. In terms of the controller performances, we found no significant difference between the algorithms.

Even though PPO is not conclusively better from these results, we chose PPO as the single algorithm to represent the class of policy gradient RL algorithms for comparison with other types of control algorithms in Chapter 5 and for the comparison with gradient-based optimisation in the next section as we found: (1) it is faster for data collection to get enough statistics from multiple training runs, and (2) it is sufficient to empirically represent the class of policy gradient algorithms for our problem. TD3 and DDPG or any of the other RL algorithms might also be suitable with more tuning for the study but were not pursued chiefly due to time constraints and their comparative stability w.r.t. PPO was worse.

4.2.3 Cost of PPO vs. L-BFGS

A first step to compare PPO with gradient-based optimisation is to analyse the costs in terms of number of $\mathcal{E}$ calls (see Section 4.2.1) under the noiseless dynamics of the ideal model. For gradient-based optimisation, we use L-BFGS with restarts, which performed well on the studied control problem in earlier work [LSJ15b].

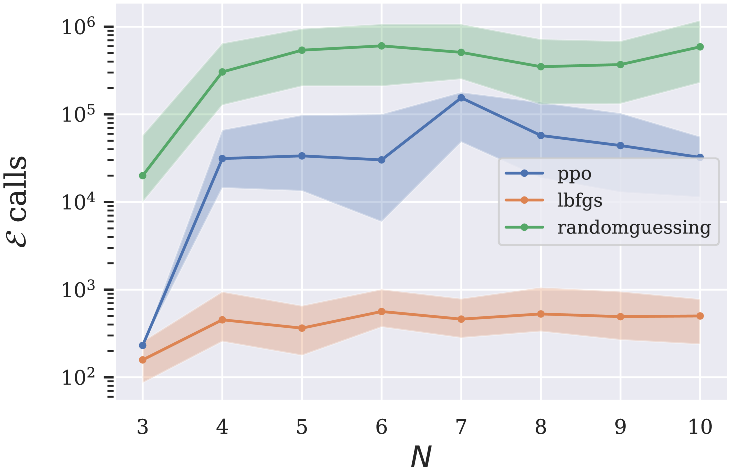


Figure 4.7: Comparision between L-BFGS, PPO and randomly guessing controllers for $|0\rangle$ to $|2\rangle$ for chains of length $N = 3$ to $N = 10$ without noise. The algorithms were run 50 times and the median $\mathcal{E}$ calls are plotted with the interquartile range. A threshold of $\mathcal{F} = 0.99$ is set for termination.

Fig. 4.7 shows how function calls scale with the length of the spin chain, $M = 3, \ldots, 10$, for a transition $|0\rangle$ to $|2\rangle$ for PPO, L-BFGS and randomly guessing controllers. The randomly guessed controllers are used to benchmark potential deviations in the computational difficulty of the problem. We stop once a fidelity threshold of 0.99 is crossed. The spin chain transition is computationally similar for all M as it depends largely on the relative distance between the spins, the control and time constraints, which are kept constant for all the problems we study. There is an initial jump from $M = 3$ after which all algorithms manifest a quite flat increase in the number of function calls as the length of the chain increases. This is likely because transitions in the three-chain are easier to achieve as simple Rabi oscillations which are generally trap free, and due to the existence of analytical solutions for this case which are absent for longer chains.

It is not surprising to observe that for an accurate model L-BFGS is mostly two orders of magnitude better than PPO. PPO has to consume most of the calls to build up an internal representation of the model before it can start optimizing. Adding small stochastic noise to the Hamiltonian should degrade the performance of L-BFGS considerably in terms of the number of function calls. To analyze this, we relax the termination constraint on the fidelity to 0.98 and consider only perturbations to H_{ss}, during optimisation, without single shot measurement noise. Single-shot measurement or perturbation noise renders L-BFGS incapable of estimating fidelities

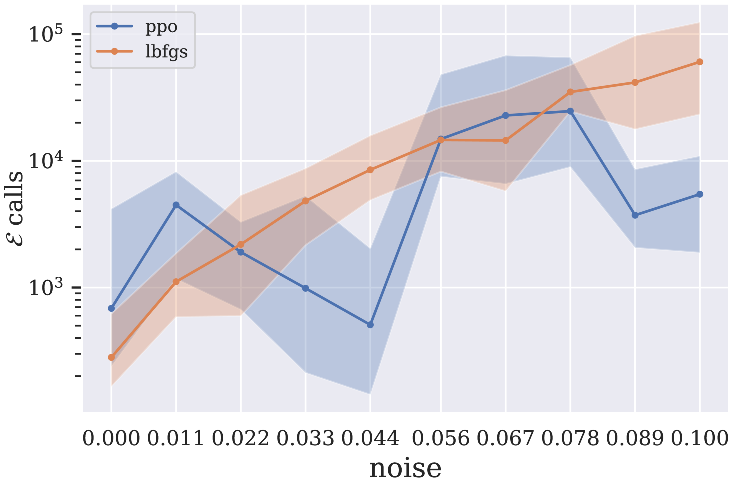


Figure 4.8: Number of $\mathcal{E}$ calls comparison between L-BFGS and PPO for $|0\rangle$ to $|2\rangle$ for a chain of length $M = 4$ as a function of Hamiltonian perturbation noise σ_{noise} with a termination fidelity threshold of 0.98. The algorithms were run 50 times and median $\mathcal{E}$ calls are plotted with interquartile range.

over 0.99 without making many millions of function calls (hence the reduction to 0.98). Fig. 4.8 demonstrates an approximately exponential rise in $\mathcal{E}$ calls for L-BFGS as the strength of the perturbation σ_{noise} is increased from 0 to 0.1. Clearly Hamiltonian perturbations deteriorate the performance of L-BFGS, while PPO keeps performing on a similar level than without noise. We conjecture that the noise helps PPO to explore the landscape more effectively and find optima as it only relies on an approximate gradient direction in contrast to L-BFGS that relies on the model-based gradient heaviliy in its protocol.

Large fluctuations for PPO at certain noise levels likely imply that it is unable to find robust solutions there. The fluctuations may be linked to the noise level and the existence of an optimal noise level at which highly robust solutions can be found. More work, however, is needed to test this idea which is pursued further in Chapter 5.

Single shot measurement noise considered in Section 4.2.1 has not been employed here, as this would have made the problem even harder for L-BFGS as it has not been designed for noisy optimisation. Overall these results are likely due to high sensitivity of the optimization descent step of L-BFGS to small perturbations in the low rank Hessian components. This causes the number of iterations to steeply increase. Note that $\mathcal{E}$ calls go down for PPO from around 10^5 to around 10^4 in Fig. 4.7, and we observe a similar effect in Fig. 4.5.

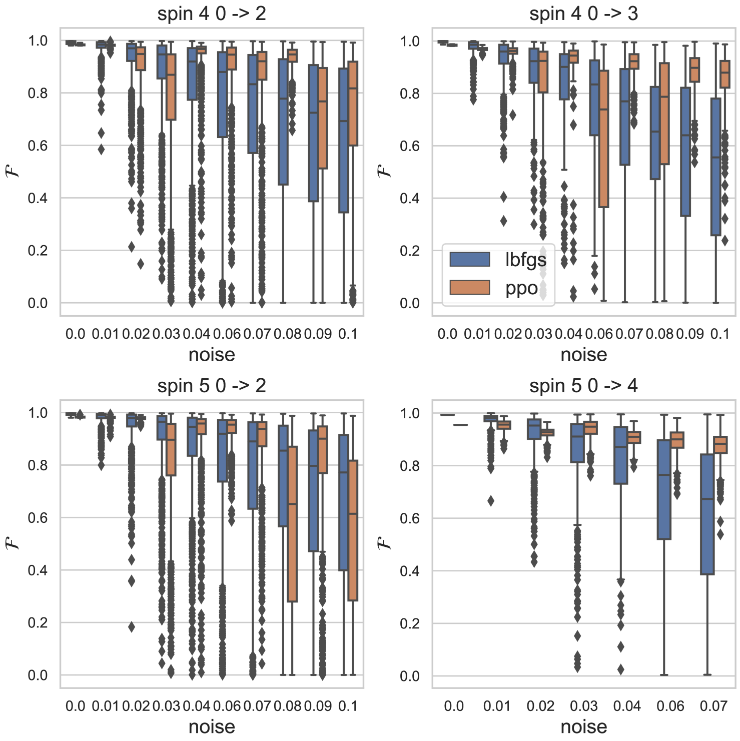


Figure 4.9: Comparison of 100 L-BFGS controllers computed without noise and 100 PPO controllers trained under Hamiltonian perturbation noise σ_{noise} for transitions to the middle and end of chains of length $M = 4, 5$.

4.2.4 Robustness of PPO and L-BFGS Controllers

We conduct an MCRA (see Section 4.2.2) to compare robustness of 100 controllers found by L-BFGS under ideal conditions and model-free PPO under low Hamiltonian perturbation noise. There are two cases worth considering: (1) the robustness of PPO controllers found at different levels of Hamiltonian perturbation; (2) the robustness of PPO controllers w.r.t. Hamiltonian perturbation found at a particular noise level. Both cases are compared to 100 L-BFGS controllers for each transition using the ideal model without noise. The termination condition, in all cases, is $\mathcal{F} \geqslant 0.99$.

For (1), we consider transitions to the middle and end for $M = 4, 5$, as shown in Fig. 4.9. We use PPO controllers trained with Hamiltonian perturbation noise σ_{noise} that corresponds to the noise level on the x axis from 0.01 to 0.1. We find, as expected, that the width of the fidelity distribution for L-BFGS controllers slowly increases as σ_{noise} is increased from 0 to 0.1. The expected fidelity is further dropping from being concentrated around $\mathcal{F} = 0.99$ to a very flat width and increasingly heavier tail, down to $\mathcal{F} = 0$. For PPO controllers, however, we observe that at certain noise levels, e.g., $\sigma_{\text{noise}} = 0.01, 0.04, 0.07$, the controllers found for all problems have narrow distributions compared with L-BFGS. At other noise levels, e.g., $\sigma_{\text{noise}} = 0.08, 0.1$ for $M = 5, |0\rangle$ to $|2\rangle$, they have wider distributions for some problems, but also narrow distributions for others, e.g., $\sigma_{\text{noise}} = 0.08, 0.1$ for $M = 4, |0\rangle$ to $|2\rangle$. We conjecture that added perturbations may have a smoothing effect on the optimization landscape which would result in either filtration or creation of "barriers" near optima in some cases.

For (2), we consider in addition to the cases of (1), also transitions to the middle for $M = 6, 7$. Here the PPO controllers have been computed for low Hamiltonian perturbation noise $\sigma_{\text{noise}} = 0.01$. Both, the L-BFGS controllers and the PPO controllers, become worse with increasing noise levels. However, the PPO controllers drop off slower, except in the case of $M = 6$, $|0\rangle$ to $|3\rangle$. This suggests that overall PPO is more likely to find robust controllers.

To investigate this further, the performance of a well-performing PPO and L-BFGS controller for the $M = 5$, $|0\rangle$ to $|4\rangle$ transition is compared. For each algorithm, we select the controller with the the highest median fidelity across the ten noise levels to account for the heavy-tail nature of the performance distribution. The Hamiltonian is perturbed as $H_{ss} + \delta P$ where P is the perturbation direction and δ its strength. P is sampled uniformly on a nine-dimensional Euclidean sphere, created by the five perturbation for Δ_n and a further four for the coupling strengths. The fidelity was computed along these directions for δ from -0.1 to 0.1. The density of the curves is estimated at specific perturbation strengths and plotted (see Fig. 4.11). The PPO controller is clearly not at a fidelity maximum, so some perturbations have a chance to improve the fidelity. The L-BFGS controller is at a fidelity maximum, which means that most perturbation directions, including those on the couplings which are not control parameters, reduce the fidelity. Similar behaviour has been observed for other controllers.

4.3 Conclusions

In this chapter, we showed how policy gradient RL algorithms can be used for non-parametric constructions of optimization models for quantum control even under highly noisy conditions as seen in Section 4.2.1 where pure model-based methods perform poorly as seen in Section 4.2.3. This was possible through a novel model-agnostic MDP formulation of the quantum control problem of state preparation. Importantly, our proposed formulation has the advantage of being scalable in that the parameters of the control algorithm need not scale exponentially with the size of the controllable system due to the fact that it does not need to simulate the system's dynamics using a model and furthermore does not make the unrealistic assumption of having access to the unitary that is not observable. Moreover, we also demonstrated the utility of this formulation in the setting where the dynamics are stochastic and the model might not be very useful. This technique can also be extended to gate optimization or higher level VQAs [Per+14] whose classical optimization sub-routine can be replaced by a policy gradient algorithm. The other useful feature of our technique is that it can be used in a real experimental setting. Here, the tomography costs, needed to measure the state for feedback control, are exponentially growing with system size and can be circumvented using our approach.

We also explored the control landscape via probing local regions around optimal controllers via Monte Carlo perturbations and showcased the variation in optimal controllers found for a specific control problem that is evident in their values and also their robustness w.r.t. perturbations. This idea of robustness is connected with narrowness/flatness of the fidelity peaks in the control landscape and is an important takeaway of this chapter.

Moreover, by quantifying the cost of operation in terms of the number of function or environment calls, we systematically benchmarked the performances of different policy gradient algorithms covered in Chapter 2. We motivate our choice of PPO for the rest of the comparisons and for Chapter 5 beyond the intra-RL benchmarking due to its consistency in performance, i.e., stability for variations of the energy landscape control problem. For these reasons, we only use PPO for control algorithm benchmarking on a much wider class of control algorithms in Chapter 5.

In the absence of noise, RL performance is lower bounded by model-based optimisation and upper bounded by pure random guessing. This implies that a nonparametric model is being constructed. The cost of model construction is relatively bounded by random guessing for RL under noisy conditions. However, the number of calls is still high. Model-based RL or Bayesian methods could be explored to reduce the reliance on information acquisition. Towards that end, we show in Chapter 6 how a model-based RL approach that incorporates partial knowledge of the controllable system can reduce the number of calls significantly.

In Section 4.2.2, a Monte Carlo robustness analysis is conducted for comparison of RL controller between each other and PPO controllers with L-BFGS controllers obtained with restarts to understand robustness of controllers found in various settings. This is a qualitative visual comparison that is further refined in Chapter 5 through the development of a novel Robustness Infidelity measure (RIM). We demonstrate that RL controllers found under low Hamiltonian perturbation noise levels are typically more robust compared with those found by L-BFGS, under no perturbation noise, but there is variation in the quality of their robustness that needs to be explored more as a function of their clustering and correlation of locations in the optimization landscape. It appears that in some cases RL finds controllers that may not be optimal for the ideal model, but perform robustly at high fidelity under noisy conditions. This suggests that Hamiltonian noise in particular can improve robustness of some controllers. RL is a promising avenue for feedback adaptive control with less overhead compared with variational methods and is arguably comparatively better with uncertainties. However, a careful construction of the control problem in an RL paradigm is needed before its application.

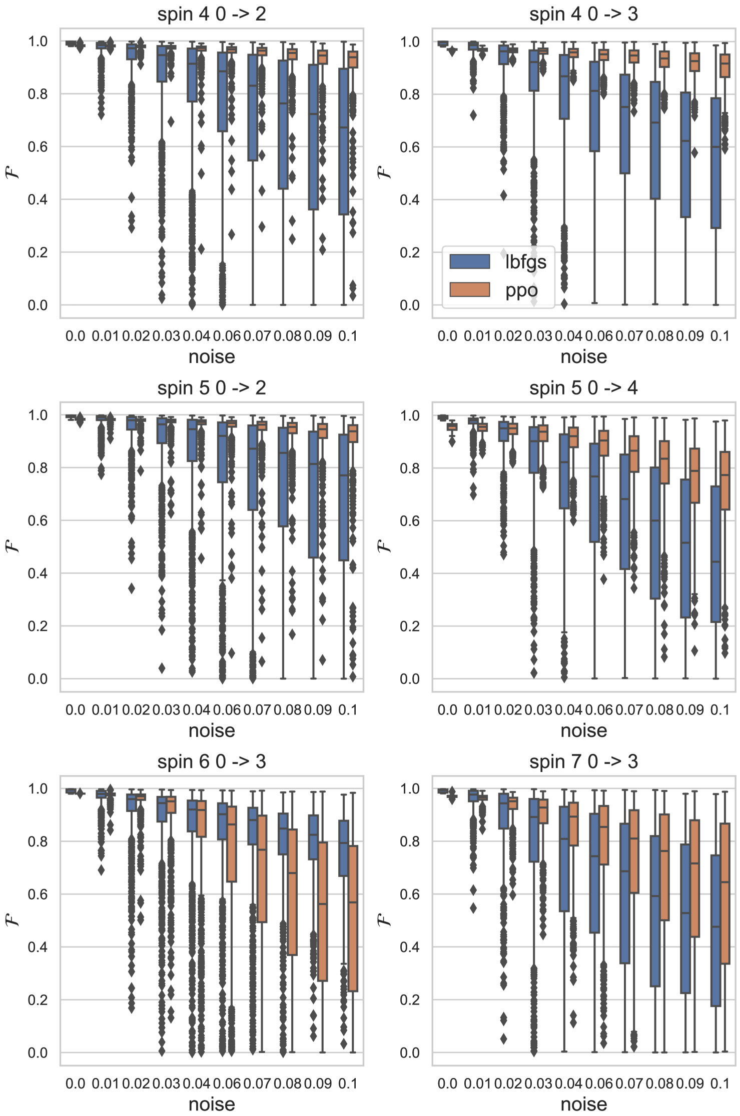


Figure 4.10: Comparison of controllers found by L-BFGS without noise and PPO trained under low Hamiltonian perturbation noise $\sigma_{\text{noise}} = 0.01$ and perfect measurements. We consider transitions to the middle and end of chains of length $M = 4, 5$ and to the middle for $M = 6, 7$.

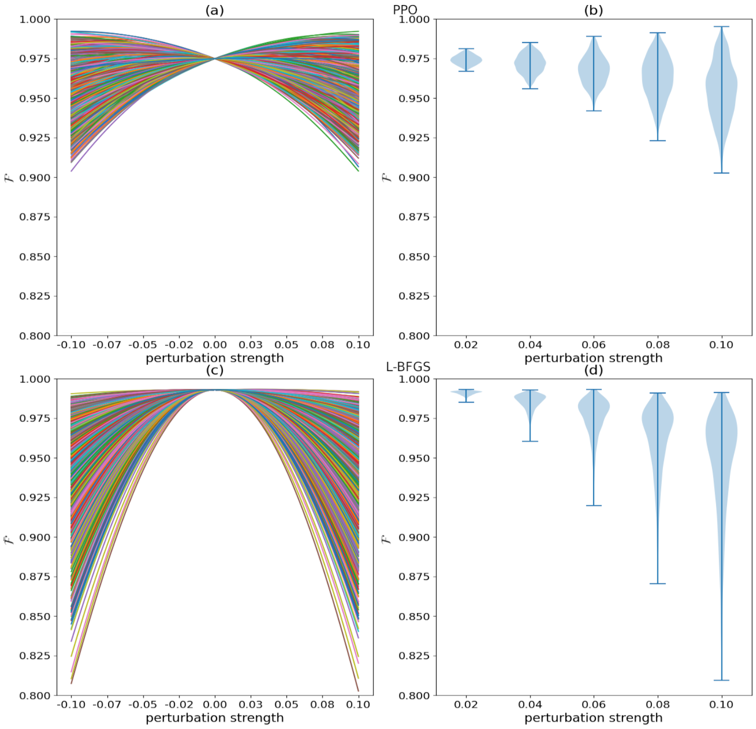


Figure 4.11: Robustness comparison of a well performing PPO (top) and L-BFGS (bottom) controller for $M = 5$, $|0\rangle$ to $|4\rangle$. (a) and (c) show $1,000$ fidelity curves, sampled along different Hamiltonian perturbation directions; (b) and (d) show density distributions of these curves at the perturbation strengths.

Chapter 5

Robustness certification of quantum controllers

Standard quantum optimal control (QOC) methods for steering quantum devices mostly focus on finding controls that have high fidelity using mathematical models [CX20; Sri+21; Blu+21a]. However, if the operation of quantum devices is subject to noise, high fidelity itself is insufficient to gauge performance of a control scheme, and extra effort is required to systematically search for solutions that are both, robust against noise and have high fidelity [AGS21; JSL18]. This requires a notion of robustness and ideally a single measure that can capture robustness and fidelity, enabling the identification and construction of more efficient methods to find controls that satisfy both properties.

In this Chapter, we introduce a general statistical diagnostic based on the Wasserstein distance of order p [Vil09] to evaluate the robustness and fidelity of quantum control solutions and the algorithms used to find them. This is applicable to any quantum control problem where the fidelity is a random variable with a probability distribution over $[0,1]$. The Wasserstein distance between probability distributions is a measure of the minimal costs of probability mass transport between two distributions. In Sec. 5.2, the p-th order *Robustness-Infidelity Measure* (RIM_p) is defined to quantify the robustness and fidelity of a quantum controller. The RIM_p p-th order Wasserstein distance between the probability distribution for the fidelity induced by noise and the ideal distribution for a perfectly robust controller, described by a Dirac delta function at unit fidelity. We show that the RIM_p is the p-th root of the p-th raw moment of the infidelity distribution – a non-parametric measure independent of any particular assumption for the distribution. This has the advantage of recovering the average infidelity as RIM_1 which is easy and practical to compute and intuitively makes sense as a RQOC optimization target. We also showcase the RIM to quantify and improve the qualitative robustness evaluation of individual controllers and control algorithms presented in Chapter 4.

5.1 Outline of the study

In Sec. 5.3 we illustrate the RIM and draw useful insights for robust quantum control by generating controllers for static energy landscape control of the XX Heisenberg model, introduced in Chapter 4.1.1 and exploiting the degree of freedom afforded by the existence of multiple optima in quantum control [BCR10]. We analyze their robustness properties and the performance of algorithms in finding effective controllers using four optimization algorithms representing different, commonly employed approaches: (1) L-BFGS: a second-order gradient-based optimization using an ordinary differential equation model of the quantum system to compute the fidelity under perfect conditions [Zhu+97]; (2) Proximal Policy Optimization (PPO): a model-free reinforcement learning algorithm, having no prior knowledge of the system [Sch+17]; (3) Nelder-Mead: a derivative-free simplex-based heuristic search method [NM65]; and (4) Stable Noisy Optimization by Branch and Fit (SNOBFit): another derivative-free method that performs model-free learning by using regression to estimate gradients via a branch and fit method [HN08].

For individual controller comparisons, (1) performs optimization over a noise-free fidelity objective functional under ideal conditions. To that end, we use standard L-BFGS with an ordinary differential equation model to compute the fidelity without perturbations during optimization. This serves as a baseline to understand the performance of optimizing noiseless objective functionals compared to the noisy optimization performed by all other selected algorithms and its impact on the robustness of the controllers found. We have explored stochastic gradient descent methods (e.g. ADAM [KB17]) and also tested a noisy version of L-BFGS that has been recently proposed that modifies the line search and lengthening procedure during the gradient update step [Shi+21] and found that our training noise scales were too large and washed away gradient information, rendering these algorithms unsuitable for our study. (2) represents a machine learning approach with minimal knowledge. (3) and (4) are derivative-free methods to handle stochastic objective functionals. We have covered all algorithms in Chapter 2. In particular, we recall that PPO uses a discounted reward signal (e.g. fidelity) accumulated over multiple interactions with the optimization landscape using non-parametric models: the policy and value functions. Both are estimated using neural networks in a control problem agnostic fashion. Doing this allows the incorporation of perturbations during training which specifically has advantages in finding robust controls for energy landscape problems [Kha+21]. In this chapter, we use a control problem formulation of PPO for Eq. (4.2) as described in Chapter 4.

The motivation behind the RIM is that it has practical utility in that it allows us to choose among similar, high-fidelity controllers, as a post-selector or filter for robust controllers. Moreover, it is agnostic of the algorithm used to find these controllers. This may be computationally more efficient than optimizing the RIM directly, as we see in Sec. 5.3.3. Moreover, it can also be adapted to compare the performance of control algorithms in finding not only high-fidelity but also robust controllers. To that end, we introduce an Algorithmic RIM (ARIM), averaging RIMs over multiple controllers, in Sec. 5.2.

The ARIM compares algorithm performance in finding robust controllers. For selecting the algorithms for our numerical study, we used the following motivating principles:

1. investigate and understand the performance of algorithms commonly used in quantum control;
2. consider algorithms that do and do not require gradient information; and
3. consider reinforcement learning, more recently also used in quantum control

These choices are not exhaustive but serve as a diverse set of algorithms to which we apply the RIM and ARIM, illustrating their utility and giving some indication of the performance of common control algorithms for the specific robust control problem.

Our experimental motivation is four-fold:

(A) By comparing the robustness of controllers without regard to the optimization algorithm, we wish to answer whether high fidelity implies high robustness using the RIM of the individual controllers (Sec. 5.3.1).

(B) By conducting a distributional comparison of controllers we wish to understand how likely it is that a given algorithm produces controllers in an ideal (no-noise) setting that are robust in noisy conditions (Sec. 5.3.1.2).

(C) To study the effect of training noise of the same nature as the robustness noise model applied *during* optimization on an algorithm's ability to find robust controllers using the ARIM (Sec. 5.3.2). For a fair comparison, we conduct (B) and (C) with a fixed number of objective function calls allotted to each algorithm.

(D) In Sec. 5.3.3, we try to understand an algorithm's asymptotic ability to find robust controllers using the ARIM through optimizing the RIM by allowing unlimited objective function calls. We consider two settings in this scenario: stochastic and non-stochastic fidelity optimization. In the latter case we optimize over a fixed set of Hamiltonians sampled once according to a noise model, while in the former case the Hamiltonians are stochastically chosen at each objective function evaluation using the same noise model.

5.2 Measuring robustness and fidelity of quantum controls

5.2.1 Robustness Infidelity Measure

Uncertain dynamics turn the fidelity $\mathcal{F}$ into a random variable with a probability distribution $\mathbf{P}(\mathcal{F})$. Intuitively, we call a controller robust if this distribution has a low spread. While a low spread alone may indicate robustness, low fidelity means the controller does not realize the target operation well. So we also expect a fidelity close to 1. That means the perfect distribution under any uncertainties is δ_1 – the Dirac delta distribution at maximum fidelity 1. In particular, we consider the delta function δ_x to be defined by an indicator cumulative distribution function (CDF),

$$C(a) = \begin{cases} 1 & \text{if } a \geqslant 0, \\ 0 & \text{if } a < 0. \end{cases} \tag{5.1}$$

This permits the familiar delta function property for integration w.r.t. a basic (rapidly diminishing) function,

$$\int_{-\infty}^{\infty} g(x)\delta_{x-a}\, dx = \int_{-\infty}^{\infty} g(x)\, dC(x-a) = g(a). \tag{5.2}$$

Our goal is to define a distance between probability distributions that measures closeness between the ideal and the achieved probability distribution in order to combine high fidelity and its robustness into a single measure.

For this we take the Wasserstein or Earth mover's distance $\mathcal{W}$ [Vil09; ACB17] due to the facts that: (1) it allows us to compare two probability distributions that do not share a common support, and, in particular, compare discrete and continuous distributions; (2) its easy geometric interpretation helps with its optimization; and (3) a simplification allows it to be calculated easily, as shown next.

The dual formulation of the p-th order Wasserstein distance [RTC17] between two distributions μ, ν is given by

$$\mathcal{W}_p(\mu, \nu) = \sup_{h,g} \left[\int h(x)\, d\mu(x) - \int g(y)\, d\nu(y) \right]^{\frac{1}{p}}, \tag{5.3}$$

where $h(x) - g(y) \leqslant \|x - y\|^p$. Even though this form seems abstract, for one-dimensional distributions, we can analytically compute the optimal maps h, g with

Theorem 5.1. *(Prop. 1 in [RTC17]) The p-th Wasserstein distance $\mathcal{W}_p(\mu, \nu)$ for one-dimensional probability distributions μ and ν with finite p-moments can be rewritten as*

$$\mathcal{W}_p(\mu, \nu) = \left(\int_0^1 |Q_\mu(z) - Q_\nu(z)|^p \, dz \right)^{\frac{1}{p}}$$

where $Q_\mu(z) = \inf\{x \in \mathbb{R} : C_\mu(x) \geqslant z\}$ denotes the quantile function and C_μ is the cumulative probability function of μ and likewise for Q_ν.

Remarkably, the optimal transport distance between one-dimensional distributions μ, ν over all possible transportation plans can be computed in terms of their quantile functions Q_μ, Q_ν. From here, following Thm. 5.1, it is straightforward to define the p-th *Robustness-Infidelity Measure*,

$$\mathrm{RIM}_p := \mathcal{W}_p(\mathbf{P}(\mathcal{F}), \delta_1) = \left(\int_0^1 |Q_{\mathbf{P}(\mathcal{F})}(z) - 1|^p \, dz \right)^{\frac{1}{p}}. \tag{5.4}$$

We now show that RIM_p can written in terms of the raw moments. That is, we prove the following

Proposition 5.2. *RIM_p is the p-th root of the p-th raw moment of the infidelity:*

$$RIM_p = \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}} \tag{5.5}$$

where f is a fidelity sample drawn from the distribution $\mathbf{P}(\mathcal{F})$ and $1-f$ is the corresponding infidelity sample. We use the expectation operator defined as $\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}[(\cdot)] := \int(\cdot)\mathbf{P}(\mathcal{F}=f)\,df$.

Proof. In the subsequent argument, recall that the quantile function is the inverse of the CDF function. Following Thm. 5.1, we can write the RIM_p as

$$\mathrm{RIM}_p = \left(\int_0^1 |Q_{\mathbf{P}(\mathcal{F})}(z) - Q_{\delta_1}(z)|^p\,dz\right)^{\frac{1}{p}}. \tag{5.6}$$

Note that both terms in the integrand are 0 at $z = 0$. Then, for $z \in (0,1]$, by definition,

$$\begin{aligned} Q_{\delta_1}(z) =& \inf\{x \in \mathbb{R} : C_{\delta_1}(x) \geqslant z > 0\} \\ =& \inf\{x \in \mathbb{R} : C(x=1) \geqslant z > 0\} \\ & \text{ using the CDF of } \delta_1 \text{ in Eq. (5.1)} \\ =& 1. \end{aligned} \tag{5.7}$$

Another way to see this is to use the inverse property $Q_{\delta_1}(\cdot) = C_{\delta_1}^{-1}(\cdot)$. The CDF is 0 in the interval $[-\infty, 1)$ and 1 in $[1, \infty]$. Next, we perform a change of variable $z = C_{\mathbf{P}(\mathcal{F})}(f)$. The differential is given by, $dz = \frac{dC_{\mathbf{P}(\mathcal{F})}(f)}{df}df = \mathbf{P}(\mathcal{F} = f)df$ as the derivative of the CDF w.r.t. the random variable is the probability distribution function. Substituting the terms, we get

$$\mathrm{RIM}_p = \left(\int_0^1 |Q_{\mathbf{P}(\mathcal{F})}\left(C_{\mathbf{P}(\mathcal{F})}(f)\right) - 1|^p \mathbf{P}(\mathcal{F}=f)\,df\right)^{\frac{1}{p}}. \tag{5.8}$$

Now we use the fact that $Q_{\mathbf{P}(\mathcal{F})}(C_{\mathbf{P}(\mathcal{F})}(f)) = f$ (inverse property) to obtain,

$$\mathrm{RIM}_p = \left(\int_0^1 \mathbf{P}(\mathcal{F}=f)|f-1|^p\,df\right)^{\frac{1}{p}}. \tag{5.9}$$

Since the domain of integration remains invariant, for fidelity measures with support in $[0, 1]$, it can be extended to $[-\infty, \infty]$. We obtain,

$$\begin{aligned} \mathrm{RIM}_p &= \left(\int_{-\infty}^{\infty} \mathbf{P}(\mathcal{F} = f)|f-1|^p \, df\right)^{\frac{1}{p}} \\ &= \left(\int_{-\infty}^{\infty} \mathbf{P}(\mathcal{F} = f)(1-f)^p \, df\right)^{\frac{1}{p}} \\ &\quad \text{as } f \leqslant 1 \text{, switch the order and drop } |\cdot| \\ &= \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]. \end{aligned} \tag{5.10}$$

We obtain the last line using the expectation operator defined in the propositon. □

For $p = 1$, using Eq. (5.10), we recover the average infidelity,

$$\mathrm{RIM}_1 = \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[1-f\right] = 1 - \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[f\right]. \tag{5.11}$$

Further expansions of the RIM_p in terms of the scaled moments are presented in Appendix A.1. Moreover, we note that one can also recover the probability distribution $\mathbf{P}(\mathcal{F})$ from RIM_p and we show its theoretical possibility in Appendix A.2.

To compute the RIM_p, we estimate $\mathbf{P}(\mathcal{F})$ using n fidelity samples $f_1, f_2, \dots, f_n$. Such samples may be obtained in practice via Monte Carlo simulation or physical experiments [FL11b]. Hence, barring the computational or experimental expense of obtaining these samples, the RIM_p is easy to compute. In case the dynamics of the system are certain, i.e., $\mathbf{P}(\mathcal{F}) = \delta_f$ for some constant fidelity value f, the RIM_1 is equal to the infidelity $1 - f$. Moreover, the RIM_1 is small if and only if the controller is robust (in the sense of the fidelity distribution having a low spread) and is also close to the maximum fidelity.

5.2.2 The Average Fidelity is Sufficient for Robustness Comparisons

We motivate why the RIM_1 is sufficient for comparing robustness and fidelity of controllers by making use of the fact that the RIMs of different orders computed on the estimated fidelity distribution are in agreement. We obtain bounds between the lower and higher order RIMs with

Proposition 5.3. *The following bounds hold:*

$$RIM_{p'} \leqslant n^{\left(\frac{1}{p}-\frac{1}{p'}\right)} RIM_p, \tag{5.12a}$$

$$RIM_p \leqslant RIM_{p'} \tag{5.12b}$$

for $p < p'$, where n is the number of samples used to estimate the RIM.

Proof. Using Lyapunov's inequality, stating that $\mathbb{E}[|X|^q]^{1/q} - \mathbb{E}[|X|^p]^{1/p} \geqslant 0$ for $q \geqslant p > 0$ for some $\mathbb{E}[|X|^t] < \infty$, we show that

$$\begin{aligned} \mathrm{RIM}_q - \mathrm{RIM}_p &= \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^q\right]^{\frac{1}{q}} - \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}} \\ &= \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[|(1-f)|^q\right]^{\frac{1}{q}} - \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[|(1-f)|^p\right]^{\frac{1}{p}} \geqslant 0. \end{aligned} \tag{5.13}$$

For any $q \geqslant p \geqslant s > 0$, it follows that $\mathrm{RIM}_q \geqslant \mathrm{RIM}_p \geqslant \mathrm{RIM}_s$. The converse is true without the p-th roots. The linearity of expectations implies that $\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p - (1-f)^q\right] \geqslant 0 \iff 0 < p \leqslant q$.

We can also derive a lower bound on RIM_p. For some $p' \geqslant p$, we have

$$\begin{aligned} \mathrm{RIM}_{p'} \leqslant \mathrm{RIM}_p^{\frac{p}{p'}} &= \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p'}} \\ &= \frac{\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}}}{\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}-\frac{1}{p'}}} \\ &\leqslant \frac{\mathrm{RIM}_p}{\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)\right]^{1-\frac{p}{p'}}} \\ &\leqslant \frac{\mathrm{RIM}_p}{\left(\min_f (1-f)\right)^{1-\frac{p}{p'}}} \end{aligned} \tag{5.14}$$

where the relation in the second last line is obtained by applying Jensen's inequality and the final line is obtained from the observation that $\min_f (1-f) < \mathbb{E}[1-f]\ \forall f$. Note that this result still depends on the data. Higher orders p and p' of the RIM are related to each other in a concave sense and when $p, p' \to \infty$, the RIMs become more equivalent. Conversely, near perfect fidelity, all the RIMs are converging to 0, but the presence of an outlier fidelity sample strongly governs how much discrepancy there still is between a higher-order RIM and a lower order RIM. This discrepancy is still concavely dependent on p and p'.

We arrive at the proposed relations for RIMs of different order by noting that

$$\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right] \geqslant m \sup_f (1-f) = m$$

for the smallest positive finite measure $m > 0$ on the domain set on which we define the probability distribution $\mathbf{P}(f)$. This follows from the continuity of f and the continuity of $\mathbf{P}(f)$. If f already has an ideal distribution, then this is trivially true. Assume there exists some subspace $S \in F = [0,1] \in \mathbb{R}$. Now assume that there exists some ϵ such that $1 - f + \epsilon > \sup_f(1-f)$. S is the subspace where this is true. So,

$$\int_{F \text{ without } S} \mathbf{P}(f) df (1-f)^p + \int_S \mathbf{P}(f)\, df (1-f+\epsilon)^p \geqslant \int_S \mathbf{P}(f)\, df (1-f+\epsilon)^p \tag{5.15}$$

$$> \int_S \mathbf{P}(f)\, df \sup_f (1-f) \tag{5.16}$$

$$\geqslant m \sup_f (1-f). \tag{5.17}$$

Let $\epsilon \to 0$ and we have $\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right] \geqslant m$. Eq. (5.14) yields

$$\mathrm{RIM}_{p'} \leqslant m^{\left(\frac{1}{p'} - \frac{1}{p}\right)} \mathrm{RIM}_p. \tag{5.18}$$

In practical settings, e.g. when using the ECDF, $m \geqslant \frac{1}{n}$. Intuitively, this follows from the observation that for any $\mathbb{E}_{\mathbf{P}(X)}[X^p] = \int P(X) X^p \, dX \approx \frac{1}{n}\sum_{i=1}^n X_i^p$ using samples $X_1, \ldots, X_n$. For the estimated[1] $\widehat{\mathrm{RIM}}_p$,

$$\widehat{\mathrm{RIM}}_{p'} \leqslant n^{\left(\frac{1}{p} - \frac{1}{p'}\right)} \widehat{\mathrm{RIM}}_p. \tag{5.19}$$

This implies that RIMs of different orders are similar (in the convergence sense) when the bound holds. □

Note that Eq. (5.12b) is stronger and states that RIM_p is less sensitive to outliers than $\mathrm{RIM}_{p'}$ while Eq. (5.12a) states that for fixed n and p, $\mathrm{RIM}_{p'}$ growth is sublinear ($\propto \exp(-1/p')$). This can be made tighter by adding additional assumptions on the nature of $\mathbf{P}(\mathcal{F})$, but these depend on the specific control problem. The upper bound becomes loose with increasing n, but highlights the constraining nature of deviation of higher-order RIMs from RIM_1.

This means that the higher-order RIMs do not capture more useful robustness information for comparisons, with the base case in Eq. (5.12b) being decided by the RIM_1. RIM_1 has low sensitivity to outliers, which makes it easier to estimate than

[1] hats are (generally) used to denote estimates

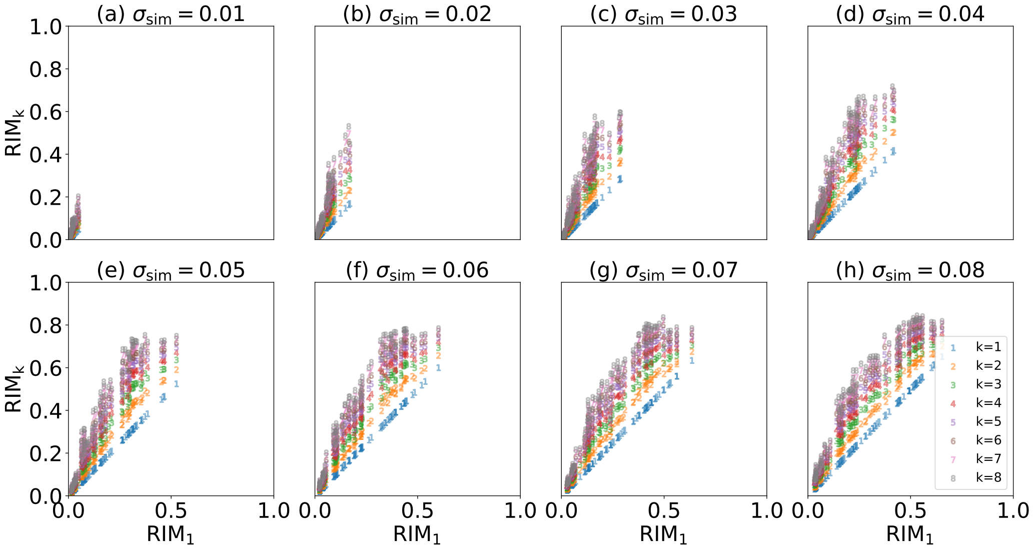


Figure 5.1: RIM_k scaling as a function of RIM_1 for 100 controllers for $M = 5$ and the transition from $|1\rangle$ to $|3\rangle$ and $N = 100$ samples per controller for RIM evaluation. Each subplot (a)-(h) corresponds to a noise level σ_{sim} indexing the fidelity probability distribution $\mathbf{P}_{\sigma_{\text{sim}}}(\mathcal{F})$. There is convergence and thus more agreement in RIM_k for small values.

higher-order RIMs, which, like the worst-case fidelity, are harder to accurately practically obtain (as more samples are required, which also explains the presence of n in the inequality).

The bound in Eq. (5.12b) gets tighter for large n but also for decreasing infidelities, so in this regime, the RIMs are in agreement. Another way to see this is to note that the Wasserstein distance provides a structure-preserving geodesic between any fidelity distribution to the ideal δ_1: the distributions converge together with their RIMs of any order. In other words, the convergence in the Wasserstein distance, for fixed n, RIM_1 can effectively constrain any RIM_p with $p > 1$, since growth in p is sublinear. This implies that the RIMs converge when they tend to 0 as seen in Fig. 5.1. We also note that the higher order RIMs increase the measure's sensitivity to outliers greatly, even though growth in the RIM is sublinear in p. So especially when approaching the ideal distribution δ_1, i.e., in case RIM_1 is small for high-fidelity, robust controllers, there is strong agreement between RIMs of all orders. For example, the variance of distributions decreases as $\sim (1 - \min \mathcal{F})^2$ as $\min \mathcal{F} \to 1$ in $[0, 1]$.

However, the fact that outliers are more influential for higher RIM orders proves useful for optimization [GW21] where such behavior is sought after, while our goal here is robustness/fidelity comparison. For this goal, in general, outliers are obstructive as they would hide the general distributional trend. From now on we refer to the RIM_1 without the subscript.

5.2.3 Connecting perturbations with the fidelity random variable

Next, we define the noise in the system as perturbations (defined in Chapter 4.1.1) of its uncertain dynamics that give rise to $\mathbf{P}(\mathcal{F})$. Recall that a perturbation to the full Hamiltonian in Eq. (2.1) can be expressed as $\tilde{H}(t, \mathbf{u}) = H(t, \mathbf{u}) + \gamma S \in \mathbb{C}^{n \times n}$ where $\gamma \in \mathbb{R}$ describes the strength of a perturbation and $S \in \mathbb{C}^{n \times n}$ its structure, usually normalized using some matrix norm. To induce an uncertainty into the dynamics we treat γ and S as random variables drawn from some probability distributions. This give us a general way to represent any physically relevant uncertainties in Hamiltonian parameters.

The structure S may be fixed, e.g., describing the uncertainty in some coupling parameter for the Hamiltonian, while γ is drawn from a normal distribution. This would be consistent with a (linear) *structured perturbation* in classical robust control theory [JSL17]. Instead, S may also be drawn from a probability distribution, describing uncertainties across multiple Hamiltonian parameters. While this generalizes structured perturbations, note that they do remain linear w.r.t. the strength. If S is sampled uniformly on the unit-sphere, according to its normalization, we have an *unstructured perturbation*, with (uncertain) strength determined by γ. Conceptually, if γ is drawn from a normal distribution with zero mean and standard deviation σ, γS describes a "fuzzy" ball $\mathcal{B}_\sigma$ around $H(t, \mathbf{u})$. In this chapter, we consider unstructured perturbations that are less idealized, in some sense, than the structured perturbations (usually considered in classical control [Doy82]), allowing the robustness results to be interpreted generically without the need to consider specific sources of uncertainties arising from specific quantum device designs. For simplicity, we write $\mathbf{P}_\sigma(\mathcal{F})$ for a fidelity distribution obtained by unstructured perturbations drawn from $\mathcal{B}_\sigma$.

Our quantification of robustness is dependent on the choice of γS and the uncertainties in these quantities. Note that neither the choice of the noise model nor the magnitude of the noise level is restricted, as our approach is not perturbative around the optimum $(t_{\mathrm{opt}}, \mathbf{u}_{\mathrm{opt}})$, which is how noise is usually modelled in the literature [Hou+12; Hoc+14; Kab+14; SHR06; BWS15; DR98]. This approach becomes relevant when confidence in an analytical physical model is low or there are missing

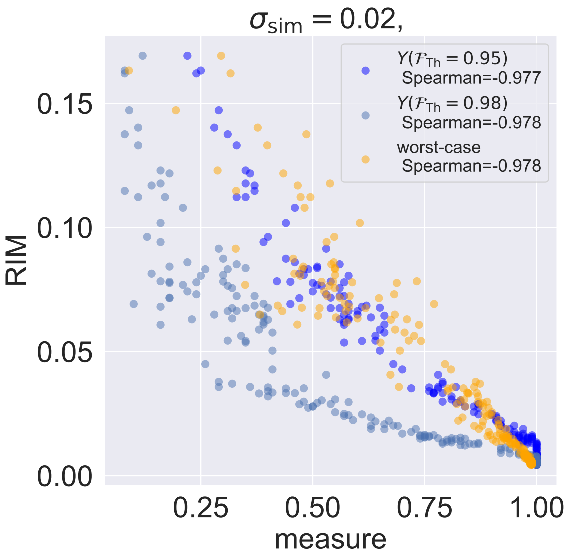


Figure 5.2: RIM values generated from $\mathbf{P}_{0.02}(\mathcal{F})$ with $N = 100$ samples for 200 controllers are plotted against the yield $Y(\mathcal{F}_{\rm Th})$ at fidelity thresholds $\mathcal{F}_{\rm Th} = 0.95, 0.98$ and the worst case fidelity. Both measures are correlated, as encapsulated by the high negative Spearman correlation coefficients [Spe04] and p-values $< 10^{-4}$.

terms that cannot be analytically or perturbatively accounted for, e.g. complicated noise sources. This is also in accordance with modern robustness theory and the μ function in classical [ZD98] and quantum [Sch+22a; JSL17; Sch+21b] settings.

To further motivate the RIM, we study how it compares with other statistical measures of robustness. The RIM generally correlates with worst-case or minimum sample fidelity, variance or higher moments and the yield function $Y(\mathcal{F}_{\rm Th})$, which is the fraction of fidelities greater than a threshold fidelity $\mathcal{F}_{\rm Th}$. Fig. 5.2 shows a scatter plot of RIM values versus $Y(0.95)$, $Y(0.98)$ and the worst-case fidelity for an example problem using Eq. (4.2), discussed in Chapter 4.1.1. The RIM has an advantage over Y in that it does not depend on an arbitrary choice of $\mathcal{F}_{\rm Th}$.

5.2.4 Connection with classical robustness measures

We can relate the RIM with the differential sensitivity that is essentially the scaled log sensitivity [ONe+23] (the dimensionless measure of classical robustness introduced in Chapter 3.2)

$$\zeta(S,T) = \left.\frac{\partial e(T; S_\mu, \sigma)}{\partial \sigma}\right|_{\sigma=0} \tag{5.20}$$

where recall that $e = 1 - \mathcal{F}$ is the infidelity. We shall obtain the connection by considering the expected differential sensitivity $\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}[\zeta]$. The dependence of the function $\mathbf{P}_\sigma(\mathcal{F} = f)$ and ζ on σ, the noise strength requires careful attention and the technical description that follows suit relies on removing the dependence of the probability distribution function on σ.

The main idea is to shift the randomization from the fidelity $\mathcal{F}$ to perturbation $\mathbf{S}$ and fix the noise scale σ as part of the fidelity functional that is now simply a function $\mathcal{F}(\ldots, \sigma)$ with an additional parameter. This is described in more detail below.

For a simple noise model, where the perturbation is simply added to a base Hamiltonian, such as the Gaussian perturbations from Chapter 4 that is used throughout this thesis, this is straightforward and is similar to the reparametrization trick introduced in for example Ref. [KW19]. Since each entry in the structured perturbation is a Gaussian sample drawn from $\mathcal{N}(0, \sigma^2)$, we can factorize the noise strength and just draw a scaled sample from $\sigma\mathcal{N}(0, 1)$ where the scaling is done after a random sample is drawn. This way, the dependence on the noise term is isolated deterministically in the fidelity function which becomes independent of the randomization which is averaged over in the expectation. We encode that randomness in the perturbation operators, represented by the random variable $\mathbf{S}$ and a sample $S \sim \mathbf{P}(\mathbf{S})$ drawn from its distribution has its non-zero entries as standard Gaussian samples.

This reparametrization is not always possible, since not all perturbations are linear. Nevertheless, under the linearity of perturbations assumption, we are able to write an equivalent expectation operator that isolates the dependence of $\mathbf{P}_\sigma(\mathcal{F} = f)$ on σ to just the fidelity/error sample with a new probability distribution function independent of σ. Using the definition of the differential sensitivity in Eq. (5.20), we derive the following result:

Theorem 5.4. *Under the linear noise modelling assumption, the expected differential sensitivity is the differential sensitivity of the* RIM*, i.e.,* $\mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}[\zeta(S_\mu, T)] = \left.\frac{\partial \mathrm{RIM}(\sigma)}{\partial \sigma}\right|_{\sigma=0}$.

Proof: We first unpack the differential sensitivity using the definition of the derivative,

$$\begin{aligned}\left.\frac{\partial e(T; S, \sigma)}{\partial \sigma}\right|_{\sigma=0} &= \lim_{\epsilon \to 0^+} \left.\frac{e(T; S, \sigma + \epsilon) - e(T; S, \sigma)}{\epsilon}\right|_{\sigma=0} \\ &= \lim_{\epsilon \to 0^+} \frac{e(T; S, \epsilon) - e(T; S, 0)}{\epsilon}. \end{aligned} \tag{5.21}$$

We apply the expectation operator $\mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}\left[\cdot\right]$ on Eq. (5.21) and simplify using the reparametrization trick: $\mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}[\cdot] \leftrightarrow \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[\cdot\right]$:

$$\begin{aligned}
\mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}[\zeta(S,T)] &= \mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}\left[\lim_{\epsilon\to 0^+}\frac{e(T;S,\epsilon)-e(T;S,0)}{\epsilon}\right] \\
&= \lim_{\epsilon\to 0^+}\frac{\mathbb{E}_{\mathbf{P}(\mathbf{S}=S)}\left[e(T;S,\epsilon)-e(T;S,0)\right]}{\epsilon} \\
&= \lim_{\epsilon\to 0^+}\frac{\mathbb{E}_{\mathbf{P}_\epsilon}\left[e(T;S,\epsilon)-e(T;S,0)\right]}{\epsilon} \\
&= \lim_{\epsilon\to 0^+}\frac{\mathrm{RIM}(\epsilon)-\mathrm{RIM}(0)}{\epsilon} \\
&= \left.\frac{\partial \mathrm{RIM}(\sigma)}{\partial\sigma}\right|_{\sigma=0}.
\end{aligned}$$

Swapping the limit and the expectation in the second line is justified as long as the limit in the mean of the sequence $\left\{\frac{e(T;S,\epsilon)-e(T;S,0)}{\epsilon}\right\}_{\epsilon>0}$ exists. □

5.2.5 Measuring the Performance of Control Algorithms

We can also apply the previous arguments to derive a measure to compare the ability of control algorithms to find high-fidelity, robust controllers. Let $\mathbf{P}(\mathrm{RIM})$ be a distribution of RIM values of controllers obtained by a particular algorithm and a particular control problem with specific uncertainties. This can be estimated by sampling L controllers produced by the algorithm. The ideal of this distribution is δ_0, so that we can define the *Algorithmic Robustness Infidelity Measure*,

$$\mathrm{ARIM} := \mathcal{W}_1(\mathbf{P}(\mathrm{RIM}),\delta_0) = \mathbb{E}_{r\sim\mathbf{P}(\mathrm{RIM})}\left[r\right], \tag{5.22}$$

following the same argument as before. The ARIM is small if and only if the underlying RIM distribution $\mathbf{P}(\mathrm{RIM})$ has higher density at or near RIM $= 0$, i.e., is close to the ideal δ_0.

Note that δ_0 ideal might not be the best realistically achievable RIM in all settings and can be tailored with more knowledge/data of the specific system. However, since we are going to be comparing ARIMs across different control algorithms, its choice is immaterial if the best achievable RIM for all algorithms is the same. Another approach could be to use a value that is different for each algorithm (i.e. the achievable ideal robustness values differ for each algorithm). This would make the ARIM a relative algorithmic robustnss measure which could be a useful metric of consistency or reliability of a control algorithm in its ability to produce robust controllers. However, this would also make absolute robustness comparisons across the control algorithms difficult which is indeed our goal in the proceeding sections.

5.3 Numerical experiments

We study the robustness of static control problems, where the controls are time independent, instead of the usual time dependent controls. Previous work has shown that particularly robust controls can be found for these systems [SJL18; AGS21; JSL18; JSL17]. While these systems are often not fully controllable, solutions for specific operations can be found via optimization [LSJ15a]. We use the same setup as the XX Heisenberg Hamiltonian with unstructured perturbations defined in Chapter 4.

Recall that the the static approach is simpler in the sense of having fewer control parameters to optimize over, which reduces computational and experimental complexity. This makes the problem suitable to demonstrate the practical usage of the RIM and improve the qualitative robustness analysis conducted in Chapter 4. It also provides a concrete example to explore the robustness properties of the control algorithms as well as the controllers they find. A numerical example illustrating the RIM via the empirical CDF (ECDF) for two controllers for the information transfer control problem is shown in Fig. 5.3.

To explore the robustness of controllers and corresponding control algorithms whose experimental motivation is presented in Sec. 5.1, we perform a Monte Carlo robustness analysis using the RIM on numerical solutions to the same spin chain information transfer problem covered in Chapter 4 for chains of length $M = 4, 5, 6, 7, 8, 9$ with $J = 1$.

We look at transitions from the start of the chain $|1\rangle$ to the end $|M\rangle$ and from $|1\rangle$ to the middle $|\lceil \frac{M}{2} \rceil\rangle$. The former transition is physically easy to control while the latter is more challenging [LSJ15a] as transitions to the middle exhibit anti-core behavior – where the central spin state is the hardest to excite [JLS14].

We collect the best 100 solutions, ranked by their fidelity, obtained by all the control algorithms. Each algorithm has a budget of 10^6 fidelity function evaluations. The budget correlates with the run time for each algorithm. It is imposed to allow for a fair comparison of the algorithm robustness performance under similar resources, while being agnostic to specific implementations and speed differences.

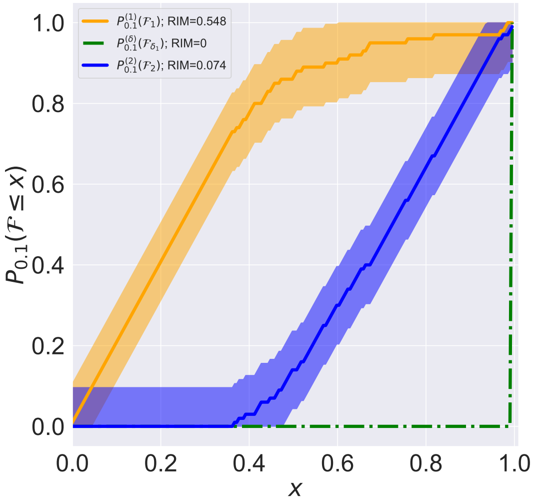


Figure 5.3: To illustrate the RIM robustness measure, two static controllers for an XX spin chain of length five for transferring an excitation from spin $|1\rangle$ to $|3\rangle$ are compared. The empirical approximations to the CDFs for the two controllers, $l = 1, 2$, were simulated using 100 bootstrapped perturbations with $\sigma = 0.1$, giving fidelity distributions $\mathbf{P}_{0.1}(\mathcal{F}_l)$ for the fidelity random variables $\mathcal{F}_l$. The fidelity distribution $\mathbf{P}_{0.1}(\mathcal{F}_{\delta_1})$ for a perfectly robust controller with $\mathcal{F}_{\delta_1}$ is also shown. The ECDFs are estimated using 500 bootstrap repetitions. The 0.95 confidence bounds on their error are obtained using the Dvoretsky-Kiefer-Wolfowitz inequality [DKW56]. Closeness to the perfectly robust controller can be interpreted as having a smaller area under the curve and is indicated by the RIM values.

We initialize $\mathbf{\Delta}, t$ with quasi Monte Carlo samples from the Latin Hypercube [Loh96; Ste87; Owe19] to increase convergence rate and decrease clustering of controllers. This permits coverage of the parameter domain with $O(1/\sqrt{N})$ samples as opposed to $O(1/N)$ for random sampling, where N is the number of initial values. Our constraints are $0 \leq t_f \leq 70$ and $-10 \leq \Delta \leq 10$. We use 100 bootstrap samples to estimate fidelity distributions throughout. The perturbation strengths γ^J_j and γ^C_c are scaled by J and Δ respectively as per Eq. (4.3). Note, for $\sigma = 0$, $\mathbf{P}_0(\mathcal{F}) = \delta_{\mathcal{F}}$ is deterministic.

The perturbation strengths are drawn from a normal distribution with standard deviation σ_{train} determining the strength of the noise added for the optimization. σ_{sim} is the noise level used in the simulations to assess the robustness of the controllers found. Implementation details are in Appendix A.4. The optimization objective is noiseless $\mathcal{F}$ for Sec. 5.3.1, stochastic $\mathcal{F}$ with unstructured perturbation S_σ for Sec. 5.3.2, and the RIM for the non-stochastic problem and a stochastic $\mathcal{F}$ with S_σ in Sec. 5.3.3.

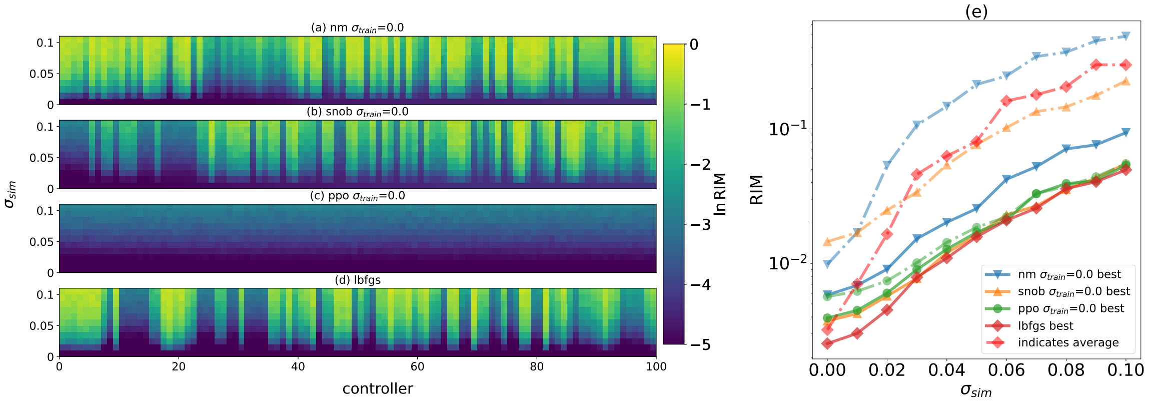


Figure 5.4: (a)-(d) 100 controllers found for the XX spin chain model, Eq. (4.2), using Nelder-Mead, SNOBFit, PPO, and L-BFGS for $M = 5$ and the spin transition from $|1\rangle$ to $|3\rangle$ with $\sigma_{\text{train}} = 0$. The controllers are ranked in increasing order of infidelity at $\sigma_{\text{sim}} = 0$ from left to right. Each column represents a single controller's RIM at $\sigma_{\text{sim}} = 0, 0.01, 0.02, \ldots, 0.1$ from the bottom to the top on a log scale. Even if the infidelity or RIM at $\sigma_{\text{sim}} = 0$ is close to 0, some controllers' RIM values degrade faster than others and are hence less robust despite starting at very low infidelities. (e) RIM as a function of σ_{sim} for the average and best controller (i.e., most dark over all σ_{sim} levels) out of the 100 shown in (a)-(d) in terms of how much they preserve their corresponding RIM rank average across all σ_{sim}. Each algorithm is indicated by a marker shape, and the solid and dash-dotted lines denote the best and average controller lines respectively. All the best controllers have very high initial fidelities and are very similar across the different control algorithms, with Nelder-Mead being only moderately worse.

5.3.1 Characterization of All Controllers Found with Constrained Resources

5.3.1.1 Ranking Individual Controllers

In this section, we address our motivating question (A) in Sec. 5.1, whether high fidelity implies high robustness for an individual controller. We also numerically demonstrate the non-linear and non-uniform deterioration of robustness with increasing noise which implies a trade-off between higher fidelity at no noise and robustness at higher noise levels.

To this end, we employ control algorithms to optimize an objective functional without noise, i.e., setting $\sigma_{\text{train}} = 0$ (see Sec. 5.2.3), under the general optimization conditions outlined at the start of Sec. 5.3. We rank these controllers by their infidelity values and then compute the RIM values for various levels of simulation noise,

$\sigma_{\text{sim}} = 0.01, 0.02, \ldots, 0.1$.

For example, Figs. 5.4(a)-(d) show a pseudo-color plot of the RIM values for 100 controllers found for the chosen test control problem (chain of length $M = 5$, target spin transfer $|1\rangle$ to $|3\rangle$). The lowest infidelity controllers start from the left and are indexed by columns 1 to 100 indicating their respective ranks according to their RIM at $\sigma_{\text{sim}} = 0$. The RIM values, as a function of σ_{sim}, for individual controllers grow at different rates despite starting at quite similar small values for all algorithms. The main result, that applies also to all transitions (not explicitly shown here), is that the high fidelity controllers do not, in general, preserve their ranks as σ_{sim} increases. E.g., for SNOBFit (see Fig. 5.4(b)), the RIM for controllers 6, 8, 9, 11–13 grows much more rapidly than for controllers 24–33, indicated by rapid color changes from dark (low RIM) to light (high RIM) in the vertical direction. Interestingly, almost all controllers found by PPO have very low RIM across σ_{sim} values compared to the other control algorithms (color remains dark for longer). This is, however, not reflective of PPO's general behavior on the extended sample of problems we examined (see Appendix A.6). It could be limited fundamentally by the existence of robust controllers and/or the resource budget for a particular problem (see Fig. A.2 in Apendix A.5 showing results for other transitions).

We further evaluate the best performing individual controller. To this end, we seek the controller that preserves its overall RIM rank average the most across the noise levels. It is computed using the reshuffled RIM ranks of each controller for all values of σ_{sim}. Likewise, we locate the controller that has the median RIM rank average across the noise levels as the averagely performing controller. Most of the RIM rank sum distributions studied were symmetric, and their median was close to their average value. So we can try to understand average controller RIM rank order consistency in terms of how the median controller performs. We compare the RIM values of the median with the best controller in Fig. 5.4(e) for all algorithms, showing the RIM values for the best and median controller as a function of σ_{sim}.

For all algorithms, the best and the average controllers have similar infidelities (initial RIM value) in Fig. 5.4(e). Their behavior as a function of σ_{sim} is different and is generally non-linear. Thus, the best controllers, despite being distinguishable from the others at $\sigma_{\text{sim}} = 0$, become indistinguishable for higher σ_{sim} and point at a trade-off between infidelity (at no noise) and robustness that could be leveraged when selecting a controller to be deployed for a noisy system. Moreover, the RIM curve of the best controller among all algorithms (here L-BFGS) suggests a fundamental limitation on RIM for this problem. It is likely not possible to obtain curves that are lower, but this remains theoretically unresolved.

5.3.1.2 Ordinal Kendall Tau for $\mathrm{RIM}_{\sigma_{\mathrm{sim}}}$-Rank Consistency

To address the motivating question (B) in Sec. 5.1, how likely a given algorithm is to produce robust controllers that were obtained in an ideal (no-noise) setting, we are interested in how consistently a controller acquisition strategy produces controllers with low RIM.

To that end, we reduce the RIM rank consistency property of the top-k controllers across two perturbation strengths $\sigma_{\mathrm{sim}}^{(i)}$ and $\sigma_{\mathrm{sim}}^{(j)}$ to a prediction problem by asking the following: **(Q)** *How well does the RIM rank of a controller, when ordered at strength $\sigma_{sim}^{(i)}$, predict the RIM rank of the controller at strength $\sigma_{sim}^{(j)}$?*

To answer this question, let us denote the controller RIM $\sigma_{\mathrm{sim}}^{(i)}$-rank order by the vector $\mathbf{r}^{\sigma_{\mathrm{sim}}^{(i)}}$, and compute an ordinal (binned/categorical) version of the Kendall-tau-B statistic $\tilde{\tau}$ [Ken62; Agr10], a measure of statistical dependence between $\mathbf{r}^{\sigma_{\mathrm{sim}}^{(i)}}$ and $\mathbf{r}^{\sigma_{\mathrm{sim}}^{(j)}}$. The ordinals are constructed only for $\mathbf{r}^{\sigma_{\mathrm{sim}}^{(i)}}$ by binning using a discrepancy parameter α that indicates the fraction of the maximum RIM value difference within a single bin. The binned rank order $\tilde{\mathbf{r}}^{\sigma_{\mathrm{sim}}^{(i)}}(\alpha)$ minimizes the effect of small movement in either rank due to noise. Then $\tilde{\tau}$ is computed by

$$\tilde{\tau}(\sigma_{\mathrm{sim}}^{(i)}, \sigma_{\mathrm{sim}}^{(j)}) = \tilde{\tau}_{i,j} = \frac{\sum_{l<m} \mathbb{I}_{l,m}^{+} + \mathbb{I}_{l,m}^{-}}{\sqrt{\left(K - t_{\mathrm{total}}^{(i)}\right)\left(K - t_{\mathrm{total}}^{(j)}\right)}} \tag{5.23}$$

where

$$\mathbb{I}_{l,m} = sgn\left(\tilde{\mathbf{r}}_l^{\sigma_{\mathrm{sim}}^{(i)}} - \tilde{\mathbf{r}}_m^{\sigma_{\mathrm{sim}}^{(i)}}\right) sgn\left(\mathbf{r}_l^{\sigma_{\mathrm{sim}}^{(j)}} - \mathbf{r}_m^{\sigma_{\mathrm{sim}}^{(j)}}\right) \tag{5.24}$$

are the l, m-th sign products of the rank order differences at $\sigma_{\mathrm{sim}}^{(i)}, \sigma_{\mathrm{sim}}^{(j)}$ with $+/-$ denoting the positive/negative pair contributions; $K = k(k-1)/2$ is the number of total pairs being compared; $t_{\mathrm{total}}^{(i)} = \sum_l t_l^{\sigma_{\mathrm{sim}}^{(i)}}(t_l^{\sigma_{\mathrm{sim}}^{(i)}} - 1)/2$ are the total numbers of ties where $\mathbb{I}_{l,m} = 0$ for $\sigma_{\mathrm{sim}}^{(i)}$ and likewise for $t_{\mathrm{total}}^{(j)}$. For complete positive/negative rank order correlation $\tilde{\tau} = \pm 1$ and $\tilde{\tau} = 0$ for zero rank order correlation. For our hypothesis test, we assumed a worst case p-value of 10^{-4} as an acceptance criterion on the numerical results that follow and also that the controllers generating these rank orders are independent of each other. In this case, this constraint is satisfied by the i.i.d. noise model for a given set of unique controllers corresponding to different points

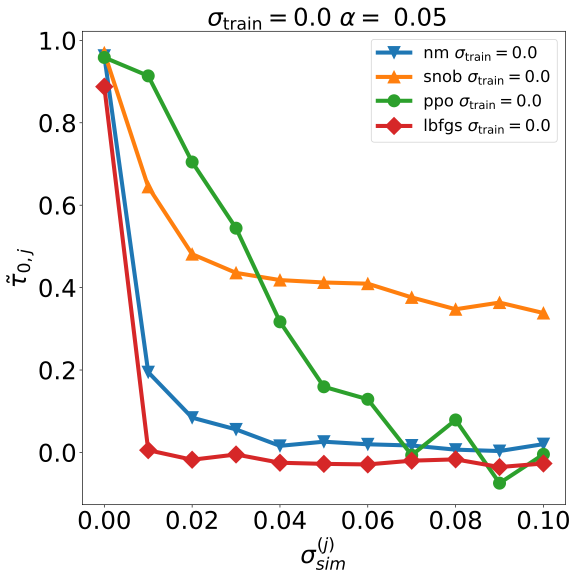


Figure 5.5: RIM rank order consistency statistic $\tilde{\tau}$ for the 100 controllers found for the problem $M = 5, |1\rangle$ to $|3\rangle$ between the two levels: no simulation noise, $\sigma_{\text{sim}}^{(i)} = 0$ and $\sigma_{\text{sim}}^{(j)}$ from $\{0.0, 0.01, \ldots, 0.1\}$ for (a) Nelder-Mead, (b) SNOBFit, (c) PPO, and (d) L-BFGS without training noise. In other words, this is the correlation of infidelity rank order with the general RIM ranks. The $\tilde{\tau}_{0,j}$ values decline the slowest for PPO until $\sigma_{\text{sim}}^{(j)} = 0.04$ and then SNOBFit takes over compared to the rest. This shows, for this case, that the PPO infidelity rank order correlates the most with RIM rank order for $\sigma_{\text{sim}} \leqslant 0.03$.

in a static optimization landscape. The independence over the choice of controllers is not necessary as all the consistency comparisons are for this fixed choice of controllers.

For our earlier spin chain example ($M = 5$ spins, transfer from $|1\rangle$ to $|3\rangle$), we focus on $\tilde{\tau}$ for $\sigma_{\text{sim}}^{(i)} = 0, \sigma_{\text{sim}}^{(j)}$ pairs that is sufficient to answer **(Q)**. More specifically, we aim to understand how well the no-noise RIM (i.e., the average infidelity) ranks correlate with the general RIM ranks. This is shown in Fig. 5.5 for each optimization algorithm for $\alpha = 0.05$. For the $\sigma_{\text{sim}}^{(i)} \geqslant 0.03$, the RIM rank order is the most consistent with $\tilde{\tau} \gtrsim 0.6$ for PPO excluding other algorithms. But there is larger shuffling of the ranks of PPO controllers as σ_{sim} increases with deteriorating $\tilde{\tau}$ and SNOBFit takes over. This may be due to small numerical differences in RIM (see Fig. 5.4(c)) observed, and thus a stronger consistency for $\sigma_{\text{sim}} \leqslant 0.03$ is captured.

We highlight next that the reason why PPO infidelities correlate more with RIM values at higher σ_{sim} is because it optimizes a discounted RIM ($\sum_i \gamma^i \text{RIM}^{(i)}$ for $0 \leqslant \gamma \leqslant 1$) as its reward function.

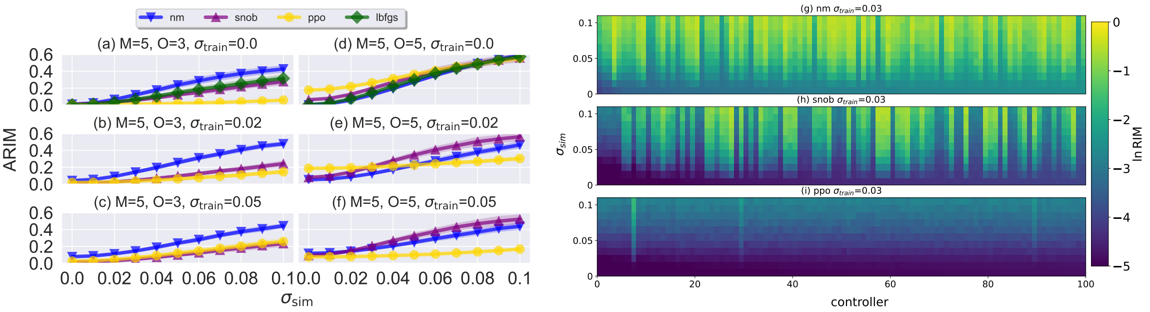


Figure 5.6: ARIM as a function of σ_{sim} for $M = 5$ where (a)-(c) are the end-to-middle ($|1\rangle$ to $|3\rangle$) and (d)-(f) are the end-to-end ($|1\rangle$ to $|5\rangle$) transitions (end denoted by O). The ARIM is computed from a distribution of RIM values for 100 controllers for each σ_{sim} for SNOBFit, Nelder-Mead, PPO and L-BFGS. We identify each algorithm with a unique marker and/or color. Both, PPO and SNOBFit, are run multiple times at $\sigma_{\text{train}} = 0, 0.02, 0.05$. PPO has higher variance w.r.t. σ_{train} than SNOBFit and Nelder-Mead, whose performance curves are more in line with the L-BFGS curve for $\sigma_{\text{sim}} \geqslant 0.05$ and mostly worse for $\sigma_{\text{sim}} \leqslant 0.05$. 95% confidence intervals (shading) are computed using non-parametric bootstrap resampling [Efr87] with 100 resamples or alternatively using PAC bounds derived in A.3. (g)-(i) show individual-controller comparisons for $\sigma_{\text{train}} = 0.03$ ranked by fidelity (leftmost is highest).

We follow the standard finite-horizon MDP formulation for the RL setting (covered in Chapter 4.1.3) for states, actions and one-step state transition rewards (s_t, a_t, r_t) that are sampled in trajectories $\tau = \{(s_t, a_t, r_t) : t = 1, \ldots, T\}$ stored in the buffer $\mathbf{D}$. Recall that the proximal policy optimization (PPO) algorithm uses a clip objective to update the policy π_θ parameters θ with first-order constraints that minimize policy distributional divergence. The policy objective is

$$\theta_{k+1} \propto \arg\max_{\theta} \sum_{\tau \in \mathbf{D}} \sum_{a_t, s_t, r_t \in \tau}^{T} A_{\pi_{\theta_k}}(s_t, a_t) \min\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \text{clip}\left(1 \pm \epsilon, \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}\right)\right], \tag{5.25}$$

where $\pi(\cdot)$ is the policy probability distribution. The advantage estimates are

$$A_{\pi_{\theta_k}}(s_t, a_t) = \sum_{i=t}^{T-1} (\gamma\lambda)^{i-t}(r_t + \gamma V_{\phi_k}(s_{t+1}) - V_{\phi_k}(s_t)) \tag{5.26}$$

with value function $V_\phi(s_t) = \mathbb{E}_\pi\left[\sum_{i=0}^{T-1} \gamma^i r_{t+i+1} | s = s_t\right]$ where ϕ are the value function parameters. The value function is regressed onto discounted rewards sampled according to $\pi(\cdot)$. The value function's optimization objective is

$$\phi_{k+1} \propto \arg\max_{\phi} \sum_{\tau \in \mathbf{D}} \sum_{t=0}^{T} \left(V_\phi(s_t) - \sum_{i=t}^{T} \gamma^i r_i(s_t^\tau)\right)^2. \tag{5.27}$$

The algorithm tries to maximize this expression. In the case of flat rewards and advantages $\lambda = \gamma = 1$, the advantage estimates are

$$A_{\pi_{\theta_k}}(s_t, a_t) = V_{\phi_k}(s_t) - \left(V_{\phi_k}(s_T) + \sum_{i=t}^{T-1} r_t \right) = V_{\phi_k}(s_t) - \widehat{V_{\phi_k}}(s_t). \tag{5.28}$$

The value function can be written in terms of an expectation under the policy, as an average reward: $V_\phi(s_t) = T\mathbb{E}_\pi \left[\frac{1}{T} \sum_{i=t}^{T} r_i | s = s_t \right]$. The optimal value function is defined by $V_*(s_t) = \max_\pi V_\phi(s_t)$, which is maximized if the policy is optimal, i.e., $\pi_\theta = \pi_{\theta^*}$ at $\theta = \theta^*$. Near optimality, the advantages are approximately 0 as there should be no advantages conferred to the optimal policy π_{θ^*} which also has an optimal value function. Thus, $\widehat{V_{\phi^*}}(s_t) \to V_{\phi^*}(s_t)$ as $A_{\pi_{\theta^*}} \to 0$. The sample rewards minus the predicted rewards by the value function go to 0 in Eq. (5.25). The same argument applies with discounts $\gamma, \lambda < 1$ and, hence, it can be shown that the algorithm optimizes a discounted RIM$_1$ estimator as its value function. Most reinforcement learning algorithms effectively optimize the average or cumulative reward $\hat{J} \propto \sum_i r_i$ due to the one-step heuristic application of the Bellman principle of optimality [Tho14].

We can extend this analysis for $\sigma_{\text{train}} > 0$ to further corroborate that the infidelity rank order for PPO correlates most with higher order RIMs. We plot the consistency statistic $\tilde{\tau}_{0,j}$ for all algorithms for $\alpha = 0.05$ for the case $M = 5$ and the transition $|1\rangle$ to $|3\rangle$ in Fig. 5.7(a)-(f) ((a) is Fig. 5.5) and $|1\rangle$ to $|4\rangle$ in Fig. A.1(a)-(f) for multiple training noise levels. Note that for each subplot the L-BFGS curve is always the same at $\sigma_{\text{train}} = 0$. The controllers found by PPO at $\sigma_{\text{train}} = 0.05$ are less consistent for some noise levels than others, e.g., $\sigma_{\text{sim}} \geqslant 0.04$ compared with the controllers found at $\sigma_{\text{train}} = 0.04$. This is also true for SNOBFit and Nelder-Mead. Moreover, the decline in the correlation values is smoothest for PPO compared to the rest for nearly all twelve instances shown in both figures. With more training noise, Nelder-Mead is sometimes closer in consistency to the controllers found to L-BFGS, e.g., Fig. 5.7(a,b). But it produces more consistent controllers with increasing training noise likely due to diminishing returns of the gradient direction, which makes its behavior more like SNOBFit and PPO.

For most PPO runs, the consistency statistic is highest for $\sigma_{\text{sim}} \leqslant 0.04$ and thus the infidelity rank order is a good predictor of RIM rank order for higher σ_{sim}, which was not observed for any of the other algorithms. Also note that this analysis does not reveal anything about how high the RIM values are for the controllers (a drawback of the non-parametric test) and should be processed as companion plots to the figures where these explicit values are shown.

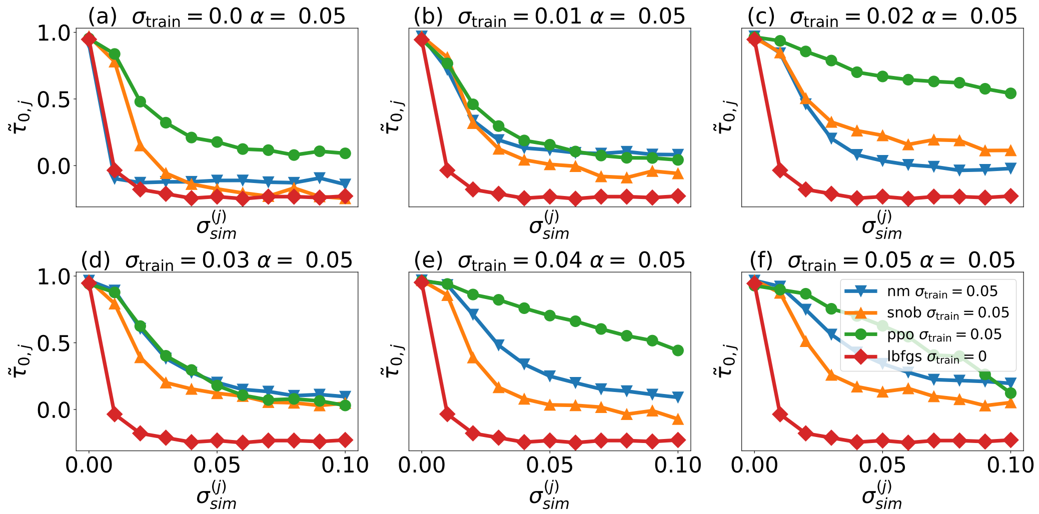


Figure 5.7: Consistency statistic $\tilde{\tau}_{0,j}$ for all algorithms at $\sigma_{\text{train}} = 0.0, \ldots, 0.05$ for discrepancy parameter $\alpha = 0.05$ for $M = 5$ and the transition from $|1\rangle$ to $|3\rangle$. Case (a) was presented earlier in Fig. 5.5. For (f), $\sigma_{\text{train}} = 0.04$, PPO is actually more robust in terms of ARIM growth compared with (e) as seen from their positions in Fig. 5.6(b). Characteristically, most low ARIMPPO controllers show high rank consistency in the region $0 \leqslant \sigma_{\text{sim}} \leqslant 0.04$. Nelder-Mead is similar to L-BFGS in all plots except (e) and (f), where it shows slightly more consistency than PPO and SNOBFit controllers.

The other algorithms typically have a sharper drop at $\sigma_{\text{sim}}^{(j)} = 0, 0.01$ step where the infidelity rank order for L-BFGS and, to a lesser extent, Nelder-Mead is completely non-informative (due to very high fidelity values without noise) and is not consistent with the orders at larger σ_{sim}. This is most likely because the controllers found are the result of second order, gradient-based or similarly successful search methods for finding optima precisely. Since PPO and SNOBFit are gradient-free, for $\sigma_{\text{sim}} \geqslant 0.03$, their controllers are more consistent in comparison. In this case, the infidelity rank order is more informative of the RIM rank order than, e.g., L-BFGS, as fidelities are not being fully maximized due to the absence of a strong gradient direction. Note that a viable link between the consistency statistic and a generic gradient-based algorithm is hard to establish, so this does not preclude the existence of algorithms that are $\tilde{\tau}$-wise better.

Finally, note that $\tilde{\tau}$ should be thought of as a proxy of reliability of an algorithm's capability to generate numerical control solutions whose infidelity values are more consistent and predictive of their RIM values at higher σ_{sim}. If strong correlation is obtained, this circumvents (or at least increases confidence for circumventing the latter's) computation.

However, high RIM rank order consistency does not imply that the RIM values remain low at higher noise. Rather, it indicates how much the RIM of a controller is predictive of the controller's relative robustness performance at a higher noise level. The non-parametric nature of $\tilde{\tau}$ removes information about the fidelity value range and should be viewed in conjunction with Fig. 5.4$(a)-(e)$. If the correlation signal is strong, it could be used to sidestep the evaluation of the RIM at non-zero noise in favor of using the infidelity instead, eliminating the need for expensive sampling.

5.3.2 Comparison of Control Algorithms with Constrained Resources

We address our motivating question (C) in Sec. 5.1: what is the effect of training noise on a control algorithm's ability to find robust controllers? The overall picture is complex in terms of algorithm rankings. We numerically confirm that there is a problem-dependent optimal noise level that best smooths the optimization landscape for algorithms to more consistently find robust controllers.

We collect 100 controllers at training perturbations $S_{\sigma_{\text{train}}}$ with training noise level $\sigma_{\text{train}} \in \{0, 0.01, \ldots, 0.05\}$ for PPO, SNOBFit and Nelder-Mead. We do not consider any training noise for L-BFGS, since only the former algorithms are designed to perform optimization with noisy perturbations. This involves using a stochastic fidelity (objective) function call evaluated under the single structured perturbation $S_{\sigma_{\text{train}}}$ (exactly analogous to $S_{\sigma_{\text{sim}}}$).

We select $\sigma_{\text{sim}} \in \{0, 0.01, \ldots, 0.1\}$ to evaluate the RIM of the controllers found at different noise levels with a budget of 10^6 objective function calls per run. Each run corresponds to 100 controllers found under this budget constraint. The ARIM is then used to quantify an algorithm's performance w.r.t. robustness and fidelity, based on the 100 controllers that it found during the run.

We only show the representative end-to-middle and end-to-end transition for the state-preparation problem for $M = 5$ at $\sigma_{\text{train}} = 0, 0.02, 0.05$ in Fig. 5.6. Results for other spin-transitions and training noises are presented in Appendix A.6.

Recall from Eq. (5.22) that the ARIM is a measure of how far the distribution $\mathbf{P}(\text{RIM})$ is from its ideal δ_0. The ARIM curves at different training noises in Fig. 5.6(a-f) increase at different rates σ_{sim}, starting from similar base ARIM values at $\sigma_{\text{sim}} = 0$ for each algorithm. Note that the base ARIM value coincides with the average infidelity over controllers, in the absence of training noise.

A spread in ARIM curves indicates that the probabilistic distance of RIM values w.r.t. the ideal for all controllers increases at different rates. So, the algorithm represented by the slowest growing curve is the best to find robust controllers.

Overall, SNOBFit's and Nelder-Mead's ARIM curves at various training noises perform similarly to L-BFGS across all problems. However, there are distinctions in the region of $\sigma_{\mathrm{sim}} \leqslant 0.05$ where L-BFGS curves start at lower ARIM values and grow more quickly compared to SNOBFit curves at various noise levels. In the region of $\sigma_{\mathrm{sim}} \geqslant 0.05$ the SNOBFit curves comparatively grow more slowly, possibly because the fidelity has degraded so much that further deterioration is less likely across all 100 controllers. The Nelder-Mead curves exhibit similar behavior to the SNOBFit curves in that there is less variance w.r.t. the σ_{train} levels, both, when overall performance is good and when it is poor.

Compared to other algorithms, there is more variance in the PPO ARIM curves across training noises for a particular spin transfer problem, with some curves overlapping each other. The best performing ARIM curve is PPO at $\sigma_{\mathrm{train}} = 0.05$ for the end-to-end transition shown in Fig. 5.6(f) (and for 6 of 8 cases in A.6). This indicates that PPO is often capable of finding robust solutions, but the optimal value of training noise varies across the transition problems.

We also present an extended RIM analysis (like in Sec. 5.3.1) for the controllers found for the same transition problem at training noises for the derivative-free approaches. The RIMs at $\sigma_{\mathrm{sim}} \in 0, \ldots, 0.1$ are plotted in Fig. 5.6(g)-(i) for PPO, SNOBFit and Nelder-Mead at $\sigma_{\mathrm{train}} = 0.03$. On an individual level, SNOBFit and Nelder-Mead controllers share more algorithmic robustness and fidelity characteristics with each other across σ_{train} than with PPO controllers, i.e., they have high RIM variance within distribution per σ_{train}. This performance is also comparable to the L-BFGS controllers shown in Fig. 5.4(d). However, individually, the controllers found by PPO differ significantly across σ_{train} where notably the RIM and ARIM values stay uniformly very low for the case $\sigma_{\mathrm{train}} = 0, 0.03$ and the controllers are generally distinctly robust compared to SNOBFit and Nelder-Mead controllers.

Finally, we suggest possible explanations for these differences in behavior between algorithms. Since SNOBFit constructs local quadratic models to estimate gradients, it effectively filters out the perturbations $S_{\sigma_{\mathrm{train}}}$. The manifestation of this effect is that the controllers at one training noise react similarly w.r.t. the RIM, compared to controllers at other training noises (including the case of no training noise) as well as controllers found by L-BFGS. For Nelder-Mead, there are fewer noise-adaptation mechanisms compared to PPO and SNOBFit for large noise perturbations that might affect the quality of the estimated gradient direction and hence the rate of growth of the ARIM w.r.t. simulation noise at higher training noise levels is unavoidable.

In contrast, PPO does not filter out the perturbations under $S_{\sigma_{\text{train}}}$ and forms its policy gradient estimates from stochastic fidelity function evaluations, which likely differentiates it from SNOBFit. PPO also effectively estimates the fidelity landscape non-linearly using a fixed two-layer linear (100×100 dimensional) neural-network, which may lead to generally better ARIMperformance.

5.3.3 Comparison of Control Algorithms with Unconstrained Resources

We consider the behavior of the aforementioned control algorithms with an unconstrained number of objective function calls to address our motivating question (D) in Sec. 5.1, that is, we seek to understand an algorithm's ability to find robust controllers via the ARIM – without the function call constraint. Furthermore, we wish to ascertain what the effect of the training noise level σ_{train} is on ARIM optimization.

We consider two objective function settings: (i) *stochastic objective*: for each evaluation, a new Hamiltonian is drawn according to the noise model, which corresponds to one $S_{\sigma_{\text{train}}}$ perturbation in $\mathcal{F}$ during a single evaluation; (ii) *non-stochastic objective*: where the evaluation is over k perturbed, but fixed, Hamiltonians, pre-drawn from the noise model such that optimization objective is a deterministic RIM computed from k fixed training perturbations $\{S^{(i)}_{\sigma_{\text{train}}}\}_1^k$. In this case, the function calls are counted as k as they amount to k different fidelity function evaluations per single optimization objective call. Furthermore, since we cannot compute the analytical gradient of both objective functions, in order to use L-BFGS [Zhu+97] in both settings, we use a version of L-BFGS that approximates the Hessian using forward differences.

For motivation, we can intuitively relate (i) to producing a number of different quantum devices corresponding to different Hamiltonians and choosing one randomly each time we measure the fidelity of a controller under optimization, while (ii) optimizes one quantum device with an uncertain Hamiltonian. Scenario (ii) is the more realistic one in the current quantum device landscape, but the stochastic setting will become more relevant as quantum devices are mass-produced.

We fix the control problem to be the end-to-middle $M = 5$ transition. We consider the change in average ARIM over $\sigma_{\text{sim}} \in \{0, 0.01, \dots, 0.1\}$ for the top 100 controllers w.r.t.

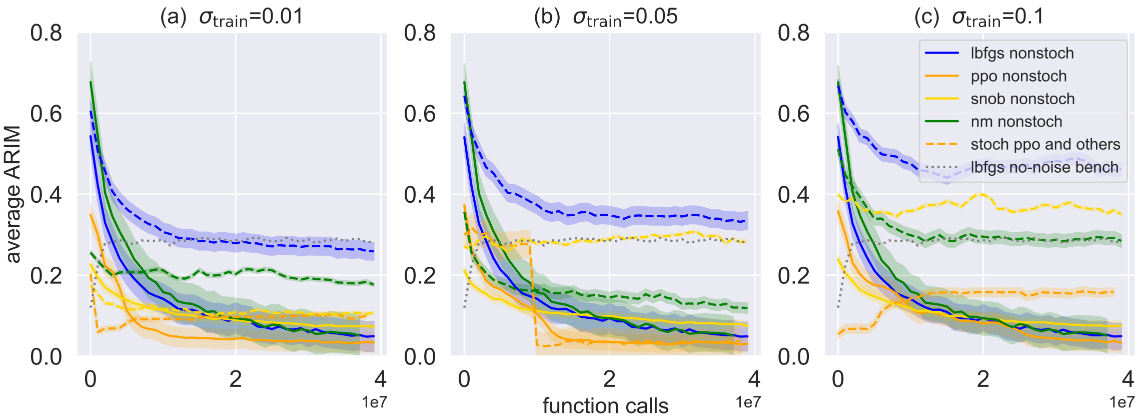


Figure 5.8: Asymptotic ARIM performance when the number of objective function calls is unconstrained for the $M = 5$ end-to-middle spin transfer problem. The ARIMs are averaged over the σ_{sim} set $\{0, 0.01, \ldots, 0.1\}$. The stochastic objective setting (i) is shown with dashed lines and deterministic RIM objective setting, (ii) with solid lines; different algorithms correspond to different colors. An L-BFGS no-noise benchmark is shown with a dotted line. The target RIM is computed using 100 non-stochastic fidelity evaluations. The ARIM is computed and averaged over a $\sigma_{\text{sim}} = 0.0, 0.01, \ldots, 0.1$ set. Plots (a)-(c) correspond to training noise $\sigma_{\text{train}} = 0, 0.05, 0.1$, where the curves are for 100 controllers ranked by the corresponding objective function evaluation (i/ii) and are updated every 10^6 function calls. For setting (ii), all control algorithms asymptotically reduce the average ARIM, but this is not cost competitive with the stochastic setting (i) where PPO performance reaches the local minimum for all noise levels with fewer function calls. We see that the training noise level can help the landscape exploration process; this positively affects PPO in (a), (c) and Nelder-Mead in (b). For setting (i), L-BFGS, then SNOBFit, then Nelder-Mead, then PPO is the most prone to performance deterioration w.r.t. σ_{train} due to the differences in their reliance on (estimated) gradient information. In these plots, the shading indicates 95% confidence intervals, determined by using bootstrap resampling.

function calls. Three training noises $\sigma_{\text{train}} = 0, 0.05, 0.1$ are considered. The controller rankings are maintained w.r.t. the objective function and are updated in steps of 10^6 function calls up to 4×10^7. For the stochastic setting (i), we maintain the controller ranking via the stochastic fidelity function evaluation. For the non-stochastic setting (ii) we maintain the ranking through the deterministic RIM obtained using $k = 100$ pre-drawn training perturbations $\{S^{(i)}_{\sigma_{\text{train}}}\}_1^{100}$. The choice of the hyperparameter k was obtained using cross-validation. Specifically, for a particular training noise and control algorithm, we picked a k from $\{10, 100, 10000\}$ to compute a RIM in the objective function and then compared it to a RIM computed using $k' = 10000$ different training perturbations $\{S^{(i)}_{\sigma_{\text{train}}}\}_1^{10^4}$ that comprise a large validation set during the optimization run for all 100 controllers. We found no significant empirical difference in error between the objective function RIMs for $k = 100$ and $k = 10000$. Note that the variance in the RIM decreases as $O(1/k)$ by the law of large numbers.

For the non-stochastic setting (ii), it can be seen from Fig. 5.8(a)-(c) (solid lines) that the average ARIM of all algorithms reduces asymptotically with the number of function calls. PPO attains the lowest final average ARIM values at 4×10^7 function calls for each σ_{train} but its final ARIM is not markedly better than the other algorithms considered here. However, this setting is quite expensive in terms of the total number of function calls.

For the stochastic setting (i) (dashed lines in Fig. 5.8(a)-(c)), increasing the training noise reduces the ability of the control algorithm to find robust controllers for all the σ_{train}, and the average ARIM is not minimized to the same extent as in setting (ii). This makes sense, since the stochastic objective (i) is a noisy fidelity with a reduced focus on robustness. The average ARIM is no longer reliably improved by any control algorithm in setting (i).

Within setting (i), we note that PPO converges to the lowest average ARIM value compared to the rest of the algorithms. This highlights the advantage of PPO as a stochastic optimizer that acquires robust controllers a smaller amount of samples compared to the rest of the control algorithms. This is theoretically justifiable as PPO optimizes a discounted RIM by design as mentioned earlier. Another thing to note is that lowest average ARIM values obtained by PPO in all three settings are similar, even though they are not attained at the same number of function calls. This suggests that PPO's ARIM performance can be made independent of the training noise levels, given an unconstrained number of objective function calls. However, this might be difficult to achieve completely since, even for PPO, there is a selection bias for low infidelity, but not low RIM. This manifests itself in the fact that the average ARIM starts increasing, albeit slowly, w.r.t. the number of function calls after the lowest average ARIM value is reached. Furthermore, we note that sharp transitions, like the stochastic PPO curves depicted in Fig. 5.8(b), are also typically reported in classical reinforcement learning contexts and are linked to sharp improvements in the reward by the algorithm [Raf+05].

To compare these results with the more standard noiseless fidelity maximization as a benchmark, we also plot the average ARIM for L-BFGS with a noiseless fidelity objective function and analytical gradient information. This version of L-BFGS accumulates sharp peaks in the fidelity landscape with more function calls since it is gradient-based and is effectively climbing to the sharpest peak in the fidelity landscape. Hence its average ARIM flatlines quickly w.r.t. function calls to a higher value compared to the other control algorithms in setting (i), with the exception of the forward differencing L-BFGS.

Contrasting settings (i) and (ii) for a single control algorithm, the point at which there is an advantage for non-stochastic optimization via setting (ii) is around 10^7 function calls for the algorithms, excluding L-BFGS with noise. For the regime below 10^7 function calls, setting (i) has a clear advantage over (ii) for PPO and SNOBFit.

5.4 Conclusions

In this chapter, we have presented the robustness-infidelity measure (RIM_p), a statistical generalization of the infidelity in the robustness sense, defined w.r.t. perturbations of arbitrary noise level S_σ in the fidelity function. We have used the RIM_p to quantify the robustness and fidelity of quantum controllers obtained from various control algorithms and have improved upon the qualitative MCRA comparisons in Chapter 4.

The RIM_p is the p-th order Wasserstein distance of the infidelity distribution induced by S_σ from some ideal distribution that is impervious to S_σ. We showed that the RIM_p is the p-th root of the p-th raw moment of the infidelity distribution and can be evaluated using perturbed fidelity function evaluations in physical experiments or Monte Carlo simulations. For $p = 1$, the infidelity measure RIM_1, reduces to the average infidelity. Further, by using a metrization argument, we justified why the RIM_1 is a practical robustness measure for quantum control problems due to the convergence of RIM_p values, for all p, given highly robust and high fidelity controllers. As such, it meshes well with the concept of the average infidelity, used as a robustness target already used in robust [LK09; Wu+19b] and stochastic/adaptive quantum control settings [Tur19; Che+14] and related to the concepts of the average gate fidelity and randomized benchmarking to extract circuit error rates per gate that was covered in Chapter 2.

Moreover, the RIM_p is also related to the risk-tunable fidelity measure using a utility function, as introduced in Ref. [GW21]. The RIM_1 also has a nice interpretation as the area under the curve of the cumulative distribution of the infidelity. Further analyses of the utility of the RIM_p for, e.g., the optimization of robustness, is an exciting direction to pursue for future work.

We also noted the connection of the RIM_p with classical control. In Sec. 5.2.4, we demonstrated the connection between the derivative of the RIM and the log-sensitivity that is a classical robustness measure in the LTI framework where asymptotic dynamics is gauged instead of the transient dyanmics that we are concerned with in quantum control. The frequency-domain limitations of classical feedback control have limited applicability in quantifying the performance in the time-domain. This restriction does not map directly to quantifying fidelity versus robustness in the quantum domain [SJL18; JSL18], although it is recovered in some cases [ONe+22b]. Classically, there is a conflict between minimum error and minimum sensitivity of the error quantified as $C(j\omega) + T(j\omega) = I$, where C is the tracking error and T the sensitivity of the error relative to *unstructured* uncertainties [SLH81]. Attempts to embed S and T in a single criterion have been proposed, e.g., the "mixed-sensitivity" [ZD98]. Thus the RIM_p may be viewed as a mixed-sensitivity approach for uncertainties *structured* by their probability distribution function (PDF) where both the error (or infidelity) and its robustness (the variance of its PDF) are encoded in the Wasserstein distance of order p.

Later, building on the theoretical foundations of the RIM we further generalized it to define an algorithmic RIM (ARIM) to compare the performance of control algorithms in terms of their ability to find robust high-fidelity controllers. Even though the RIM and ARIM are illustrated for static controls, they can be computed in any situation that generates a fidelity distribution over $[0, 1]$, including time-dependent controls and open quantum systems, enabling their use for further study for a wide range of practical quantum control problems.

We have used the RIM under model and controller noise in this chapter to quantify the performance, in terms of the robustness and fidelity, of individual controllers for excitation transfer in spin chains by energy landscape shaping. The controllers were obtained by four control algorithms (PPO, SNOBFit, Nelder-Mead, L-BFGS) at simulation noise scales of up to 10%. Using the RIM we found that high-fidelity controllers can vary widely in robustness to noise across all algorithms that we studied, although there are notable differences in algorithmic efficacy w.r.t. robustness, as indicated by the ARIM. We also demonstrate a consistency statistic that can be used to differentiate control algorithms by how correlated their controller infidelities are with the RIM$_1$. This provides a method to predict robustness via the RIM$_1$ without its explicit evaluation.

To compare the control algorithms, we studied their ARIM performance for multiple spin transfer problems. Under constrained function calls of a stochastic objective function (noisy infidelity), PPO performed better than SNOBFit and Nelder-Mead at certain, problem-specific training noise levels. SNOBFit performance at different training noise levels was similar, regardless of whether it was good or bad, suggesting that it is filtering out the noise. Nelder-Mead exhibits similarly consistent behavior across training noise levels with less than optimal performance for all but one problem. With unconstrained stochastic function calls, PPO showed excellent performance compared to the other algorithms, independent of the training noise level, since its reward accumulation strategy implicitly optimizes a discounted RIM.

In contrast, when optimizing the RIM$_1$ (average infidelity) over a fixed ensemble of perturbations, we found that all algorithms were capable of asymptotically finding an optimum. However, this approach is expensive in terms of the number of function calls compared to the aforementioned stochastic optimization setting with a noisy fidelity function as the objective. Our results also show that for stochastic settings, e.g., shot noise, PPO (or more generally reinforcement learning) is a promising approach to obtain robust controllers.

We now highlight some promising directions for future work. A limitation thus far has been that we require the computation of multiple controllers per control problem. In simulation, this further involves numerous time-consuming matrix exponential evaluations to generate a large number of samples per controller to approximate the RIM measure. More work is necessary to elucidate the fundamental limitations of the optimization landscape. Nevertheless, our statistical robustness approach is a useful tool that can be applied in a wide range of quantum control scenarios where analytic approximations with small and/or uncorrelated noise are unsuitable. For future work, it would be interesting to speed up the Monte Carlo sampling or controller sampling. The former could potentially be tackled by exploiting some structure in the calculations involving specific models or interesting approximations such as the Laplace approximation [Rip07]. The latter could involve more careful theoretical analyses of the control solutions and its manifold.

Furthermore, a limited number of control algorithms were benchmarked due to time limitations. It would be interesting to explore other algorithms presented in Chapter 2 like genetic algorithms that promise a more global exploration of the control landscape or Bayesian optimization that captures the essence of RL via its elegant Bayesian handling of the exploration-exploitation dilemma. In particular, if similarities between algorithms that share a similar meta-strategy or principle are found, a systematic isolation of the features of those algorithms e.g. via ablation studies would strongly highlight their strongly performing components. Once these are found, a hybrid algorithm can be constructed using these parts to streamline its overall performance while compensating for the weaknesses of the individual algorithms. Finally, it would be interesting to explore the scalability of Bayesian optimization and RL at controlling increasingly complex quantum systems such as those involved in VQAs. Since RL excels at mastering highly complex and combinatorially demanding gameplay and/or search, as evidenced by recent progress in optimizing sorting and matrix multiplication subroutines [Man+23; Faw+22], we conjecture that RL should be more successful at complex control tasks than Bayesian optimization based approaches but this needs to be tested in a falsifiable manner – preferably in a physical experimental setting.

Lastly, PPO, or model-free RL algorithms in general, have unrealized potential to be more sample efficient in terms of $\mathcal{E}$ calls or experimental resources. In the next chapter, we look at how to perform model-based RL control with an improved sample complexity over model-free RL.

Chapter 6

Sample-efficient reinforcement learning for control

As discussed in previous chapters, control of quantum devices for practical applications requires overcoming noise that degrades performance in real-time by finding robust controls unaffected by this noise. In Chapter 4, we showed that reinforcement learning (RL) approaches are more likely to find robust controls [Kha+21] at the cost of requiring large amounts of measurements from the quantum device (samples). In this chapter, we develop a model-based RL approach to address this problem of high sample complexity that makes it difficult to allow these model-free algorithms to be deployed.

Recall from Chapter 2, that typically, a quantum control problem is formulated as an open-loop optimization problem based on a model [Kha+05a; RNK12; Mac+11a; Koc+22]. The underlying assumption is that the model represents the system sufficiently accurately. This class of control algorithms has low sample complexity (high sample efficiency) as generally with an analytical model gradient information can be leveraged. This is a strong assumption, at least in the noisy intermediate scale quantum era where noise impedes perfect characterization of quantum devices. However, the approach has merit, since significant thought goes into modelling and engineering quantum devices [Wit+21]. Alternatively, RL seeks an optimal control scheme via interaction with the physical system, building learned models to various degrees as shown in Chapter 4.

RL approaches utilizing only measurements without prior information do not suffer from model bias. Also, as seen from Chapter 5, RL approaches usually optimize the average controller performance over the noise in the system, i.e., the RIM and yield inherently robust controllers [Kha+23b]. However, this means the number of optimization function calls becomes prohibitively large, and RL's high sample complexity is a core problem limiting its practical applicability [SB18b]. This is not surprising as without a prior model considerably less information is available to the optimization algorithm that must be obtained via measurements.

From Chapter 2, we know that high sample complexity is typically addressed using model-based RL. Such methods are successful if the model and the measurements (samples) obtained during training possess some generalizability [Chu+18; Jan+19] that is captured by a function approximator (usually a neural network). However, methods involving universal function approximation of dynamic trajectories are unstable. This is because learning can be hindered by the very large space of trajectories, and interpolating from insufficient sample trajectories can be shallow or incorrect [VHA19]. More importantly, for quantum data, it is known that a time-independent Hamiltonian can generate infinitely many unitary propagators[1], so estimating the model may imply learning the entire Hilbert space of propagators for a particular control problem which is often intractable. This motivates learning the dynamical generator, i.e., the Hamiltonian, instead of the propagators.

It has recently been shown that inductive biases, i.e. encoding the symmetries of the problem into the architecture of the model space, such as the translation equivariance of images in the convolution operation [Bro+21], leads to stronger out-of-distribution generalization by the learned model. This is because inductive biases impose strong priors on the space of models such that training involves exploring a smaller subset of the space to find an approximately correct model. In this chapter, we propose a model-based RL method for time-dependent, noisy gate preparation where the model is an ordinary differential equation (ODE), differentiable with respect to model parameters [Che+18]. ODE trajectories do not intersect [CL55; DDT19] which constrains the space of potential models for learning and makes learning robust to noise [Yan+19]. We parameterise the Hamiltonian by known time-dependent controls and unknown time-independent system parameters, which, in addition, makes the model interpretable. We show that combining the inductive bias from this ODE model with partially correct knowledge (assuming we know the controls, but not the time-independent system Hamiltonian) reduces the sample complexity compared to model-free RL by roughly at least an order of magnitude.

[1] i.e. there is a many-to-one correspondence between unitaries of the form $\exp\{-iHt\}$ and a time-independent Hamiltonian H (e.g. at different times t)

We demonstrate improvement over the sample efficient soft-actor critic (SAC) model-free RL algorithm [Haa+18] for performing noisy gate control in three settings that correspond to leading quantum computing architectures: nitrogen vacancy (NV) centers (one and two qubits) [HZS20], and transmons (two qubits) [MG20], subject to dissipation and single-shot measurement noise. We also show that the learned Hamiltonian can be leveraged to further optimize the controllers found by our RL method using GRAPE [Kha+05a; Mac+11a].

We focus on time-dependent (dynamic) gate control in this chapter intead of time-independent (static) state preparation that was the focus of prior chapters. This is because of two main reasons. Firstly, the shift from static to dynamic control is made because dynamic control is fully controllable [SFS01] and necessary for universal quantum computing [Deu85]. We focussed on gates instead of states since they are building blocks of larger quantum circuits that the quantum community works with to make various quantum technologies – some of which were discussed in Chapter 2.

However, dynamic control is not necessary for a solution to exist for a particular control problem as seen in prior chapters. Even though static control based energy landscape shaping is a novel and interesting paradigm with scope for robust control for multiple control problems, gate control using static controls is limited so both paradigms are important.

This chapter is organised as follows: Sec 6.1 describes the model-based version of the RL control framework, Sec 6.2 describes how the noise in the control problem was modelled and Sec. 6.3 presents numerical studies for some example control problems on the system architectures described above in noisy and ideal settings and how to leverage the learned system Hamiltonian using GRAPE.

6.1 Model-based reinforcement learning control

6.1.1 The RL'd quantum control problem

The general quantum gate control problem Eq. (2.19) can be represented as an RL problem by sequentially constructing the control amplitudes as actions, using the unitary propagator the control implements as the state with the reward as the fidelity:

$$\mathbf{a}_k = u_k, \tag{6.1a}$$

$$\mathbf{s}_k = \prod_{l=1}^{k} \exp\bigg(-\frac{i}{\hbar}\Delta t \mathbf{G}(t_l, u_l)\bigg), \tag{6.1b}$$

$$\mathrm{r}_k = \mathcal{F}(\mathbf{\Phi}(\mathbf{E}(\mathbf{u}_k)), \mathbf{\Phi}(\mathbf{E}_{\text{target}})). \tag{6.1c}$$

As this is deterministic, the probabilities $\mathcal{P}$ are trivial, and we have a simple environment function $\mathcal{E} : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathcal{R}$, mapping the current state and action (s, a) to the next state and reward (s', r).

For QOC, we are usually just concerned with finding an optimal action sequence $\mathbf{u}^*$ producing the maximum intermediate reward $\boldsymbol{r}_k$ rather than the optimal policy function π^* which can be produced by a sub-optimal policy, too.

6.1.2 Model-Based Reinforcement Learning for QOC

SAC can be augmented to incorporate a model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_k, \mathbf{a}_k)$ that approximates the dynamics of $\mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$ using the policy's interaction data $\mathcal{D}$ [Jan+19] where ζ are the model's learnable parameters. The model acts as a proxy for the environment and allows the policy to do MDP rollouts/steps to augment the interaction data. For this to work, the dynamics obtained from interacting with $\mathbf{M}_{\boldsymbol{\zeta}}$ must be close enough to the true dynamics of $\mathcal{E}$ to allow the policy to maximize J. Fig. 2.1 presents an illustration of model-based RL which in some sense is a superset of model-free RL.

By improving the returns $\hat{\eta}(\pi)$ on the model $\mathbf{M}_{\boldsymbol{\zeta}}$ by at least a tolerance factor that depends on this dynamical modelling error, the policy's true returns $\eta(\pi)$ on the environment are guaranteed to improve which we now illustrate next following a detailed mathematical discussion.

We show that it is possible to improve the environment's reward under an incorrect model $\mathbf{M}_{\boldsymbol{\zeta}}$. For that we need the following result from [Jan+19],

Theorem 6.1. *(Monotonic improvement for model-based returns [Jan+19]) Given k-branch rollout returns $\eta_{branch}(\pi)$ for a policy π under the model, the true returns $\eta(\pi)$ are lower bounded*

$$\eta(\pi) \geqslant \eta_{branch}(\pi) - 2r_{max}\left(\frac{\gamma^{k+1}\epsilon_\pi}{(1-\gamma)^2} + \frac{\gamma^k + 2}{1-\gamma}\epsilon_\pi + \frac{k}{1-\gamma}\left(\epsilon_{model}\right)\right) \tag{6.2}$$

where the returns η are defined as

$$\begin{aligned}\eta(\pi) &:= \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\,\mathrm{r}_t(\mathbf{s}_t,\mathbf{a}_t)\right]\\ &= \mathbb{E}_{\mathrm{r}_t\sim\mathcal{E}(\mathbf{s}_{t-1},\mathbf{a}_t^{\pi})}\left[\sum_{t=0}^{\infty}\gamma^{t}\,\mathrm{r}_t(\mathbf{s}_t,\mathbf{a}_t)\right]. \end{aligned} \tag{6.3}$$

r_{max} *is the maximum reward for an MDP transition; the policy error ϵ_π is the upper bound,*

$$\epsilon_\pi \geqslant D_{TV}(\pi_D(\mathbf{s},\mathbf{a})\|\pi(\mathbf{s},\mathbf{a})) \tag{6.4}$$

where D_{TV} is the total variation distance and π_D is the data generating policy (i.e., the policy that generated the MDP data by interacting with the environment $\mathcal{E}$). The model error ϵ_{model} is the upper bound

$$\epsilon_{model} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[D_{TV}(P_{\mathcal{E}}(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\|P_M(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a}))\right]\right), \tag{6.5}$$

where $P_M(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})$ is the MDP transition probabiltiy distribution under the model M that estimates the environment $\mathcal{E}$ and likewise for $P_{\mathcal{E}}$. γ is the discount factor and k is the branch rollout length.

Proof: See proof of Theorem 4.3 in [Jan+19]. □

Informally, the theorem states that as long as the returns under the model η_{branch} are improved by at least the tolerance term $2r_{\max}(\cdots)$, then the returns under the environment η are guaranteed to improve. This also assumes that the policy π generating the model returns is reasonably close to the policy that interacts with the environment to generate the MDP data that we use to compute the statistics, including the returns. The policy error ϵ_π can be monitored online and controlled while running the algorithm by curtailing its training once it exceeds some tolerance threshold. Moreover, Ref. [Jan+19] shows that as long as the dataset size is large enough, the model error ϵ_m can de decoupled from the policy error ϵ_π. The optimal branch rollout length k^* is given by the minimizer of the tolerance. In practice, there are other considerations (e.g., the interplay between various hyperparameters) that need to be accounted for to determine k^*, so it is usually obtained numerically via hyperparameter tuning.

Using Thm. 6.1 for the ODE model, we can indirectly connect the Hamiltonian error using the validation loss $L_{\text{model}}(\mathcal{D}_{\text{val}}))$ with ϵ_{model}. If the Hamiltonian error is small, then ϵ_{model} is small and the returns from the model and the environment are similar for any interacting policy π_θ. However, the returns need not be exactly the same and just need to be better than the tolerance provided by the term $-2r_{\max}(\cdots)$ in Eq. (6.2) which is a function of ϵ_{model}. The tolerance is smaller for a more accurate model and so less of an improvement of the model returns η_{branch} is necessary. The following lemma concretizes this idea by applying Thm. 6.1 to our RL control problem setup.

Lemma 6.2. *(Model error upper bound for the ODE model) If the model error ϵ_{model} upper bounds the risk,*

$$\epsilon_{model} \geqslant \max_t \left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a})) \right] \right) \tag{6.6}$$

then it also upper bounds the unitary prediction error

$$\epsilon_{model} \geqslant \max_t \left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[\left\| U_{\mathcal{E}(\mathbf{s},\mathbf{a})} - U_{\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a})} \right\|_{\infty,t} \right] \right) \tag{6.7}$$

and the total variation distance between the model and environment probabilistic distributions,

$$\epsilon_{model} \geqslant \max_t \left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[D_{TV} \left(P_{\mathcal{E}}(\mathbf{s}' \,|\, \mathbf{s},\mathbf{a}) \| P_{\mathbf{M}_{\boldsymbol{\zeta}}}(\mathbf{s}' \,|\, \mathbf{s},\mathbf{a}) \right) \right] \right). \tag{6.8}$$

Proof: Since the model $\mathbf{M}_{\boldsymbol{\zeta}}$ and the environment are both deterministic by assumption, we need to modify the lower bound on the model error ϵ_{model} in Thm. 6.1. We can replace the total variation distance between the two supposed distributions $P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}$ by an indicator variable $\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))$ if $\mathbf{s}'_{\mathbf{M}_{\boldsymbol{\zeta}}} \neq \mathbf{s}'_{\mathcal{E}}$, which is 1 if the transitioned states do not match and 0 if they do. We can upper bound the total variation distance like this since $D_{\text{TV}}(P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}) = \sup_A |P_{\mathcal{E}}(A) - P_{\mathbf{M}_{\boldsymbol{\zeta}}}(A)| \leqslant 1$ in case the probabilities do not match and $D_{\text{TV}}(P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}) = 0$ when they match perfectly. Hence, there exists some ϵ_{model} such that

$$\begin{aligned} \epsilon_{\text{model}} &\geqslant \max_t \left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a})) \right] \right) \\ &\geqslant \max_t \left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[D_{\text{TV}} \left(P_{\mathcal{E}}(\mathbf{s}' \,|\, \mathbf{s},\mathbf{a}) \| P_M(\mathbf{s}' \,|\, \mathbf{s},\mathbf{a}) \right) \right] \right). \end{aligned} \tag{6.9}$$

The risk $\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}} \left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a})) \right]$ is essentially the fraction of unitaries that the model predicts incorrectly and is related to the unitary error in Prop. 6.6 by the fact that

$$\left\| U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}} \right\|_{\infty,t} \leqslant \mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a})), \tag{6.10}$$

provided that $\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|_{\infty,t}$ is normalised to be in $[0,1]$. So we have

$$\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\left\|U_{\mathcal{E}(\mathbf{s},\mathbf{a})} - U_{\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a})}\right\|_{\infty,t}\right] \leqslant \mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))\right]. \tag{6.11}$$

So ϵ_{model} upper bounds the expected unitary error if and only if ϵ_{model} upper bounds the expected risk in the unitary prediction error. □

Notice again that a good choice of the model function class, therefore, can impose strong and beneficial constraints on the space of possible predicted dynamics and thus lead to a smaller modelling error and returns' tolerance factor or allow the model to reduce the tolerance factor greatly after consuming an appropriate amount of training data.

Our choice of the model's functional form is motivated by the two ideas presented in the introduction: (a) incorporating correct partial knowledge about the physical system in the model ansatz parameters; (b) encoding the problem's symmetries and structure into model predictions as function space constraints. For the system in Eq. (2.1) we assume that the controls are partially characterized to address (a). Specifically, its time-dependent control structure H_c is known. We achieve (b) by parametrizing the system Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ with learnable parameters $\boldsymbol{\zeta}$, where L is the number of qubits. We make the model $\mathbf{M}_{\boldsymbol{\zeta}}$ a differentiable ODE whose generator is interpretable and has the form

$$\begin{aligned} H_{\boldsymbol{\zeta}}(\mathbf{u}(t),t) &= H_0^{(L)}(\boldsymbol{\zeta}) + H_c(\mathbf{u}(t),t) \\ &= \sum_{l=1}^{n^2} \zeta_l P_l + H_c(\mathbf{u}(t),t) \end{aligned} \tag{6.12}$$

where $\zeta_l = \operatorname{Tr}[P_l H_0(t)] \in [-1,1]$ are real. Generally, like the Choi state, $H_0/\operatorname{Tr}[H_0]$ admits an arbitrary decomposition in terms of a basis $\{\mathbb{1}\}\cup\{P_l\}_{l=1}^{n^2-1}$ of SU(n) algebra. Analogously, for an open system, we parametrize the time-independent part of any dissipation dynamics in addition to the system Hamiltonian using an SU(n^2) algebra parametrization: $\mathbf{G}_0^{(L)}(\boldsymbol{\zeta}^{\text{diss}}) = \sum_l \zeta_l^{\text{diss}} P_l$ in the full generator $\mathbf{G}_{\boldsymbol{\zeta}}$.

The model is trained by minimizing the regression loss for single timestep predictions using data uniformly sampled, $D \sim \mathcal{D}$, where $\mathcal{D}$ represents the entire dataset,

$$L_{\text{model}}(D) = \sum_D \left(\mathbf{M}_{\boldsymbol{\zeta}}\left(\mathbf{s}_k, \mathbf{a}_k\right) - \mathbf{s}_{k+1}\right)^2. \tag{6.13}$$

To understand why a differentiable ODE ansatz is a good choice for the model, we need to define an ODE path that is given by $\phi_t : \mathbf{E}(0) \xrightarrow{H_{\boldsymbol{\zeta}}} \mathbf{E}(T)$ generated by $H_{\boldsymbol{\zeta}}$

Algorithm 2: Learnable Hamiltonian model-based soft actor critic (LH-MBSAC)

```
Input   :

$H_c$              control Hamiltonian (time-dependent part of $H(t)$ in Eq. (2.1))
$T, \Delta t, M$          max time, timestep size, number of shots (if open system to estimate $\mathbf{\Phi}$
                   using Eq. (2.17))
$\mathbf{E}_{\text{target}}$          target gate
$W, C, b, \texttt{tol}$     Epochs, timesteps, rollout length, validation loss tolerance
                   (which is a problem-specific hyperparameter)
Output:

$\mathbf{u}^*$        Approximately optimal 2D array of controls that solves Eq. (2.21)
$\theta,\ \phi,\ \boldsymbol{\zeta}$  Optimised parameters of the policy, critic and learned model
Initialize empty environment dataset $\mathcal{D}_{\mathcal{E}}$, model dataset $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$, random policy
 $\pi_\theta$
 // collect random model training data
Populate $\mathcal{D}_{\mathcal{E}}$ using random policy $\pi_\theta$ with Algorithm 1 without updates
 ▷ randomly explore the environment $\mathcal{E}$ state space
for $W$ epochs do
    // Train model
    Sample a batch of training and validation data $D_{\text{train}}, D_{\text{val}} \sim \mathcal{D}_{\mathcal{E}}$ and
      minimize $L_{\text{model}}(D_{\text{train}})$ in Eq. (6.13)
    for $C$ timesteps do
        // agent-environment interaction
        Execute $\mathbf{a}_k \leftarrow \pi_\theta(\cdot|\,\mathbf{s}_k)$, observe $\mathbf{s}_{k+1}, \mathrm{r}_k \leftarrow \mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$ and store data
          $\mathcal{D}_{\mathcal{E}} \cup \{(\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k)\}$
        if $L_{model}(D_{val}) < \texttt{tol}$ then
            // agent-model interaction
            Sample uniformly a batch of initial states $\{\mathbf{s}_k\} \sim \mathcal{D}_{\mathcal{E}}$, $k \leftarrow 0$
            for $k'$ in $\{1, \cdots, b\}$ do
                Execute $\mathbf{a}_{k'} \leftarrow \pi_\theta(\cdot|\,\mathbf{s}_{k'})$ and observe $\mathbf{s}_{k'+1}, \mathrm{r}_{k'} \leftarrow \mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k'}, \mathbf{a}_{k'})$
                 ▷ $b$-length model rollout
                Store $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \leftarrow \mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \cup \{(\mathbf{s}_{k'}, \mathbf{s}_{k'+1}, \mathbf{a}_{k'}, \mathrm{r}_{k'})\}$
                $k' \leftarrow k' + 1$,
        Train policy by minimizing $J'(\pi_\theta)$ in Eq. (2.66) using $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \cup \mathcal{D}_{\mathcal{E}}$
```

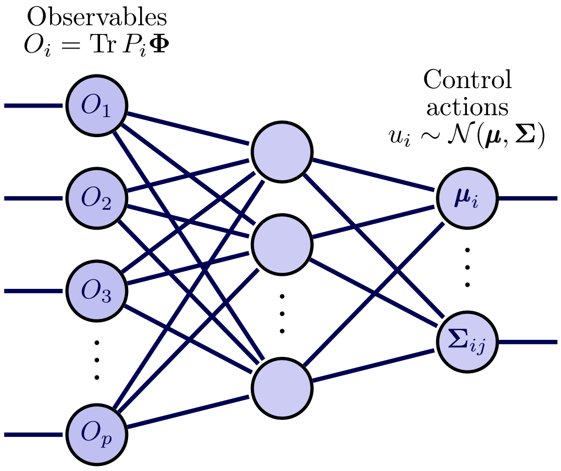

Observables
$O_i = \operatorname{Tr} P_i \boldsymbol{\Phi}$
O_1
O_2
O_3
O_p
Control
actions
$u_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\boldsymbol{\mu}_i$
$\boldsymbol{\Sigma}_{ij}$

Figure 6.1: **Policy function** $\pi_\theta(\mathbf{a}_k \,|\, \mathbf{s}_k)$: we visualize the policy inputs as the gate (unitary or Lindblad) characterizing observables about the Choi matrix $\boldsymbol{\Phi}$ given by Eq. (2.17) and the tunable outputs are the parameters of a multivariate Gaussian distribution, i.e., the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The controls u_i are drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

for some time $t \in [0, T]$ and propagator $\mathbf{E}$. The ansatz is a good choice because of the following two properties of ODE paths: (a) they do not intersect and (b) if paths $\phi_0^{(A)}$, $\phi_0^{(B)}$ start close compared to path $\phi_0^{(C)}$, then paths $\phi_t^{(A)}$, $\phi_t^{(B)}$ remain close compared to path $\phi_t^{(C)}$. We now present both facts more rigorously to better motivate the ansatz.

First, we present a simple proof (see for example [CL55; DDT19]) for why ODE trajectories do not intersect.

To start off, we define the ordinary differential equation as follows:

Definition 6.3.

$$\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t), t) \quad \mathbf{z}(0) = \mathbf{z}_0, \quad \mathbf{z}(T) = \mathbf{z}_T \tag{6.14}$$

where $f_\theta : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ is the vector field map parametrized by some learnable parameter vector θ. And $\mathbf{z} : [0, T] \to \mathbb{R}^d$ is the solution. The initial and final conditions on the state $\mathbf{z}$ are given by $\mathbf{z}_0$ and $\mathbf{z}_T$.

An ODE path or trajectory is, formally, defined by the path $\phi_t : \mathbf{z}(0) \xrightarrow{f_\theta} \mathbf{z}(T)$ or $\phi_t(\mathbf{z}_0) = \mathbf{z}(t)$ for some $t \in [0, T]$. This is the differential evolutionary path taken by the state $\mathbf{z}(t)$ prescribed by f_θ.

Theorem 6.4. *(ODE trajectories do not intersect) Let* $\mathbf{z}_t^{(1)}, \mathbf{z}_t^{(2)}$ *be two solutions to the ODE problem in Eq.* (6.14) *with different initial conditions* $\mathbf{z}_0^{(1)} \neq \mathbf{z}_0^{(2)}$. *Then,* $\forall t \in (0, T]$, *we have that* $\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)}$

Proof: The proof follows by Picard's existence theorem [CL55]. Essentially, it states that there exists a unique differentiable $\mathbf{z}_t$ that solves in Eq. (6.14). Suppose there is some $t' \in (0, T]$ where $\mathbf{z}_{t'}^{(1)} = \mathbf{z}_{t'}^{(2)}$. Solve it backwards in time to obtain $\mathbf{z}_0^{(1)} = \mathbf{z}_0^{(2)}$ at $t = 0$ which is a contradiction. □

This formalizes property (a) straightforwardly. Next we present a technical fact from which property (b) becomes more apparent.

Theorem 6.5. *(Gronwall's inequality [You10; How98]) Let* $f_\theta : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ *be a continuous function and let* $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ *obey the ODEs:*

$$\frac{d\mathbf{z}^{(1)}(t)}{dt} = f_\theta(\mathbf{z}^{(1)}(t), t) \quad \mathbf{z}^{(1)}(0) = \mathbf{z}_0^{(1)}, \quad \mathbf{z}^{(1)}(T) = \mathbf{z}_T^{(1)}$$
$$\frac{d\mathbf{z}^{(2)}(t)}{dt} = f_\theta(\mathbf{z}^{(2)}(t), t) \quad \mathbf{z}^{(2)}(0) = \mathbf{z}_0^{(2)}, \quad \mathbf{z}^{(2)}(T) = \mathbf{z}_T^{(2)}$$

Assume that there exists a constant $C \geqslant 0$ *such that*

$$\|f_\theta(\mathbf{z}^{(2)}(t), t) - f_\theta(\mathbf{z}^{(1)}(t), t)\| \leqslant C\|\mathbf{z}^{(2)}(t) - \mathbf{z}^{(1)}(t)\| \tag{6.15}$$

Then, for $t \in [0, T]$,

$$\|\mathbf{z}^{(2)}(t) - \mathbf{z}^{(1)}(t)\| \leqslant e^{Ct}\|\mathbf{z}_0^{(2)} - \mathbf{z}_0^{(1)}\|$$

Proof: See [How98]. This is a special case of the general proof where both ODEs evolve according to different f_θ f'_θ. □

Both properties are well known [You10; How98] for ODEs and become very useful when we try to predict the trajectories from noisy quantum data by imposing strong priors on the space of learnable Hamiltonians. Property (b) is a consequence of Gronwall's inequality [How98] and essentially can be interpreted as: ODE flows that start off closer (w.r.t. the initial condition) stay closer (w.r.t. the final condition). Both (a) and (b) essentially imply a sort of intrinsic robustness of the ODE flow $\phi_t(\mathbf{z}_0)$ to perturbations on $\mathbf{z}_0$ [Yan+19]. They constrain the trajectories predicted by the model $\mathbf{M}_{\boldsymbol{\zeta}}$ to be intrinsically robust to small noise in the states $\mathbf{s}_k$ and inaccuracies in the learned system Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$.

We call the SAC equipped with this differentiable ODE model the learnable Hamiltonian model-based SAC (LH-MBSAC) as listed in Algorithm 2. Crucially, we note that LH-MBSAC generalizes the SAC by allowing the policy to interact with the ODE model and the physical system. LH-MBSAC gracefully falls back to the model-free SAC in the absence of a model with low prediction error that is measured from the performance of the model's predictions on an unseen validation set of interaction data. In this case, the model is no longer used/updated and RL control is model-free as shown in previous chapters. Note that the threshold or tolerance level for switching to the agent-model interaction part of the algorithm is likely problem-dependent and thus needs to be selected along with other hyperparameters in RL. However, this allows us to improve the sample complexity of model-free reinforcement learning when possible, by leveraging knowledge about the controllable quantum system, yet still be able to control the system in a model-free manner if this is not possible.

6.2 On modelling practical noise sources

Before we proceed to the experiments' section, we need to consider noise sources that plague NISQ devices and how they might be modelled in simulation. Realistically the two most significant noise sources are errors in the system characterization of H and finite precision from single shot measurements.

6.2.1 Finite measurements

The number of single-shot measurements for estimating the realized gate Φ using finite measurements with error $\epsilon \geqslant 0$ and probability $1-\alpha$ is $O(3^k \frac{\log \frac{1}{\alpha}}{\epsilon^2})$ by Hoeffding's inequality using up to two qubit Pauli measurements [FL11b]. Here, k is $\log_2 \operatorname{rank}(\Phi) = 2n$ where n is the number of qubits. We model this by writing the Choi state in terms of the generalized Pauli basis and sampling POVM measurements from a rescaled binomial distribution per POVM operator. For example, we can expand any n-qubit quantum state $\rho \in \mathbb{SU}(2^n)$ in a tensored unnormalized Pauli basis $\{P_i\}_{i=1}^{4^n} = \{\mathbb{1}, X, Y, Z\}^{\otimes n}$ as follows,

$$\rho = \frac{1}{2^n} \sum_{i=1}^{4^n} \operatorname{Tr}\left[P_i \rho\right] P_i. \tag{6.16}$$

Note that the Pauli operator is a physical observable with binary eigenvalues in $\{+1, -1\}$ which is what we observe after making the right adjustments to the operator, e.g., $3P_i - 1$ to make it positively valued. Note that $\mathrm{Tr}\,[P_i \rho] \in [-1, 1]$. We make the adjustment so that $p_i = 2\mathrm{Tr}\,[P_i\rho] - 1 \in [0, 1]$ can be treated as a binomial variable $\mathrm{Bin}(M, p_i)$ where M is the number of single shot Bernoulli (single shot) experiments and $p_i = \mathbb{E}_{P^{(k)}}[\mathbb{I}(\text{Outcome of } P_i^{(k)} = +1)]$ their expectation. Thus, the estimator $\hat{\rho}$ is determined by a mixture of 3^k non-redundant binomial estimators $\hat{p}_i$.

6.2.2 System noise

The Hamiltonian parameters can be subject to control dependent and independent noise. We can model this by adding a random noise perturbation to the control parameters and the system parameters at each timestep that are all i.i.d. This is done explicitly in Sec. 4.1.1 and used in the previous chapters. However, sometimes, a dominating noise source can wash out the effects of other noise sources. For this reason, we do not consider stochastic noise in the Hamiltonian in this chapter and only consider finite single-shot measurement noise since that is the dominating noise in our simulations.

6.3 Experiments

We demonstrate the performance of LH-MBSAC on three quantum systems of current interest in open and closed settings with shot noise. Measurements in this section are made using Pauli instead of the generalized Gell-Mann operators mentioned in Chapter 2 and the simulated systems are all qubit systems. The proceeding descriptions are kept general but in all places a truncation to two level systems is applied in numerical simulations.

To warm up, the first system $\tilde{H}^{(1)}_{\mathrm{NV}}$ is a single-qubit NV center with microwave pulse control [Fra+17b],

$$\frac{H^{(1)}_{\mathrm{NV}}(t)}{\hbar} = 2\pi\Delta\sigma_z + \underbrace{2\pi\Omega\,(u_1(t)\sigma_x + u_2(t)\sigma_y)}_{H_c(t)}, \tag{6.17}$$

where $\Delta = 1$ MHz is the microwave frequency detuning, $\Omega = 1.4$ MHz is the Rabi frequency and the control field parameters are $u_j(t)$ in the range $\mathbb{X}^{(1)}_{\mathrm{NV}} = \{-1 \leqslant$

$u_j \leqslant 1\}$. In this and subsequent examples terms not covered by $H_c(t)$ are learned, parametrized by the learnable model parameters $\boldsymbol{\zeta}$. The gate operation time is 20 μs.

The second system H_{NV} is again NV center based, but for two qubits [HZS20]. This system is realised in the system subspace using microwave pulses of approximately 0.5 MHz and is given by

$$\begin{aligned}\frac{H^{(2)}_{\mathrm{NV}}(t)}{\hbar} &= |1\rangle\langle 1| \otimes \left(-\left(\nu_z + a_{zz}\right)\sigma_z - a_{zx}\sigma_x\right) \\ &\quad + |0\rangle\langle 0| \otimes \nu_z\sigma_z + \underbrace{\sum_{l=x,y}\sum_{k=1}^{2}\sigma_k^{(l)}u_{lk}(t)}_{H_c(t)},\end{aligned} \tag{6.18}$$

where $\nu_z = 0.158$ MHz, $a_{zz} = -0.152$ MHz and $a_{zx} = -0.11$ MHz, $\sigma_j^{(l)}$ is the lth Pauli operator on qubit k, and $u_{lk}(t)$ is a time-dependent control field. The range of control is $\mathbb{X}^{(2)}_{\mathrm{NV}} = \{-1 \text{ MHz} \leqslant u_{lk} \leqslant 1 \text{ MHz}\}$ and the final gate time is $T = 20$ μs.

The third system $\tilde{H}^{(L)}_{\mathrm{tra}}$ is an effective Hamiltonian model for cavity quantum electrodynamics (cQED) [MG20] for two transmons/qubits as a proxy for the IBM quantum circuits [Cro18],

$$\begin{aligned}\frac{H^{(2)}_{\mathrm{tra}}(t)}{\hbar} &= \sum_{l=1}^{2}\omega_l\hat{b}_l^\dagger\hat{b}_l + \frac{\eta_l}{2}\hat{b}_l^\dagger\hat{b}_l(\hat{b}_l^\dagger\hat{b}_l - \mathbb{1}) \\ &\quad + J\sum_{l=1}^{2}(\hat{b}_l^\dagger\hat{b}_{l+1} + \hat{b}_l\hat{b}_{l+1}^\dagger) + \underbrace{\sum_{l=1}^{2}u_l(t)(\hat{b}_l + \hat{b}_l^\dagger)}_{H_c(t)}\,.\end{aligned} \tag{6.19}$$

This model is comprised of Duffing oscillators with frequency $\omega_j = 5$ GHz representing the qubits with an anharmonicity $\kappa_j = 0.2$ GHz, qubit coupling J, and a control field per qubit Δ_j. Note that this a special case of the Bose-Hubbard model [KWM00] with $\hat{b}_j$ representing the boson annihilation operator on the jth qubit. The control field $\mathbf{u}_j(t) = \Delta_j(t)$ is real by construction in addition to extra constraints imposed on the space of possible controls $\mathbb{X}$. The range of control is given by $\mathbb{X}^{(2)}_{\mathrm{tra}} = \{-0.2 \text{ GHz} \leqslant \Delta_{ij} \leqslant 0.2 \text{ GHz}\}$ and the final time is $T = 20$ μs.

We demonstrate the results of LH-MBSAC, benchmarked against its model-free counterpart, SAC, for a one- and two-qubit NV center $H^{(1)}_{\mathrm{NV}}$, $H^{(2)}_{\mathrm{NV}}$ and the two-qubit transmon $H^{(2)}_{\mathrm{tra}}$. For the two-qubit system, the target gate is the CNOT and for the one-qubit system, it is the Hadamard gate. Pulses are discretized in accordance with the scheme introduced in Sec. 2.1.2.3 for a number of timesteps, $N = 20$. We follow the parameter restrictions for all systems introduced in Ref. [Wit+21; MG20; HZS20; Fra+17b]. Moreover, due to limited support in our autodifferentiation library [Pas+19], we simulate the complex dynamics by mapping the complex ODE to two real coupled ODEs [Leu+17] (see Appendix B.2 for more details on our ODE solver).

The following sections are organized as follows. In Sec. 6.3.1, we demonstrate a sample complexity improvement for the different control problems discussed above in a noisy closed setting. For the subsequent sections, we study the two-qubit transmon control problem in more detail. The results were similar for other systems that we studied. In Sec. 6.3.2, we study the effect of increasing the estimated Hamiltonian error from its true value on the sample complexity of control. Sec. 6.3.3 discusses how the learned Hamiltonian in LH-MBSAC can be further utilized for model-based control using gradient-based methods like GRAPE. Sec. 6.3.4 extends results from the closed setting to the noisy open system setting. Finally, in Sec. 6.3.5, we highlight some limitations and silver linings of the LH-MBSAC and the RL-for-control approach for our specific MDP (Eq. (6.1)) in this chapter and provide promising ideas to circumvent some of the issues.

6.3.1 Sample Efficiency for Closed System Control

We only consider closed system control for the single-shot measurements (or shots) in Eq. (2.19) and Eq. (2.10). Note that the Choi operator is still used to estimate the propagator using single shot measurements and is discussed in detail below. Unitary control (with closed system dynamics) is implemented for shots as a special case of open system control where the dissipation operators $\mathfrak{L}$ are 0.

The Choi operator $\mathbf{\Phi}$, corresponding to the gate realised by the controls, is obtained by sampling from the binomial distribution in Eq. (2.17) with $M = 10^6$ shots per measurement operator. By Hoeffding's inequality, we know that with probability $1 - 0.01$ the error in the estimator of q_l is 10^{-3}. Or generally, with probability $1 - \alpha$, for ϵ error, we require $O(\log \frac{1}{\alpha}/\epsilon^2)$ measurements. Furthermore, we pick the number of shots M to estimate the gate for the noisy control problem by understanding how the unitary error scales with the number of shots as δ, the Hamiltonian error is increased. We define the Hamiltonian or model error δ as in Ref. [Bur+22]:

$$\delta = \|H_0(\boldsymbol{\zeta}) - H_0\| \tag{6.20}$$

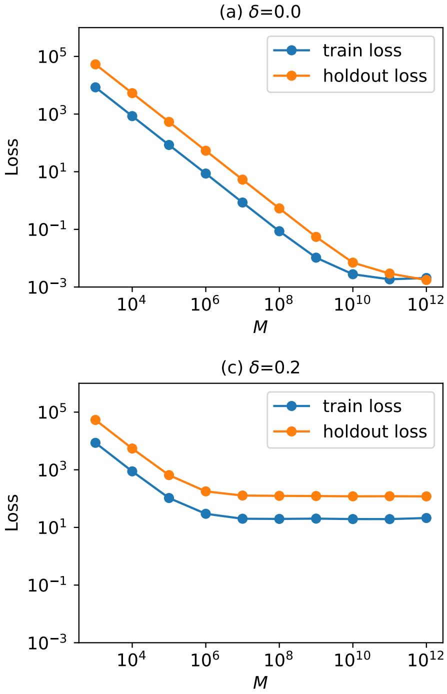


Figure 6.2: Unitary error scaling as a function of the number of shots M per observable for the two-qubit transmon unitary control problem with shot noise. Training is done for about 100 epochs for each point. Larger errors in the system Hamiltonian lead to smaller reductions in unitary error as the shot precision is improved. For the control-with-shot-noise case, a shot size of 10^6 is optimal i.e. where we get the elbow of dimishing returns to pick for different Hamiltonian errors $\delta > 0$.

where $\| \cdot \|$ is the spectral norm (the largest singular value) of $H_0(\boldsymbol{\zeta}) - H_0$.

We randomly initialize the learnable system Hamiltonian using the Pauli basis parametrization in Eq. (6.12) with coefficients $\zeta_i \sim \text{Uniform}(-1, 1)$. The perturbations on each coefficient ζ_i to create the noisy Hamiltonian are sampled from a Gaussian using rejection sampling until a desired value of δ is achieved.

We trained for a large number of epochs (100) and observed the final train and holdout loss of the model's unitary predictions for the two-qubit transmon unitary control problem with shots. The results are shown in Fig. 6.2. Larger Hamiltonian error results in larger unitary losses and there is no benefit of increasing the shot precision past around 10^6, where the Hamiltonian error contribution to the unitary prediction error starts to dominate.

The AAPT protocol uses $M \times 3^L$ shots in total for 3^L possible measurement operators, which is quite expensive.

Further sparsity restrictions on the structure of $\mathbf{\Phi}$ imposed by a k-local Hamiltonian, where qubit interactions up to only the nearest $k \leqslant L$ qubits are assumed, can allow the shot cost to go down to $O(4^k(\log M)/\epsilon^2)$ for M observables due to a reduction in the number of observables that need to be measured/tracked which is asymptotically optimal in the number of measurements [HKP20]. However, since our goal is gate control, these costs are generally unavoidable to completely verify gate performance. In practice, such gates are only limited to a few qubits and operations on many qubits are achieved in the circuit formalism through gate composition [NC10; Cro18].

The environment's data buffer $D_{\mathcal{E}}$ that stores the model's training data, i.e., the initial exploration dataset (see Algorithm 2), consists of 1, 20, and 100 pulse sequences for the one-qubit NV, two-qubit NV and two-qubit transmon systems respectively. These data are collected using random uniform policy actions during the first run of the LH-MBSAC algorithm.

The exploration dataset is then used to learn the system Hamiltonians $H_{0_{\mathrm{NV}}}^{(1)}$, $H_{0_{\mathrm{NV}}}^{(2)F}$, $H_{0_{\mathrm{tra}}}^{(2)}$ via supervised learning of $\mathbf{M}_{\boldsymbol{\zeta}}$ using the dynamics prediction loss function (Eq. (6.13)) until a validation loss of around $10^{-3} \times 2^{2q} \times \texttt{batch\_size}$ is reached, where `batch_size` is the number of samples used for a single training policy update. Here, q is the number of qubits and $q = 2$ for the theoretical unitary and $q = 4$ for the Choi state (due to the Choi-Jamiolkowski isomorphism in AAPT).

After this, we switch to the model $\mathbf{M}_{\boldsymbol{\zeta}}$ to generate synthetic samples to train the policy π. Whilst concurrently maintaining policy interactions and attempting control of the system via the policy π, the model is successively trained in periods with fresh data to reduce the model error even further. Once the policy starts producing pulses with nearly optimal fidelities of around 0.98, we terminate the algorithm and use the learned Hamiltonian to further optimize the pulses using gradient-based methods like GRAPE to (a) reduce sample complexity costs and (b) improve runtime of LH-MBSAC, since the model simulations are computationally expensive. We found that terminating around 0.98 ensures that the application of further gradient-based methods does not cause the control parameters to diverge too much from their initial values thereby retaining, at least partially, their favourable robustness properties [Kha+23b]. Step (b) is discussed in detail in Sec. 6.3.3.

The results for LH-MBSAC and model-free SAC for the one- and two-qubit control problems are shown in Fig. 6.3. We consider LH-MBSAC's performance with shots by estimating the gate using its corresponding estimated Choi state $\mathbf{\Phi}$ using AAPT with 10^6 shots per observable. The sample complexity of LH-MBSAC to achieve a

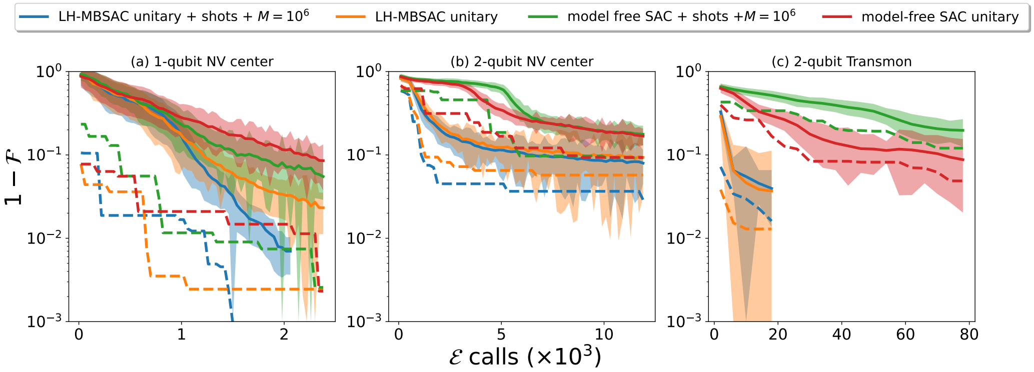


Figure 6.3: The closed system fidelity $\mathcal{F}$ of the Hadamard gate for (a) $H_{\text{NV}}^{(1)}$, and of the CNOT gate for (b) $H_{\text{NV}}^{(2)}$ and (c) $H_{\text{tra}}^{(2)}$ as a function of the number of environment $\mathcal{E}$ calls. The mean fidelity over 100 controllers is plotted as a solid line with the shading indicating two standard deviations, and the maximum fidelity is indicated by the dashed line. LH-MBSAC or model-free SAC with the unitary tag indicates the shot-noise-free closed system problem in Eq. (2.10) and single shot measurements are indicated likewise. We terminate the algorithm early at $\mathcal{F} > 0.98$ for LH-MBSAC with and without single shot measurements since the model simulations are expensive and the learned model at this point can be used to further optimize the moderately high fidelity RL pulses further as shown in Sec. 6.3.3. The sample complexity of LH-MBSAC is significantly improved for the two-qubit transmon and the NV center over model-free SAC for the closed system control problem and with single shot measurements (of size $M = 10^6$), using AAPT. We average these results over three seeds of each algorithm run where a seed refers to a single algorithm run from scratch with a fresh set of randomly initialized parameters.

maximum fidelity significantly improves, by at least an order of magnitude, upon the model-free baseline in both cases, although it is more significant for the two-qubit transmon.

The size of of the exploration dataset needs to be chosen by considering the exploration vs. exploitation dilemma of RL. It is part of the set of hyperparameters for the LH-MBSAC algorithm which effectively decide how much the learned model $\mathbf{M}_\zeta$ should be exploited after the training phase is complete and we are reasonably confident in the model's predictions. To that end, we again considered the two-qubit transmon unitary control problem, since we found that results regarding hyperparameters generalize to other problems for our algorithm[2], with a model that is perfect and has essentially no prediction error. A grid search over the number of model rollouts and model training iterations was considered while we evaluated LH-MBSAC's

[2] a sign of good inductive bias for the problem family

sample complexity performance normally as before. Recall that the model rollouts are the number of MDP transitions the RL policy π makes using $\mathbf{M}_{\boldsymbol{\zeta}}$ instead of the true environment and is a proxy for exploration. The model training iterations refer to the optimizer steps taken (we use Adam [KB14] with a learning rate of 3×10^{-3}) that update the policy parameters using stochastic gradient descent w.r.t. the unitary loss. This is a proxy for exploitation. As expected, we find no ceiling in the agent's sample complexity performance w.r.t. both exploration and exploitation using the perfect model. The highest rollout length `rl` (explore), train iterations `mtit` (exploit) combination $\texttt{rl}, \texttt{mtit} = 10, 100$ in Fig. 6.4(h) optimizes the fidelity fastest.

Moreover, we observe that a balance is necessary between the two processes as exploring too much without exploitation results in inaction (see Fig. 6.4(i)) on the part of the agent and too much exploitation leads to overfitting (see the first row in Fig. 6.4(a-d)). Moreover, practically we observe diminishing returns for larger magnitudes of $\texttt{rl}, \texttt{mtit}$. Due to this fact and that both processes are expensive in terms of LH-MBSAC's wall time, we chose $\texttt{rl} = 5, \texttt{mtit} = 10$ (see Fig. 6.4(g)) as a compromise between leveraging $\mathbf{M}_{\boldsymbol{\zeta}}$ to improve the policy's sample complexity and reduce the wall-time of our numerical simulations. Hence, our aforementioned choice of exploration datasets is chosen with this empirical utility in mind.

How much training data is really needed for model training or Hamiltonian learning? A hallmark for a good ansatz for the model $\mathbf{M}_{\boldsymbol{\zeta}}$ estimating the dynamics of the controllable system would be less demand of supervised learning MDP data needed for low prediction error.

We consider the Hamiltonian error, unitary train and holdout error. Hamiltonian error δ is the spectral norm error between the learned and true system Hamiltonian. The others are mean squared errors. Cross-validation is used to estimate the model's generalization ability on a holdout dataset of unseen random unitary data, also sampled from the MDP transitions and collected by the policy π during training.

As seen from Fig. 6.7, for the two-qubit transmon control problem, for very small dataset sizes comprising 20 to 200 unitary transitions, the single step unitary prediction error is large compared to training with about $2,000$ unitaries or about 100 full length pulses with 20 timesteps, though the decrese in error is diminishing with dataset size. All errors are in agreement across the datasets over 200 training epochs. This is further corroborated by Fig. 6.5 where the final errors after 200 epochs are plotted. There is a reduction in the final errors for the $2,000$ dataset size, but the improvement is diminishing in magnitude and plataeus at this loss for larger dataset

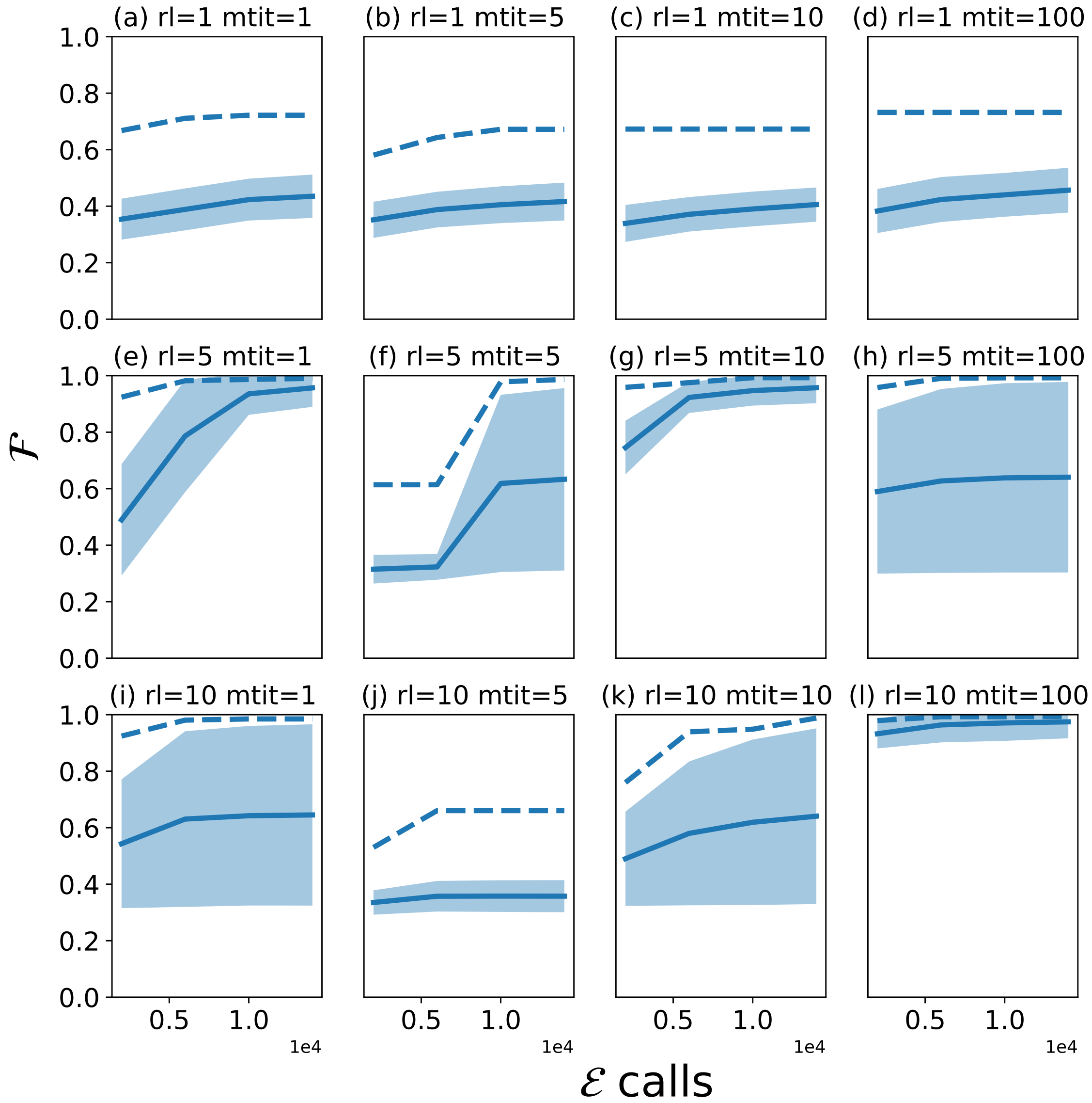


Figure 6.4: Model rollout length (rl) and model train iterations (mtit) for an LH-MBSAC with the perfect model. We study the tradeoff between training (exploitation) and the rollout length. For the perfect model, there is no ceiling. However, in practice, we find that the returns diminish in terms of rollout length and training iterations: compare for example (g) and (i).

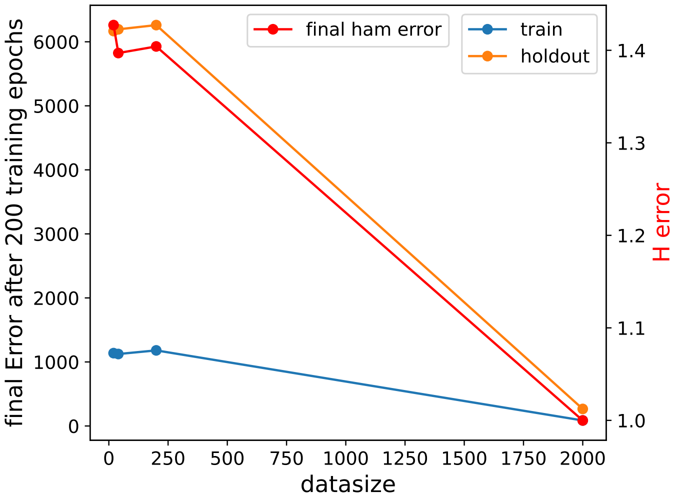


Figure 6.5: Effect of training data size on model generalization metrics: Hamiltonian error, unitary training $L_{\text{model}}(\mathcal{D}_{\text{train}})$ and validation (holdout) loss $L_{\text{model}}(\mathcal{D}_{\text{val}})$ for the noisy shots based unitary control of the transmon. The line with the same shade as the right axis represents Hamiltonian error.

sizes. This is still much less than what was required to train a neural network model for $\mathbf{M}_{\boldsymbol{\zeta}}$ during the initial stages of our research where the training dataset size needed to be of the order of 10^6. Moreover, these experiments provide us with an idea of what dataset size to use to train the model $\mathbf{M}_{\boldsymbol{\zeta}}$ by setting the number of initial exploration MDP transitions to add to the policy's buffer for the transmon control problem. We also adopted multiple training phases to continuously train $\mathbf{M}_{\boldsymbol{\zeta}}$ using fresh batches of training data collected by the policy.

The exploration dataset is used to learn the system Hamiltonians $H^{(2)}_{0_{\text{tra}}}$, $H_{0_{\text{NV}}}$ via supervised learning of $\mathbf{M}_{\boldsymbol{\zeta}}$ until a validation loss of around $10^{-3}\times 2^{2q}\times$`train batch size` is reached after which we switch to the model $\mathbf{M}_{\boldsymbol{\zeta}}$ to generate synthetic samples to train the policy π. Note that here q is the number of qubits and $q = 2$ for the theoretical unitary and $q = 4$ for the Choi state (due to the Choi-Jamiolkowski isomorphism in AAPT).

6.3.2 Sample Complexity as a Function of Hamiltonian Error

Continuing with the closed system control problem, in this section, we study the relationship between sample complexity and error in the estimated model Hamiltonian $H_0(\boldsymbol{\zeta})$ compared to the true system Hamiltonian H_0 as the error is increased. This

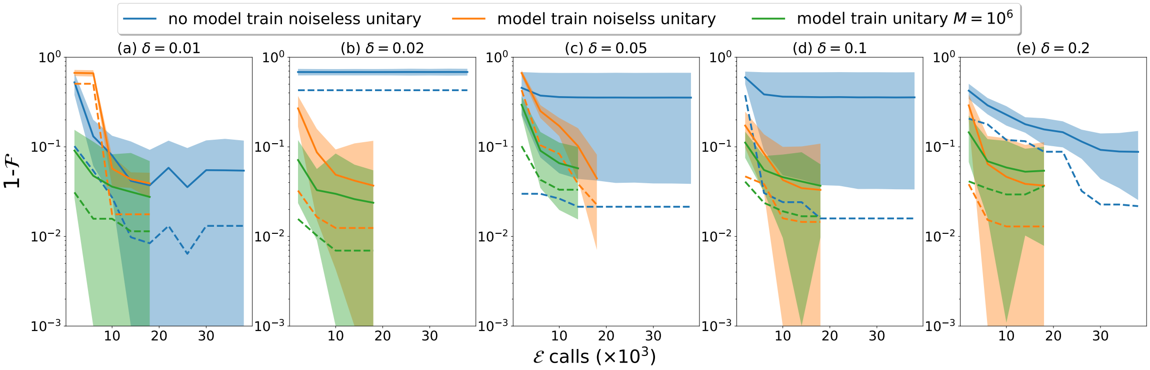


Figure 6.6: Sample complexity or $\mathcal{E}$ calls of LH-MBSAC for the two-qubit transmon control problem as a function of spectral norm error δ, quantifying closeness of the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ and the true system Hamiltonian H_0. The cases for $\delta = 0.01, 0.02, 0.05, 0.1, 0.2$ are plotted in (a)–(e). The mean fidelity over 100 controllers is plotted as a solid line with the shading indicating two standard deviations and the maximum fidelity is indicated by the dashed line. The 'noiseless unitary' is the no-shot noise setting where the exact unitary is seen by the algorithm while, alternatively, the unitary is estimated using AAPT with $M = 10^6$ shots per observable characterizing the Choi state. The 'no model train' line indicates the setting where no learning of $H_0(\boldsymbol{\zeta})$ occurs and δ is fixed while the 'model train' lines denote the setting where δ is reduced through model training. In general, we see that there are some instances where the RL agent is able to optimize the objectively wrong model $\delta = 0.2, 0.01$ and there is a non-linear dependence of $\mathcal{E}$ calls on δ, i.e., a large δ can produce better model-predictive trajectories with a smaller unitary prediction error. This points us to consider the idea of learning Hamiltonians that are only 'locally consistent'. Once learning $H_0(\boldsymbol{\zeta})$ is enabled, algorithmic performance is restored in both the noiseless (with no shot noise) and shot-noise unitary settings. The number of measurements is $M = 10^6$ per observable.

relationship is highly non-linear/irregular and is discussed in detail later in the section. On a high level, the purpose of this section is to understand the interplay between control and model learning especially if the model is inaccurate. Can we still learn a near optimal control policy even if the model is incorrect? To an extent, yes: we show that when the model error is small, LH-MBSAC is able to successfully find a near optimal control pulse, even with an incorrect model.

For this study, we compare two settings for some value of the Hamiltonian or model error δ in each experimental run: (i) *learning the system Hamiltonian*, i.e., δ is decreased from its initial value; (ii) *not learning the system Hamiltonian*, i.e., δ remains fixed throughout the experiment. Case (ii) effectively corresponds to Algorithm 2 without any model training, i.e., we do not attempt to minimize $L_{\text{model}}(D_{\text{train}})$ to update the model and instead set the model to have a fixed constant Hamiltonian error δ. The range of Hamiltonians corresponding to different δ values are chosen by randomly sampling the true Hamiltonian with rejection using Gaussian perturbations. The non-linear dependence on the sample complexity of LH-MBSAC as a function of δ for the two-qubit transmon control problem for both cases is shown in Fig. 6.6(a)–(e) for $\delta =\in \{0.01, 0.02, 0.05, 0.1, 0.2\}$.

For the two-qubit transmon problem, the $\delta = 0.02, 0.05, 0.1$ results show worse performance compared to the $\delta = 0.2$ results for the theoretical unitary control problem (without measurement noise). This indicates that some model system Hamiltonians $H_0(\boldsymbol{\zeta})$ with a larger δ predict dynamics more consistent with the true system Hamiltonian H_0 dynamics than $H_0(\boldsymbol{\zeta})$ with a smaller δ. However, learning $H_{0_{\text{tra}}}^{(2)}$ for all shown cases restores performance for both the noiseless unitary and shots-based closed system control problems.

To explain these empirical results and make them more intuitive, we now make use of the integration by parts lemma of Ref. [Bur+22] that bounds δ by the unitary prediction error of the ODE model w.r.t. the environment for the unitary control problem Eq. (2.10).

To make our empirical results more intuitive, we bound the unitary prediction error of the ODE model w.r.t. the environment for our idealised control problem Eq. (2.10) with the error in the model parameters from truth. Here, we focus, proof-wise, on the unitary case for simplicity since the arguments are similar for open systems. We also consider a continuous version of the propagators and the generators since the result makes a qualitative point only and again the discretization generalizations can be approximately made if necessary.

Consider a unitary RL control problem with the MDP in Eq. (6.1), where the environment's Hamiltonian and propagator at some timestep t_l are given by $H_{\mathcal{E}}(t_l, u_l) = H_0 + H_c(u_l, t_l)$ and $U_{\mathcal{E}}(\mathbf{u}_k)$. Now consider the model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k+1} \,|\, \mathbf{a}_k, \mathbf{s}_k)$ that predicts a single step of unitary dynamics $\mathbf{s}_k \xrightarrow{H_{\boldsymbol{\zeta}}} \mathbf{s}_{k+1}$ under its parameterised generator $H_{\boldsymbol{\zeta}} = H_0^{(L)}(\boldsymbol{\zeta}) + H_c(u_l, t_l)$ following our assumptions in Sec. 6.1. Now we bound the error in the single step predicted propagator $U_{\boldsymbol{\zeta}}$ using the integration-by-parts lemma from Ref. [Bur+22].

Proposition 6.6. *(Bound on the model predictions) The following bound between the unitary model's predicted state $U_{\boldsymbol{\zeta}}(\mathbf{u}_{:k})$ and the environment's unitary state $U_{\mathcal{E}}(\mathbf{u}_k)$ holds*

$$\left\| U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}} \right\|_{\infty,t} \;\leqslant\; t^2 \left\| H_0^{(L)}(\boldsymbol{\zeta}) - H_0 \right\| \cdot \left(\frac{1}{t} + \frac{2}{t} \| H_c \|_{1,t} + \| H_{\boldsymbol{\zeta}} \| + \| H_{\mathcal{E}} \| \right) \tag{6.21}$$

Proof: Note that the generator difference $H_{\boldsymbol{\zeta}} - H_{\mathcal{E}} = H_0^{(L)}(\boldsymbol{\zeta}) - H_0$ is time-independent. So the integral action difference term becomes

$$\begin{aligned} \left\| \int_0^t ds \; H_0^{(L)}(\boldsymbol{\zeta}) - H_0 \right\|_{\infty,t} &= t \left\| H_0^{(L)}(\boldsymbol{\zeta}) - H_0 \right\|_{\infty,t} \\ &= t \| H_0^{(L)}(\boldsymbol{\zeta}) - H_0 \|, \end{aligned} \tag{6.22}$$

where, in the last line, we drop the supremum over time due to time independence. Now we can rewrite

$$\| H_{\mathcal{E}}(\mathbf{u}(t), t) \|_{1,t} = t \| H_0 + H_c(\mathbf{u}(t), t) \|_{1,t} \tag{6.23}$$

$$\leqslant t \left(\| H_0 \| + \| H_c(\mathbf{u}(t), t) \|_{1,t} \right) \tag{6.24}$$

using the triangle inequality. Combining both facts yields the inequality. □

The inequality in Eq. (6.21) can be analogously extended to the open system setting w.r.t. the Choi matrix $\boldsymbol{\Phi}$. Note that $\left\| U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}} \right\|_{\infty,t} \leqslant 2$ and so the unitary prediction error in the left hand side in Eq. (6.21) is generally not linearly related to the Hamiltonian error. Finally note that Prop. 6.6 will be qualitatively used and tighter bounds are possible [Bur+22] but will not be qualitatively different.

There are two observations worth mentioning about inequality Eq. (6.21): (a) when all other variables are fixed, the error in the model's unitary predictions w.r.t. the environment's ground truth grows as a function of time; (b) the model prediction error

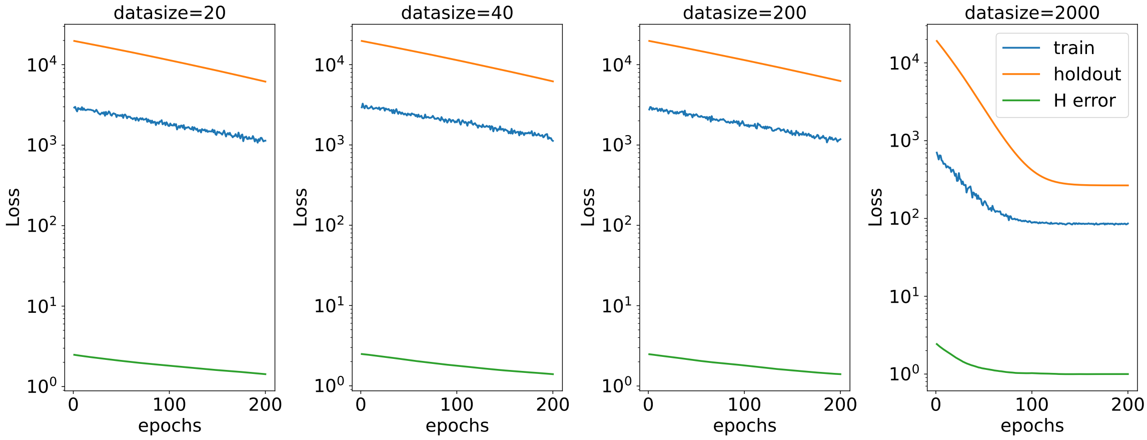


Figure 6.7: The Hamiltonian error, unitary training $L_{\text{model}}(\mathcal{D}_{\text{train}})$ and validation (holdout) loss $L_{\text{model}}(\mathcal{D}_{\text{val}})$ as functions of training epochs for the two-qubit transmon unitary control problem with noisy measurements and $M = 10^5$. Data size denotes the number of single-step unitary transitions. The validation set is fixed to $5,000$ transitions under random policy actions $\mathbf{a}_k$. All three error measures improve as a function of training. Adding more training data appears to provide diminishing returns in predicting the local unitary dynamics.

is a lower bound of the error in the model parameters $H_0(\boldsymbol{\zeta})^{(L)}$ w.r.t. the ground truth parameters H_0. The prediction error $L_{\text{model}}(\mathcal{D}_{\text{val}})$ can be estimated using a validation dataset $\mathcal{D}_{\text{val}}$ and relates this observed validation loss to the Hamiltonian difference. Importantly, we note that the inequality implies that the closeness in the propagator does not always translate to closeness in the Hamiltonian. Therefore, a model Hamiltonian can be locally a good fit for propagator predictions while still having a large Hamiltonian error $\left\|H_0^{(L)}(\boldsymbol{\zeta}) - H_0\right\|$. So arbitrary closeness in terms of the Hamiltonian error need not be necessary for good unitary predictions. But conversely, if we can be certain that the model Hamiltonian is close to the system Hamiltonian, then the unitaries must be close. This motivates that a good guess (in the form of partial knowledge about the system) of the true Hamiltonian is useful in bounding the prediction errors.

We exploit this fact to learn the local Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ that approximates the dynamics of H_0 w.r.t. $U_{\mathcal{E}}$. Qualitatively, we observe that Hamiltonian error, propagator validation and training error are both improved during training (i.e., the propagator loss on the validation set is predictive of Hamiltonian error). This can be seen in Fig. 6.7 for the noisy shot setting. But we also note in this example that the Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ that is learned is local, as seen from the Hamiltonian error plateauing at a non-zero value.

From this, we infer that the unitary model prediction error or the supervised learning regression loss $L_{\text{model}}(D)$ in Eq. (6.13) being small does not imply closeness between learned and true system Hamiltonian, i.e., $\delta \to 0$. This is illustrated for the two-qubit transmon control problem in Fig. 6.8(a). Note that there is also a lot of variation in the unitary model prediction error, even for the same value of δ. However, we see that with decreasing δ, the variation decreases, which is also explained by the above bound. Moreover, we confirm that the unitary model prediction error grows as a function of time. This makes intuitive sense since predictions far into the future compared to their time-wise preceding counterparts must necessarily have more built-up error.

Although there are some works with better relational bounds on the Hamiltonian error in terms of the observable error, these hinge on the ability to maintain a privileged basis and/or access to special probe states such as the Gibbs state basis [Ans+21; HKT22]. These bounds crucially do not include the propagator error, thanks to previous assumptions, which is a more general approach to bounding the quantum dynamical evolution error. Of course, there is always a price to be paid for generality and in this case, it is that the error bounds are less constrained and the link between the Hamiltonian and the unitary error becomes non-linear for the general case of the bound.

From Prop. 6.6, we infer that the unitary model prediction error or the supervised learning regression loss $L_{\text{model}}(D_{\text{train}})$ in Eq. (6.13) being small does not imply closeness between learned and true system Hamiltonian, i.e., $\delta \to 0$. However, in the converse case, δ being very small necessarily implies small propagator error. This is illustrated for the two-qubit transmon Hamiltonian in Fig. 6.8(a). The Hamiltonians are again sampled using Gaussian perturbations to the transmon Hamiltonian. There is also significant variation in the unitary model prediction error, even for the same value of δ for different repetitions of the random Hamiltonian. However, we see that with decreasing δ, the variation decreases, which is also explained by the above bound. Finally, the same pattern can also be observed if we take δ to be the mean squared difference between the Pauli coefficients of the true and learned Hamiltonian. Thus, this behaviour is general and not limited to the choice of δ.

The main takeaway of this section, that is taken further in the next section, is that for the control problems considered in this thesis it is only necessary to learn models that are 'locally consistent' in terms of the unitary trajectories they generate, and small unitary prediction errors can be achieved by models with non-negligibly small δ.

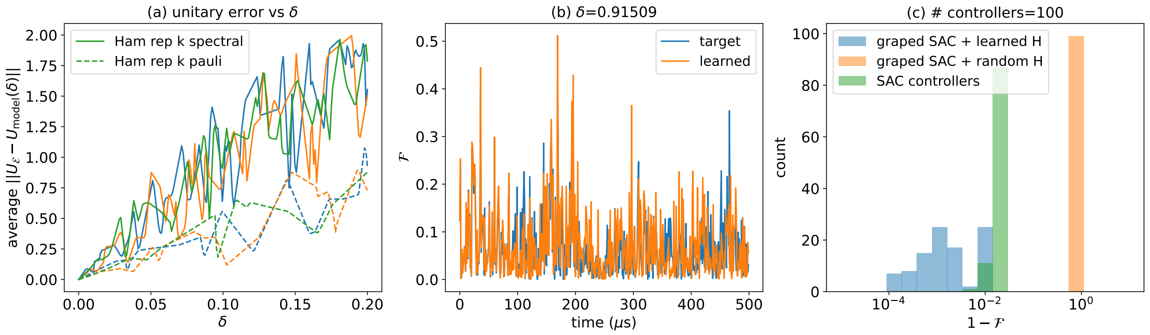


Figure 6.8: (a) An illustration of the non-linear relationship between the unitary model prediction error $\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|$ and Hamiltonian spectral norm (solid) error or mean squared Pauli basis difference (dashed) error as δ for the two-qubit transmon control problem. For the same $1,000$ random control pulses, we evaluate the average unitary prediction error of $\mathbf{M}_{\boldsymbol{\zeta}}$ with increasing δ for three different uniform randomly sampled two-qubit Hamiltonians $H_0(\boldsymbol{\zeta})$ to illustrate the variation in response to the unitary error. (b) Local and global unitary trajectories: $\mathcal{F}$ as a function of a random control pulse with either the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ or the true system Hamiltonian H_0. The learned $H_0(\boldsymbol{\zeta})$ trajectories do not coincide with the global trajectory with $\delta = 0.91509$, with the majority contribution coming from a global phase factor such that $\text{Tr}[H - H_0(\boldsymbol{\zeta})] \approx 0.9$. Both trajectories start off extremely close and start diverging as time increases due to accumulation of small errors in the predicted dynamics. (c) The learned $H_0(\boldsymbol{\zeta})$ can be leveraged using GRAPE to further optimize the fidelities of LH-MBSAC's controllers. We plot a histogram of 100 LH-MBSAC controller infidelities $1 - \mathcal{F}$ before and after applying GRAPE on these controllers using the learned Hamiltonian and a random Hamiltonian. The LH-MBSAC fidelities are significantly improved after applying GRAPE. The appropriate baseline/benchmark representing our ignorance of H_0 is a random $H_0(\boldsymbol{\zeta})$ (with uniform random Pauli parameters) which, when plugged into GRAPE, yields extremely low fidelities near 0 towards the extreme right-hand side of the plot.

6.3.3 Leveraging the Learned Hamiltonian with GRAPE

Proposition 6.6 paves the way to learning system Hamiltonians that are locally consistent with the unitary trajectories they generate. By local we mean that the learned Hamiltonian is consistent with the true Hamiltonian on only a subset of all possible generatable trajectories relevant to the control problem. In this section, we delve deeper into the learned model errors and also show that these local models can be leveraged to further optimize the fidelities of LH-MBSAC's controllers using gradient-based methods like GRAPE [Kha+05a; Mac+11a].

During the model's $\mathbf{M}_{\boldsymbol{\zeta}}$ training phase, $H_0(\boldsymbol{\zeta})$ is made consistent with trajectories uniform randomly drawn from the data buffer $D_{\mathcal{E}}$ by minimizing the regression loss $L_{\text{model}}(D_{\mathcal{E}})$. This allows us to learn a model of the environment that can predict locally consistent unitary trajectories (i.e., at the scale of the control problem). In other words, the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ does not have to coincide with the true system Hamiltonian H_0 for it to be useful for the optimal control task. Indeed, we take the Hamiltonian learned for the two-qubit transmon in Fig. 6.3(c) and find that it has $\delta = 0.91509$. Diving deeper, the matrix difference between the true H_0 and learned Hamiltonian $H_0(\boldsymbol{\zeta})$ is,

$$H \; - \; H_0(\boldsymbol{\zeta}) \quad = \quad \begin{bmatrix} -0.912 & 0.001 & -0.001 & 0.001 \\ 0.001 & -0.914 & 0.001 - 0.001i & 0.001 + 0.001i \\ -0.001 & 0.001 + 0.001i & -0.913 & -0.001 \\ 0.001 & -0.001 - 0.001i & -0.001 & -0.914 \end{bmatrix}.$$

Notably, we can see that most of the error is actually in $\mathrm{Tr}[H - H_0(\boldsymbol{\zeta})]$ with the true Hamiltonian being learned up to a scale factor of around 0.9 with the rest of the parameter error being small. This is precisely the global phase error that cannot be learned [EHF19].

Despite this discrepancy between the true and learned system Hamiltonians, we find mostly good local agreement between the two random trajectories they induce thanks to the supervised training phase of the model. We show in Fig. 6.8(b) the local and global trajectories corresponding to $H_0(\boldsymbol{\zeta})$ and H_0 for the two-qubit transmon which shows that the two unitary trajectories w.r.t. the CNOT fidelity are not always coinciding. More specifically, we can see a high overlap in the fidelities induced by random pulses for times between 0 and 100 μs. Moreover, the small differences in the generator only start manifesting as the time scales get longer and this can be explained by accruing of small errors in predicted dynamics. This confirms that the

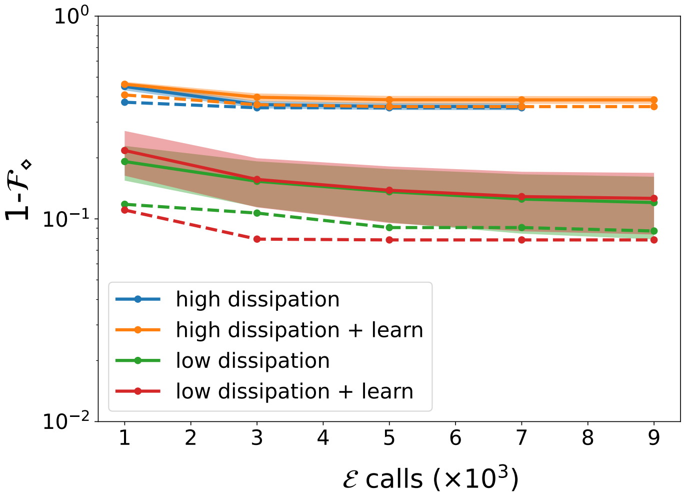


Figure 6.9: Diamond norm fidelity $\mathcal{F}_\diamond$ for the two-qubit transmon control problem in low and high Lindblad dissipation regimes for LH-MBSAC. The results are averaged over two seeds with the mean $\mathcal{F}_\diamond$ over 100 controllers shown in solid and the maximum $\mathcal{F}_\diamond$ in dashed lines. Shading denotes two standard deviations from the mean. Here, the 'learn' label signifies that dissipation operators are being learned in addition to the system Hamiltonian.

unitary model prediction error grows as a function of time. This makes intuitive sense since predictions far into the future, compared to their time-wise preceding counterparts, must necessarily have more built-up error. Furthermore, this learned 'local' $H_0(\boldsymbol{\zeta})$ and the controllers found by LH-MBSAC can be used in conjunction with the model-based GRAPE control algorithm [Kha+05a; Mac+11a] to optimize the SAC controller fidelities much more quickly than via just RL alone using accelerated second-order gradient descent. The LH-MBSAC controllers act as seeds, so GRAPE does not move too far away in pulse parameter space compared to where it started. Although not done here, this can also be imposed as an explicit constraint. Note that the question of exactly when to switch over to GRAPE beyond heuristics remains unanswered.

The fidelities after applying GRAPE are evaluated w.r.t. the true system Hamiltonian H_0. Usually LH-MBSAC controllers have moderately high fidelities around $\mathcal{F} > 0.98$ which are improved to $\mathcal{F} > 0.999$. In Fig. 6.8(c), we show the RL controllers being optimized further using the learned $H_0(\boldsymbol{\zeta})$ with GRAPE. Experiments in this section for the two-qubit NV center system yield similar results and can be found in Appendix B.4.

6.3.4 Open System Control with Single Shot Measurements

Due to the interpretable nature of our ODE model's ansatz in Eq. (6.12), it is pertinent to ask if two competing but linear terms in the model $\mathbf{M}_{\boldsymbol{\zeta}}$ can be learned simultaneously. In this section, we find that for our model learning setting, the answer to this question is no. However, this is not general to all problem settings and could potentially be pursued in future work.

In the previous sections, we only learn one term represented by $H_0(\boldsymbol{\zeta})$. Utilizing the open system formulation of the control problem in Eq. (2.19), we consider Lindblad dissipation along with shot noise for the two-qubit transmon control problem in Eq. (2.19). Specifically, we consider the decoherence operator $\mathfrak{L}^{(l)}_{\text{diss}} = \sqrt{\frac{2}{R_l^*}} b_l b_l^\dagger$, acting on the lth qubit, and the decay operator $\mathfrak{L}^{(l)}_{\text{deca}} = \sqrt{\frac{2}{R_l}} b_l$ for $l = 1, 2$. R_l^* and R_l are the decoherence and decay rates. Both operators are time-independent. Comprising both of these time-independent operators, the Lindblad term $\mathbf{L}_1$ is learned concomitantly with the system Hamiltonian.

We perform experiments for high and low dissipation corresponding to the times $R_l^{*^{\text{hi}}} = R_l^{\text{hi}} = 4$ μs, and $R_l^{*^{\text{lo}}} = R_l^{\text{lo}} = 20$ μs. The results are shown in Fig. 6.9 where the 'learn' label signifies that $\mathbf{L}_1$ is being learned in addition to the system Hamiltonian $H_0(\boldsymbol{\zeta})$.

The diamond norm fidelity [BS10] $\mathcal{F}_\diamond$,

$$\mathcal{F}_\diamond(\boldsymbol{\Phi}(\mathbf{u}(t), t), \boldsymbol{\Phi}_{\text{target}}) = 1 - \|\boldsymbol{\Phi}(\mathbf{u}(t), t) - \boldsymbol{\Phi}_{\text{target}}\|_\diamond, \tag{6.25}$$

is used instead of the generalised state fidelity since the latter lacks the sensitivity to detect the low dissipation regime (see Appendix B.3). We find that attempting to learn $\mathbf{L}_1$ while learning $H_0(\boldsymbol{\zeta})$ confers little to no advantage in both the high and low dissipation regimes for this control task. Further investigation shows that the estimate of the system Hamiltonian $H_0(\boldsymbol{\zeta})$ compensates for the observed discrepancy in observed dynamics due to dissipation as much as it is unitarily possible. Moreover, the learning processes for $\mathbf{L}_1$ and $H_0(\boldsymbol{\zeta})$ become mixed so learning multiple independent terms in $\mathbf{M}_{\boldsymbol{\zeta}}$ might not be suitable for LH-MBSAC.

6.3.5 Limitations and Silver Linings

We note that there are two major limitations of LH-MBSAC. The first is that only the system or time-independent part of the Hamiltonian can be learned using the

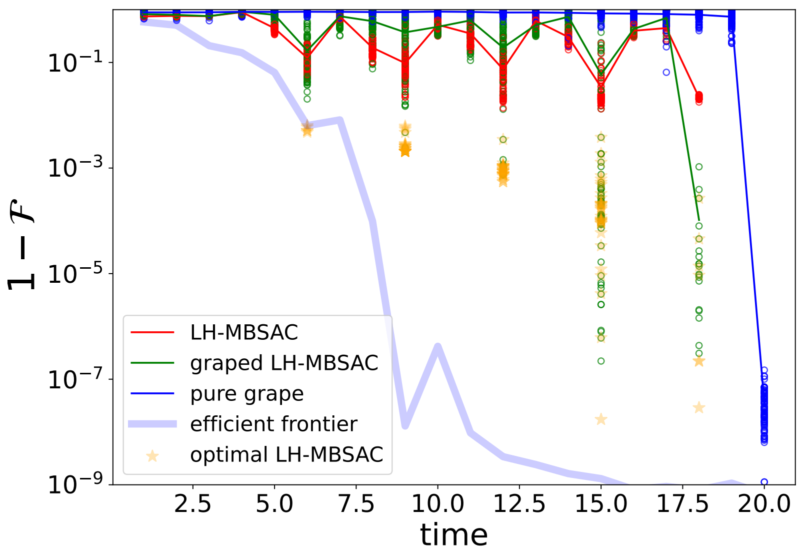


Figure 6.10: The infidelities over time for 100 different control pulses found by LH-MBSAC and by GRAPE using the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ for the two-qubit transmon control problem with final time $T \leqslant 20$ μs. RL pulses are further optimized using GRAPE. GRAPE is also used to obtain pulses without the RL controls as initial values for a fixed final gate time $T = 20$ μs. Short optimal controls found by RL are identified by truncating RL pulse parameters at times $t \geqslant \{6, 9\}$ μs whose final infidelities are shown as stars with $t = 6$ μs being Pareto optimal w.r.t. the efficient frontier (the surface indicating the best fidelity for that time).

algorithm, while the more difficult problem of learning the time-dependent part of the Hamiltonian [EHF19] is left as future work.

Moreover, we found that LH-MBSAC was not able to tackle a three-qubit transmon control problem to obtain a Toffoli gate on an extension of the transmon system. The limitation applied mostly to the RL agent; a viable Hamiltonian is learned that can be leveraged with GRAPE as before. Specific computational details are discussed in Appendix B.5. Essentially, our findings indicate this is an optimization landscape problem and an issue specific to the meta RL strategy of finding optimal pulses instead of a hyperparameter problem. There are two major reasons behind this assessment. Firstly, the values and the gradients for policy and value functions saturate with large training times, i.e., both are stuck in suboptimal extrema, which ultimately culminate with a prematurely optimized reward function. Secondly, since the model Hamiltonian is known beforehand (or also learned), GRAPE equipped with this Hamiltonian and initialized with the highest fidelity LH-MBSAC controllers also gets stuck.

However, the LH-MBSAC strategy is not limited to SAC and can augment different RL algorithms for which the three-qubit problem may be tractable. Also, since this is likely an optimization landscape issue, a reformulation of the RL control problem could also alleviate this issue by reducing the probability of SAC getting stuck by increasing the range of fidelities the RL agent sees as 'proximally optimal'. At present, the agent's goal is to maximize all fidelities it observes, with most of the observations being premature, i.e., before the final gate time. This is highlighted in Fig. 6.10 which shows the infidelity $1 - \mathcal{F}$ as a function of time for 100 pulses found by LH-MBSAC and GRAPE for the two-qubit transmon control problem. Compared to GRAPE, LH-MBSAC pulses are much more consistent and periodic in terms of the intermediate fidelity values. This highlights that the RL approach is biased towards optimizing intermediate fidelities along with the final target fidelity (since the objective function in Eq. (2.63) is the regularized expected cumulative fidelity). This is quite different from the approach taken by the gradient-based GRAPE algorithm. Despite being interesting from a controller robustness point of view [Kha+23b], this bias can prevent solutions that do not admit high intermediate fidelities from being found as RL can get stuck in a loop mining medium-level fidelity values. Stepping away from this particular sequential decision-making MDP formulation might be one solution to consider in future work.

There are silver linings for the aforementioned MDP formulation. RL pulses are fidelity-wise better, on average, across the duration of the pulse. Leveraging the learned system Hamiltonian, we can further improve the performance of the RL pulses by using GRAPE with the RL pulse parameters as initialization. As seen in Fig. 6.10, these pulses are still better than the ones found by GRAPE using the learned system Hamiltonian but with completely random pulse initializations, i.e., without LH-MBSAC controllers as seeds.

Furthermore, this RL bias towards valuing intermediate fidelities allows us to identify optimal pulses that can be executed in short times, which is a difficult problem for GRAPE even if the final gate time is explicitly added to the control objective [Mac+11a].

Truncating the control sequence for pulses at time t if the infidelity is below 5×10^{-2}, we again leverage GRAPE to maximize the final fidelities at these shorter times. These are shown as stars in Fig. 6.10 with the fidelities at $t = 6\ \mu\text{s}$ being approximately Pareto optimal, i.e., the best fidelity for that time. The Pareto optimal efficient frontier is constructed by sampling 100 GRAPE pulses with random intializations at different final gate times.

Finally, in the next section we demonstrate that controllers found by LH-MBSAC before and after applying GRAPE are more robust in comparison to GRAPE controllers found from scratch using the ARIM from Chapter 5.

6.3.6 Robustness of LH-MBSAC Controllers

In this section, we conduct a RIM analysis of LH-MBSAC controllers both before and after applying GRAPE to them. We compare both to the baseline where GRAPE is used without RL seeds to find controllers. The control problem is again for the two-qubit CNOT for the transmon system. In addition to the 100 controllers found earlier by LH-MBSAC, we compute 100 new GRAPE controllers with random initialization. We use the ARIM to compare the 3 control acquisiton algorithms in Fig. 6.11 using the techniques discussed in Chapter. 5. Also for the gate control problem like in previous chapters, it can be seen that RL controllers before and after applying GRAPE are more robust to noise in the control amplitudes and the system Hamiltonian compared to the pure GRAPE baseline. More importantly, this highlights that the advantages of robustness conferred by RL, which optimizes indirectly for the RIM, are not lost when GRAPE is applied to the RL controllers. Note that the computation of the RIM is expensive in the number of samples, using the RIM as a post evaluator of robustness rather than during optimization is more beneficial if the goal is sample efficiency (as discussed in Sec. 5.3.3). In contrast, since GRAPE is a second-order gradient-based algorithm (relying on L-BFGS), we find that it optimizes for the fidelity only and therefore expectedly performs poorly in terms of the ARIM compared to LH-MBSAC.

However, the application of GRAPE to RL controllers does well in contrast and this is likely due to the fact that the starting position (the SAC controller before applying GRAPE) of the second-order ascent to increase the fidelity coincides with being closer to a better local optimum than the one found by GRAPE on its own. The final controller amplitudes after finetuning using GRAPE are not too far from their starting position since GRAPE stops after the maximal peak sought by the RL algorithm, which it attains partially, is fully attained since here gradients become zero. Therefore, applying GRAPE to RL controllers allows us to avoid unnecessarily increasing the sample complexity of the RL algorithm to reach very high fidelities by stopping early whilst retaining the robustness properties of RL controllers.

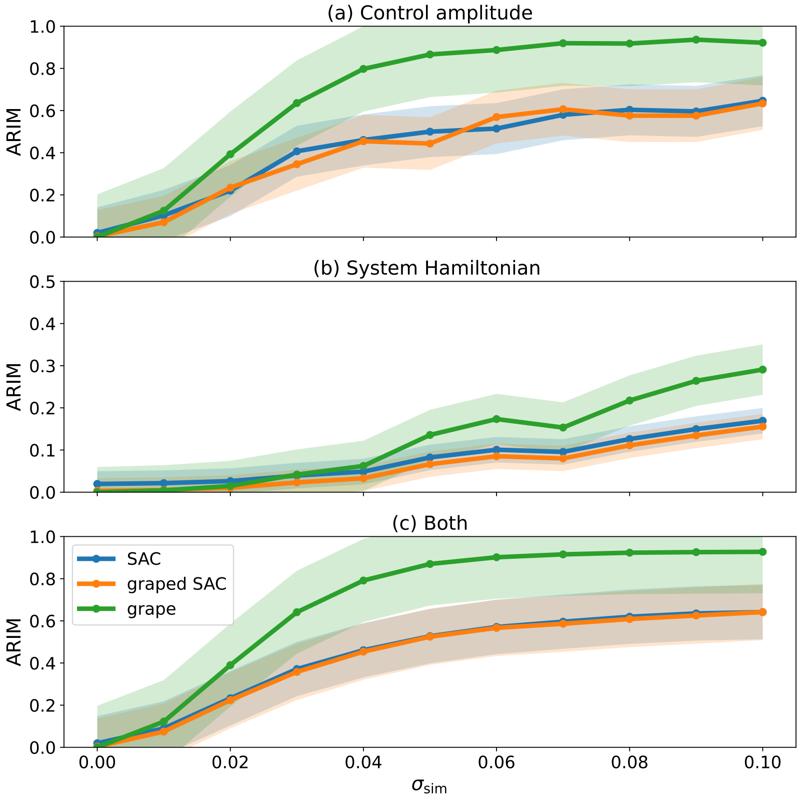


Figure 6.11: ARIM comparison of RL controllers (SAC) found by LH-MBSAC, the variant when GRAPE is applied to the SAC controllers and the controllers found by GRAPE without RL seeds i.e. with random initialization. The ARIM is computed using the techniques discussed in Chapter 5. We use 1000 bootstrap samples to compute the 95% confidence intervals. Noise in (a), (b) and (c) is drawn from $\mathcal{N}(0, \sigma_{\text{sim}}^2)$ where σ_{sim} is a percentage computed w.r.t. the maximum value being perturbed. For (a) only the controller amplitudes are perturbed. For (b), the noise is added to the non-zero parameters of the system Hamiltonian and (c) considers both noise sources simultaneously. In all cases, RL controllers are ARIM-wise better than GRAPE controllers.

6.4 Conclusions

We have presented the learnable Hamiltonian soft actor-critic (LH-MBSAC) algorithm for time-dependent noisy quantum gate control. LH-MBSAC augments model-free SAC by allowing the RL policy to query a learnable model of the environment or the controllable system. It thereby reduces the total number of queries (sample complexity) required to solve the RL task. The model is a differentiable ODE that is equipped with a partially characterized Hamiltonian, where only the parametrized time-independent system Hamiltonian is required to be learned. We show why this is a good inductive bias for the quantum control task as ODE trajectories do not intersect, thereby sensibly constraining the space of models to be learned. Using exploration data acquired from the policy during the RL loop, we train the model by reducing a model prediction error over the data. We show that LH-MBSAC is able to reduce the sample complexity for gate control of one- and two-qubit NV centers and transmon systems in unitary and single-shot measurements' settings.

We note that our approach is similar in spirit to Ref. [CKW22] where a novel Hamiltonian learning protocol via quantum process tomography is proposed for the purpose of model-predictive control. The complete Hamiltonian (including the control and system parts) is identified term by term via a Zero-Order Hold (ZOH) method where only one term is turned on at a time, e.g., by setting the control parameters to zero, and it is learned individually using optimization over the Stiefel manifold. As a side remark, a sample complexity advantage between learning the Hamiltonian with quantum control than without it has recently been shown [DOS23]. The learned Hamiltonian is then used to obtain a viable control sequence for a variety of state and gate preparation problems for closed (unitary) systems under the influence of initial state preparation errors. While it is possible for our Hamiltonian learning protocol to also learn the full Hamiltonian using the ZOH method, we have instead solely focussed on the problem of improving the sample complexity of RL in this chapter through the incorporation of a partially known physics-inspired model. Furthermore, our focus has also been directed on the interplay of concurrently learning the model and controlling the system in noisy closed and open system settings. Exploring more efficient Hamiltonian learning protocols for the sake of control is certainly an active area of research and should be looked at in conjunction with modern machine learning theory and learning dynamics' algorithms like the ones considered in this thesis such as Neural ODEs [Kid22].

Moreover, we highlight that despite the non-linear relationship between the error in the learned Hamiltonian and the model prediction error, LH-MBSAC's performance is robust to this variation. Furthermore, even if the learned Hamiltonian that minimizes the model prediction error is not the same as the true system Hamiltonian, the learned Hamiltonian can be leveraged using gradient-based methods that require full knowledge of the controllable system, like GRAPE, to further optimize the controllers found by LH-MBSAC. Applying LH-MBSAC in high and low Lindblad dissipation regimes with shot noise, we found that its performance in both was not improved if the Lindblad dissipation terms are also learned in addition to the system Hamiltonian. This is likely because the learning of the system Hamiltonian is affected by the need to *incorrectly* account for the extra dissipation effects that should only be represented by the learned Lindblad operators compared to the case where we solely learn the system Hamiltonian. Finally, we showed that applying GRAPE to RL controllers still retains the robustness performance of RL controllers discussed in Chapter 5 using the ARIM and this procedure could therefore be used to improve the relatively moderate fidelities found by RL without expending significantly more queries of the system.

Despite LH-MBSAC's limitations requiring it to know the time-dependent Hamiltonian and system scalability beyond two qubits (four with single shot measurements due to AAPT), the algorithm can be used to augment many existing model-free RL approaches for quantum control. This should afford more sample-efficient RL-based optimization of quantum dynamics for near-term noisy quantum processors on a variety of architectures as shown in this chapter. Specific tasks can include noisy small circuit optimization, state preparation [Siv+22b; Buk+18] or gate optimization using a partially known model of the underlying dynamics [Dal+20c]. Since having an accurate model can be extremely useful for validation of quantum operations and model bias can be crippling, model-based RL methods like LH-MBSAC can improve the model specifically tailored for some downstream task, e.g., quality assessment of topological codes [Val+19] or fine-tuning current implementations of a two-qubit cross resonance gate on some novel architecture [Din+23] using a pre-existing but partially correct model. Here, the goal for the RL agent would be to help learn effective and potentially scalable models of the target system whilst optimizing the target functional. Another interesting goal in this direction could just be to incorporate the number of measurements or queries of the system in the RL objective so that the learning is sample-efficient. A further avenue of future work is to combine LH-MBSAC with a more feasible measurement protocol than AAPT. AAPT is not a hard requirement for our approach and was used here for its theoretically simple estimation of a quantum process. Two angles of attack are either sparsity assumptions on the dynamics generator [Hua+22a] and the generated evolution [HKP20] or a partially observed MDP formulation of the control problem [HS17; Kha+21].

Chapter 7

Conclusion

7.1 Takeaways

Quantum technologies is an exciting emerging field with a lot of potential real-world utility most notably in sensing and simulation applications. However, the path from the lab to industrial application will need to be paved by strong and robust quantum control frameworks that work well especially in the presence of noise. This poses a barrier for NISQ devices to realize the full extent of the promised theoretical quantum advantage over current classical hardware on various problems of interest. Making quantum devices perform optimally in the presence of noise of various forms was the overarching goal of this thesis. Towards this main goal, we made contributions addressing the problems of *robustness certification* and *robust optimization* of quantum control schemes.

For robustness certification, in Chapter 5, inspired by randomized benchmarking and control landscape theory, we developed a novel probabilistic robustness measure called the Robustness Infidelity Measure (RIM) by treating the fidelity as a random variable whose stochasticity is determined by noise in the controllable system and does not need to be explicitly modelled or assumed. The RIM is the probabilistic distance of the fidelity's probability distribution to the ideal unperturbed point mass distribution located at the optimum value. Using theoretical arguments, we showed that the RIM is intuitive, easy to compute and in its simplest form is just the average infidelity which allowed us to provably generalize the fidelity in the robustness sense and encapsulate both the optimality and robustness into a single figure-of-merit. Notably, the main takeaway is that not all optimal control schemes are robust and robustness needs to be an extra objective that should be considered. It is possible to extend any fidelity measure into a RIM as the only requirements for straightforward extension is stochasticity in the fidelity that can be estimated using multiple fidelity samples and boundedness of the figure-of-merit. Moreover, our framework allows us to extend the expectation operator hierarchically to measure the RIMs of control algorithms (ARIM) or even higher categories like the family to which the control algorithms belong. We also showed that the RIM is connected to a classical robustness measure, the log-sensitivity, and the RIM directly captures the intuition of measuring the variation in a local region on the control landscape around the optimum.

The smaller that variation, the smaller the RIM and higher the robustness of the control scheme. We expect the RIM to be useful in understanding robustness in objectives that are empirically designed to encode robustnesss in quantum control settings without the strong theoretical motivations – that we have helped provide.

For robust optimization, we considered the problem of control when the theoretical model of the system is absent or is faulty i.e. when the control system is noisy or the model is uncertain. Our goal, here, was to study the limitations of pure model-based methods that rely on gradient descent on the model's controllable parameters and also of model-free methods that are not dependent on any model but still suffer from large convergence times and consume too many experimental resources to be effective in real-world settings. Towards this goal, we explored a suite of policy gradient RL algorithms designed for continuous parameter control problems and benchmarked their performances on a standard quantum control task in Chapter 4 contrasting their performance with model-based gradient-descent methods. We presented a novel formulation of the RL control problem as a partially observed MDP that scales favourably with system size since the RL agent does not need to observe an exponentially growing quantum state space – an approach that existing RL strategies for VQA or circuit design type problems could benefit from if scalability is desired. Moreover, we explored the stability and practicality of these RL algorithms and found their performances to be broadly similar. However, the on-policy PPO algorithm was, on average, better than other algorithms on both fronts. Thus, we used PPO subsequently in Chapter 5 as our benchmark model-free RL algorithm. We also found that as expected, increasing model uncertainty to a reasonable extent significantly deteriorated the performance of model-based methods while model-free methods like RL remained mostly unaffected. Furthermore, using the RIM, we studied robustness profiles of control algorithms and found that optimizing the average infidelity, i.e. the RIM leads to a greater proportion of the computed control schemes to be robust

– which is what RL does indirectly through its objective function. The RL control schemes are better in contrast with those produced by other benchmarked control algorithms. We justified this empirical finding in Chapter 4 using the average infidelity but it is conditional on existence of robust and optimal control solutions in the first place which we found was not always the case for all the control problems that we studied.

Taking it further, we improved upon a central problem of model-free methods, in particular RL, which is their inability to incorporate any partial knowledge of the controllable system, instead, relying on the construction of *ab initio* models of the control system, which wastes potentially precious experimental resources. In Chapter 6, we introduced a novel model-based actor-critic RL algorithm that is equipped with such an ability by initializing its learnable model of the system with partial knowledge and making it learn the rest using quantum data. We demonstrated how this significantly improves the number of experimental resources required by the control algorithm by over an order of magnitude and extends model-free RL in the model-based direction, thereby improving the deployability of RL algorithms in near term NISQ devices.

7.2 Future work

In light of the main contributions of this thesis, there is a lot of scope of extending the tools that were presented both on the certification and optimization fronts. Some of the more direct or concrete directions were discussed in earlier chapters. Here, we discuss future directions of a more broader and general nature that are also not necessarily tied to a specific topic.

Firstly, given that a central theme of this thesis was acquisition of robust optimal control schemes, there is still a question of how to generally formally define the concept of robustness. In this thesis, we presented a definition in terms of the local variation of the fidelity around the optimum in the control landscape and tried to capture that property statistically. However, this necessarily forced us to choose the size of the local region within which the robustness was defined, i.e. the noise scale or strength that is used in constructing the probability distributions representing fidelities of a specific control scheme. An interesting question to pursue next would be how to define robustness in a more general way that does not depend on the choice of the noise scale or essentially probing the trade-off between the noise-scale and the robustness to generate a best-case robustness measure for a maximum noise scale. Ideally, this should afford a more global view of robustness. Furthermore, using these ideas would help us understand the effect of noise scale on the ARIM performances of control acquisition algorithms where variation was observed across different noise-scale.

Secondly, there remains the question of finding theoretical robustness guarantees for the specific or general control problems that were studied in this thesis. What is the highest (lowest) robustness permitted for a given fidelity for a particular control problem? Understanding the fundamental limitations of a control problem from a more mathematical perspective would be extremely useful before deployment of large scale computational searches for robust control schemes. Indeed, some more insight into the nature of the control problem and what specifically needs to be optimized could be used as a stepping stone to develop more sample efficient robust control acquisition algorithms. For instance, in chapter 5, robustness is defined in terms of the stability of state transfer w.r.t. system Hamiltonian fluctuations. Number-theoretic constraints on perfect [Bur07] and almost perfect state transfer [VZ12] in the presence of imperfections could be a useful starting point to understand the fundamental limitations of robustness of the control problem for spin chains and beyond.

Characterisation of fundamental limitations is also useful for providing incentive to reformulate the control problem since solutions of desired robustness are not possible in its existing form. These bounds on performance would also accelerate the performance of control acquisition algorithms by providing useful criteria for terminating the search for control schemes. It might generally not be possible to find such bounds for all control problems theoretically but computational tools motivated along this direction should still be useful for practical purposes and augmentation of control scheme searches.

Thirdly, motivated by the idea of model bias or absence or uncertainty for the control system, RL techniques were used to find control schemes and the sample efficiency of these techniques can be improved through the idea of learning effective models of the controllable system. It is possible to only learn effective models of the system as far as controlling it is concerned instead of learning the full model. This is very useful for scalability of such a technique. But there still remains the question of how to build these effective models from scratch using data with or without prior knowledge with a small and realistic probe resource overhead. If learning the exact system might not be necessary for the control problem at hand, is it possible to quickly identify and isolate the regions that are? And is it possible to get some resource guarantees on such a learning protocol? The idea of learning for control imposes a useful constraint and reduces the need for learning the whole system to just the relevant components. This should be explored further. More specifically, it might be possible to learn low level projections of the dynamics of controllable systems by learning to predict the dynamics using quantum data of the relevant observables. Identifying regions of interest in the system that are relevant for the control problem should permit sparsification of the dynamical observables that are necessary to predict.

On the whole, in this thesis, we have tried to address the present limitations in quantum optimal control frameworks when faced with modelling noisy quantum dynamics by developing novel certification and optimization techniques. However, the story is far from being over and there remains a need for more generic and powerful learning algorithms that can provide the robust performance of model-free and the efficiency of model-based methods on challenging noisy control problems. Further advancements in the field of classical machine learning and optimization as well the development or consideration of stronger Hamiltonian learning protocols in conjunction with learning algorithms should be able to push the frontier beyond the results proposed by this thesis.

Appendix A

RIM calculations and extended numerical analyses

A.1 p-th Order RIM decompositions

We can further decompose the RIM_p as a sum of expectations of various powers of the fidelity,

$$\begin{aligned}\mathrm{RIM}_p &= \left(\sum_{k=0}^{p}\binom{p}{k}(-1)^k\int_{-\infty}^{\infty}\mathbf{P}(\mathcal{F}=f)f^k\,df\right)^{\frac{1}{p}}\\ &\quad\text{using the binomial theorem}\\ &= \left(\sum_{k=0}^{p}\binom{p}{k}(-1)^k\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[f^k\right]\right)^{\frac{1}{p}}. \qquad \text{(A.1)}\end{aligned}$$

For example, using Eq. (A.1) for $p=2$, we obtain

$$\begin{aligned}\mathrm{RIM}_2 =&\sqrt{1-2\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[f\right]+\mathrm{Var}(f)+\mathbb{E}^2{}_{\mathbf{P}(\mathcal{F}=f)}\left[f\right]}\\ &\quad\text{using A.1, and}\mathrm{Var}(X)=\mathbb{E}_{X\sim\mathbf{P}}\left[X^2\right]-\mathbb{E}^2_{X\sim\mathbf{P}}\left[X\right]\\ =&\sqrt{\mathrm{RIM}_1+\mathrm{Var}(f)-\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[f\right]\mathrm{RIM}_1}\\ =&\sqrt{\mathrm{Var}(f)+\mathrm{RIM}_1^2} \qquad \text{(A.2)}\\ &\quad\text{expanding }\mathrm{RIM}_1\text{ and simplifying.}\end{aligned}$$

Likewise, we get

$$\mathrm{RIM}_3=\left(\mathrm{RIM}_1^3+3\mathrm{Var}(f)+\mathbb{E}^3{}_{\mathbf{P}(\mathcal{F}=f)}\left[f\right]-\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[f^3\right]\right)^{\frac{1}{3}}. \qquad \text{(A.3)}$$

The degree of distinguishability of the fidelity distribution from the ideal becomes better for higher p at the cost of the outliers becoming more influential.

A.2 Recovering the probability distribution from the RIM

As an aside, we will now show that it is possible to recover the probability distribution $\mathbf{P}(\mathcal{I})$ of the infidelity random variable $\mathcal{I}$ using the RIM_p values by taking the inverse Fourier transformation of the characteristic function $C_t(\mathcal{I}) = \mathbb{E}_{x\sim\mathbf{P}(\mathcal{I})}[\exp\{itx\}]$. Recall the definition of the RIM,

$$\begin{aligned}\mathrm{RIM}_p^p =& \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right] \\ =& \mathbb{E}_{1-f\sim\mathbf{P}(\mathcal{I})}\left[(1-f)^p\right].\end{aligned} \tag{A.4}$$

We write the characteristic function for as

$$\begin{aligned}C_t(\mathcal{I}) =& \mathbb{E}_{1-f\sim\mathbf{P}(\mathcal{I})}\left[\exp\{it(1-f)\}\right] \\ =& \mathbb{E}_{1-f\sim\mathbf{P}(\mathcal{I})}\left[\sum_{k=0}^{\infty}\frac{(it)^k(1-f)^k}{k!}\right] \\ =& \sum_{k=0}^{\infty}\frac{(it)^k\mathbb{E}_{1-f\sim\mathbf{P}(\mathcal{I})}\left[(1-f)^k\right]}{k!} \\ & \text{using linearity of expectation} \\ =& \sum_{k=0}^{\infty}\frac{(it)^k\mathbb{E}_{f\sim\mathbf{P}(\mathcal{F})}\left[(1-f)^k\right]}{k!} \\ & \text{using expectation equivalence} \\ =& \sum_{k=0}^{\infty}\frac{(it)^k\mathrm{RIM}_k^k}{k!}.\end{aligned} \tag{A.5}$$

By taking the inverse Fourier transform of $C_t(\mathcal{I})$, we recover

$$\begin{aligned}\mathbf{P}(\mathcal{I}=1-f) =& \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-it(1-f)}C_t(\mathcal{I})\,dt \\ =& \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\sum_{k=0}^{\infty}\frac{(it)^k\mathrm{RIM}_k^k}{k!}e^{-it(1-f)}\,dt.\end{aligned} \tag{A.6}$$

Finally, the distributional equivalence $\mathbf{P}(\mathcal{F}=f) = \mathbf{P}(\mathcal{I}=1-f)$ permits a change of variable between the two distributions to recover $\mathbf{P}(\mathcal{F})$. Alternatively, a more direct way to obtain $\mathbf{P}(\mathcal{F})$ is to use the raw moments of $\mathcal{F}$,

$$\mathbf{P}(\mathcal{F}=f) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\sum_{k=0}^{\infty}\frac{(it)^k\mathbb{E}_{f\sim\mathbf{P}(\mathcal{F})}\left[f^k\right]}{k!}e^{-itf}\,dt. \tag{A.7}$$

A.3 Error bound on the RIM$_p$ and ARIM$_p$ estimators

Here we propose a probably approximately correct (PAC) alternative error bound for an estimation $\widehat{\mathrm{RIM}}_p$ of RIM$_p$ in Eq. (5.10) based on an empirical estimate $\widehat{\mathbf{P}}(\mathcal{F})$ of its generating probability distribution $\mathbf{P}(\mathcal{F})$. With probability at least $1-\delta/2$,

$$\begin{aligned}
|\widehat{\mathrm{RIM}}_p - \mathrm{RIM}_p| &= \left|\mathbb{E}_{f\sim\widehat{\mathbf{P}}(\mathcal{F})}\left[(1-f)^p\right]^{\frac{1}{p}} - \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}}\right| && \text{(A.8)}\\
&= \left|\|\mathbb{E}_{f\sim\widehat{\mathbf{P}}(\mathcal{F})}\left[(1-f)^p\right]^{\frac{1}{p}}\| - \|\mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]^{\frac{1}{p}}\|\right| && \text{(A.9)}\\
&\leqslant |\mathbb{E}_{f\sim\widehat{\mathbf{P}}(\mathcal{F})}\left[(1-f)^p\right] - \mathbb{E}_{\mathbf{P}(\mathcal{F}=f)}\left[(1-f)^p\right]|^{\frac{1}{p}} && \text{(A.10)}\\
&= \left|\int_0^1 \widehat{\mathbf{P}}(\mathcal{F}=f)(1-f)^p\,df - \int_0^1 \mathbf{P}(\mathcal{F}=f)(1-f)^p\,df\right|^{\frac{1}{p}} && \text{(A.11)}\\
&\leqslant \left(\int_0^1 |\widehat{\mathbf{P}}(\mathcal{F}=f) - \mathbf{P}(\mathcal{F}=f)|(1-f)^p\,df\right)^{\frac{1}{p}} && \text{(A.12)}\\
&= \left(\int_0^1 \left|\widehat{\mathbf{P}}(\mathcal{F}=f) - \mathbb{E}_{\widehat{\mathbf{P}}\sim\mathbf{D}}\left[\widehat{\mathbf{P}}(\mathcal{F}=f)\right]\right|(1-f)^p\,df\right)^{\frac{1}{p}} && \text{(A.13)}\\
&\leqslant \frac{C^{\frac{1}{p}}}{p+1} = \frac{1}{p+1}\left(\frac{\log\frac{4}{\delta}}{2n}\right)^{\frac{1}{2p}} && \text{(A.14)}
\end{aligned}$$

where the third line come from using Jensen and in the sixth line we rewrite the true distribution $\mathbf{P}(\mathcal{F})$ as $\mathbb{E}_{\widehat{\mathbf{P}}\sim\mathbf{D}}\left[\widehat{\mathbf{P}}(\mathcal{F})\right]$ which is true for any unbiased empirical estimator. We use McDiarmid's inequality to obtain the bounding constant C using the fact that the probability distribution $\mathbf{D}$ generates a family of random variable empirical distributional estimators $\widehat{\mathbf{P}_j} = \frac{1}{n}\sum_{i=1}^{n}\delta_{f_i}$ where we have the differences occurring only on the k-th coordinate,

$$\left|\widehat{\mathbf{P}}(f_1,\ldots,f_k,\ldots,f_n) - \widehat{\mathbf{P}}(f_1,\ldots,f_{k'},\ldots,f_n)\right| \leqslant \frac{1}{n}, \qquad \text{(A.15)}$$

where n is the number of samples. A similar bound can also be derived for the $\widehat{\mathrm{ARIM}}$ estimator. This error bound is similar to the DKW (Dvoretzky-Kiefer-Wolfowitz) bound for the ECDF and would suffice in generating the 95% confidence intervals for Fig. 5.6 without the need to do bootstrap resampling.

Table A.1: Implementation details for various optimization settings in Chapter 5. For Sec. 5.3.3, the asymptotic setting, (i) refers to the stochastic scenario and (ii) refers to the non-stochastic scenario where the RIM is optimized using the same 100 fixed set of perturbations $\{S_{\sigma_{\text{train}}}\}$ per function call.

Sec.	Obj. Function (OF) and args.	Train (OF) noise	Algorithm	Total OF Calls	Single Call Cost
5.3.1.1	$\mathcal{F}$	No	L-BFGS	10^6	1
5.3.1.1	$\mathcal{F}$	No	PPO	10^6	1
5.3.1.1	$\mathcal{F}$	No	SNOBFit	10^6	1
5.3.1.1	$\mathcal{F}$	No	Nelder-Mead	10^6	1
5.3.2	$\mathcal{F}$	No	L-BFGS	10^6	1
5.3.2	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	PPO	10^6	1
5.3.2	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	SNOBFit	10^6	1
5.3.2	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	Nelder-Mead	10^6	1
5.3.3(i)	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	L-BFGS	∞	1
5.3.3(i)	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	PPO	∞	1
5.3.3(i)	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	SNOBFit	∞	1
5.3.3(i)	$\mathcal{F}$ & 1 $S_{\sigma_{\text{train}}}$	Yes	Nelder-Mead	∞	1
5.3.3(ii)	RIM & 100 fixed $S_{\sigma_{\text{train}}}$	No	L-BFGS	∞	100
5.3.3(ii)	RIM & 100 fixed $S_{\sigma_{\text{train}}}$	No	PPO	∞	100
5.3.3(ii)	RIM & 100 fixed $S_{\sigma_{\text{train}}}$	No	SNOBFit	∞	100
5.3.3(ii)	RIM & 100 fixed $S_{\sigma_{\text{train}}}$	No	Nelder-Mead	∞	100

A.4 Implementation details of the optimization objectives

Details about the optimization objectives for the numerical results in Sec. 5.3 are given in Table A.1. In every section in Chapter 5, for every σ_{sim}, the RIM is evaluated using $N = 100$ Monte Carlo $S_{\sigma_{\text{sim}}}$ perturbations to the fidelity function. The RIM itself is only optimized in Sec. 5.3.3 for the non-stochastic case (ii) where 100 $S_{\sigma_{\text{sim}}}$ are sampled at the start and are reused for every function call. Note, however, that we count these as 100 function calls as these amount to 100 fidelity function evaluations. Also, for better performance, in Sec. 5.3.3 for the stochastic case (i), instead of using the analytical form for the gradient of fidelity $\mathcal{F}$, we use finite differences to approximate the gradients $\nabla_{\Delta}\mathcal{F}$ (where Δ are the controls).

A.5 More individual controller plots

The results presented in Fig. 5.4 ($M = 5$ and transition $|1\rangle$ to $|3\rangle$) are not reflective of PPO's general behavior on the extended sample of problems examined in Sec. 5.3.2. Fig. A.2 shows the case ($M = 5$ and transition $|1\rangle$ to $|4\rangle$) where all the controllers found are not very robust. This is likely either due to unlucky sampling of the space of possible controllers or their non-existence. Note that SNOBFit and PPO are similar in their RIM degradation as observed from Fig. A.2(e). We also provide some more cases ($M = 5$ and transition $|1\rangle$ to $|5\rangle$; $M = 6$ and transitions $|1\rangle$ to $|4\rangle$, $|1\rangle$ to $|6\rangle$) for algorithm comparison of controllers under noisy training in Sec. 5.3.2 to highlight some of the variation of controller quality for different regimes of noise and spin chain transitions observed in the main ARIM comparison presented in Fig. A.8. Each individual subplot is the result of an independent run of each algorithm with a stochastic fidelity function evaluated under the unstructured perturbations using with σ_{sim}. These are also plotted for a more distributional comparison as pairwise box-plots in Fig. A.7. For both, Figs. A.3 and A.7, we also show L-BFGS results for comparison.

A.6 Full ARIM comparisons

For the cases $M = 6, 7$, both types of transitions appear to be challenging for PPO, SNOBFit and Nelder-Mead at most, if not all, training perturbation strengths; especially the end-to-end $M = 7$ transition (Fig. A.8(d)), where PPO at $\sigma_{\text{train}} = 0.05$ is only marginally better than the rest of the algorithm runs, excluding Nelder-Mead. A pertinent question is whether this is genuinely reflective of the landscape or if, for PPO, our budget constraint of 10^6 target functional calls is insufficient for larger system sizes, as the control problem is exponentially dependent on the number of control degrees of freedom. The former hypothesis might hint at a fundamental limitation on robustness of this particular control landscape. The fact that most noisy Nelder-Mead curves for these problems are clustering together suggests that noise could also help in reaching robust areas in the control landscape faster by regularizing or smoothing the landscape by an appropriate degree. We investigate asymptotic algorithm behavior w.r.t. the training noise in Sec. 5.3.3 to illustrate this and show that there is convergence in PPO performance for all the noise levels at sufficiently many function calls.

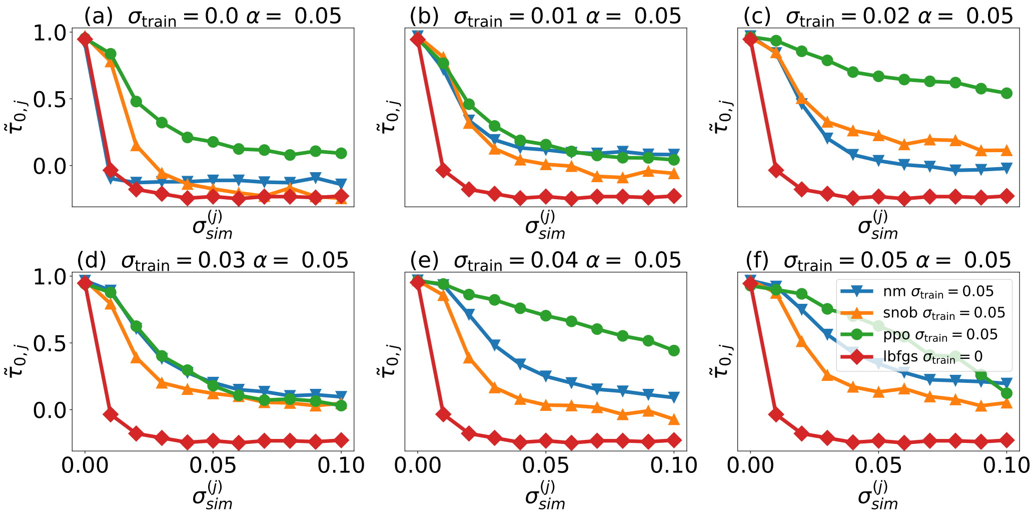


Figure A.1: Consistency statistic $\tilde{\tau}_{0,j}$ for all algorithms at $\sigma_{\text{train}} = 0.0, \ldots, 0.05$ for discrepancy parameter $\alpha = 0.05$ for $M = 5$ and the transition from $|1\rangle$ to $|4\rangle$. Again, the PPO curves are the most consistent.

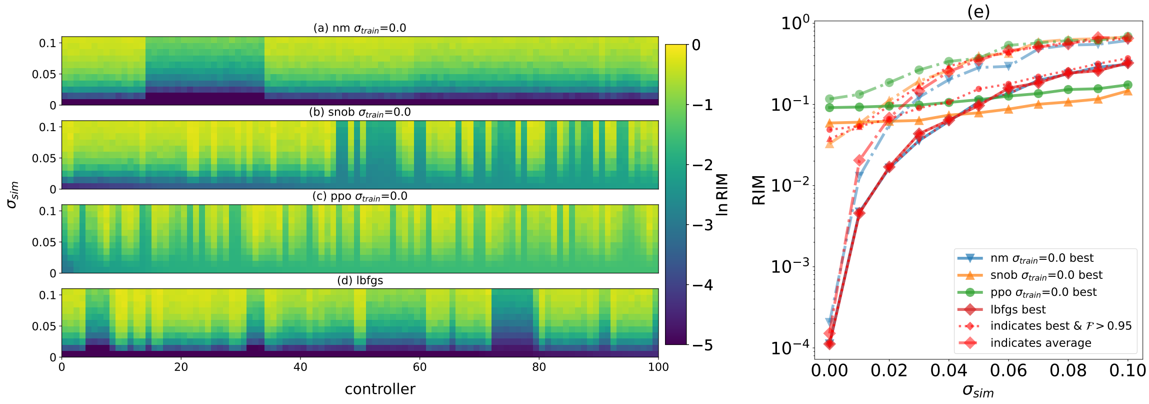


Figure A.2: (a)-(d) 100 controllers found for the XX spin chain model, Eq. (4.2), using Nelder-Mead, SNOBFit, PPO ($\sigma_{\text{train}} = 0$), and L-BFGS for $M = 5$ and the spin transition from $|1\rangle$ to $|5\rangle$. All algorithms find controllers that are not very robust as indicated by the RIM. PPO has notably worse initial infidelities for all controller compared to Fig. 5.4(c), but their degradation is slow as seen from (e). This is only the case for this noise level and Fig. A.4(r) indicates the existence of a much better controller set at $\sigma_{\text{sim}} = 0.05$ that is similar in performance to Fig. 5.4(c). From (e), we can see that Nelder-Mead and L-BFGS optimize the infidelity to $< 10^{-4}$. However, these best controllers decay in robustness very quickly as well.

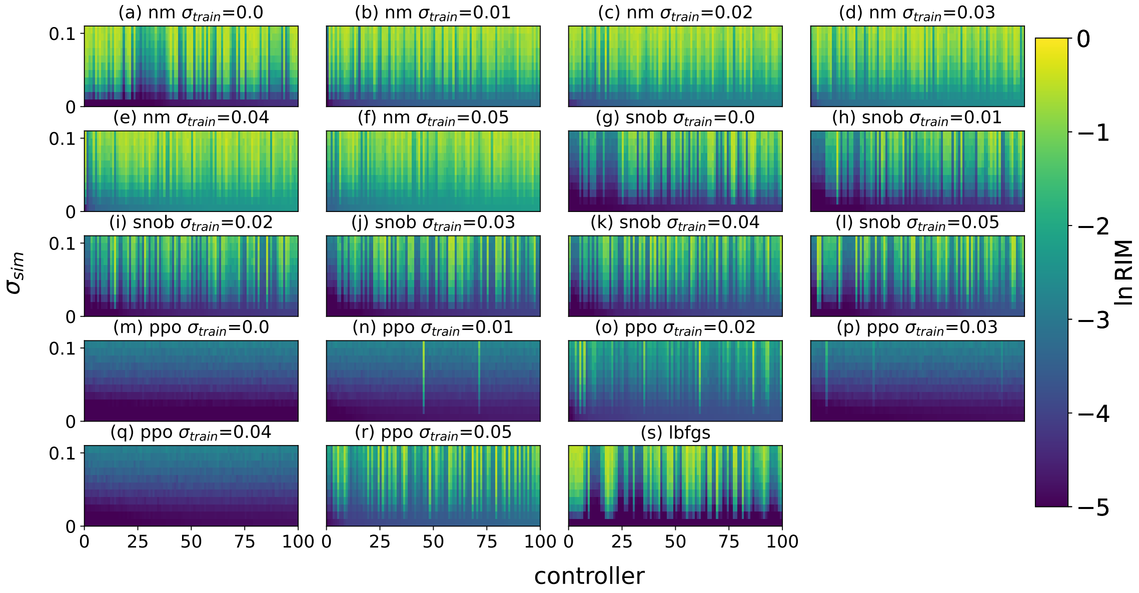


Figure A.3: Individual-controller comparison between (a)-(f) Nelder-Mead, (g)-(k) SNOBFit and (m)-(r) PPO with $\sigma_{\text{train}} = 0, 0.01, \ldots, 0.05$, using 100 controllers ranked by lowest infidelity (left) for the case $M = 5$ and the spin transition from $|1\rangle$ to $|3\rangle$. (s) shows the L-BFGS results for the same spin transition problem.

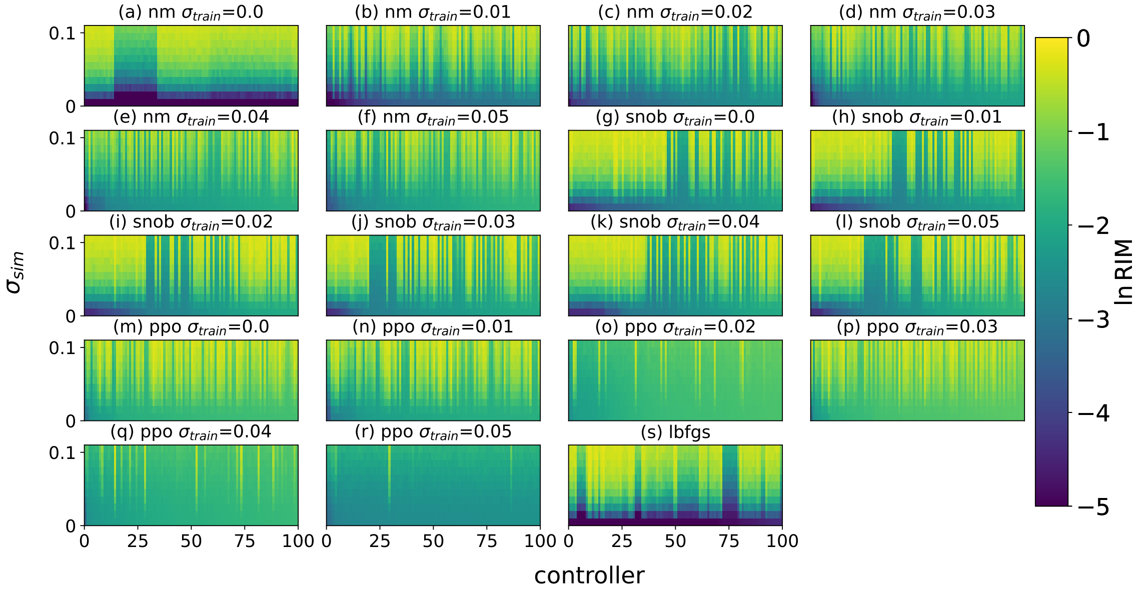


Figure A.4: Individual-controller comparison between (a)-(f) Nelder-Mead, (g)-(k) SNOBFit, (m)-(r) PPO with $\sigma_{\text{train}} = 0, 0.01, \ldots, 0.05$, using 100 controllers ranked by lowest infidelity for the case $M = 5$ and the spin transition from $|1\rangle$ to $|5\rangle$. (s) shows the L-BFGS result for $\sigma_{\text{train}} = 0$.

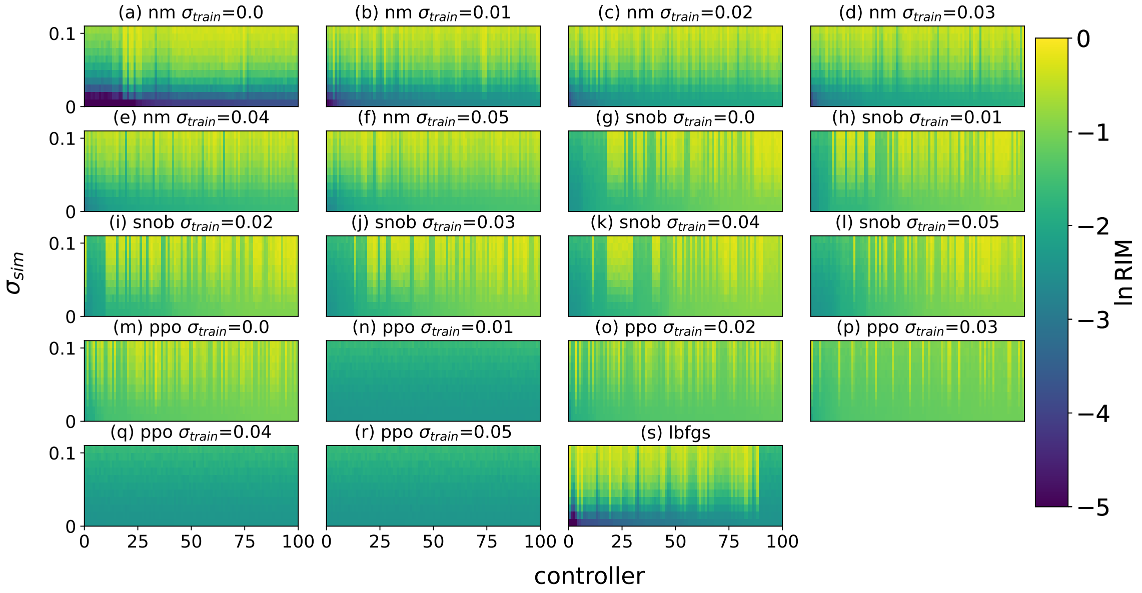


Figure A.5: Individual-controller comparison between (a)-(f) Nelder-Mead, (g)-(k) SNOBFit, (m)-(r) PPO with $\sigma_{\text{train}} = 0, 0.01, \ldots, 0.05$, using 100 controllers ranked by lowest infidelity for the case $M = 6$ and the spin transition from $|1\rangle$ to $|6\rangle$. (s) shows the L-BFGS result for $\sigma_{\text{train}} = 0$.

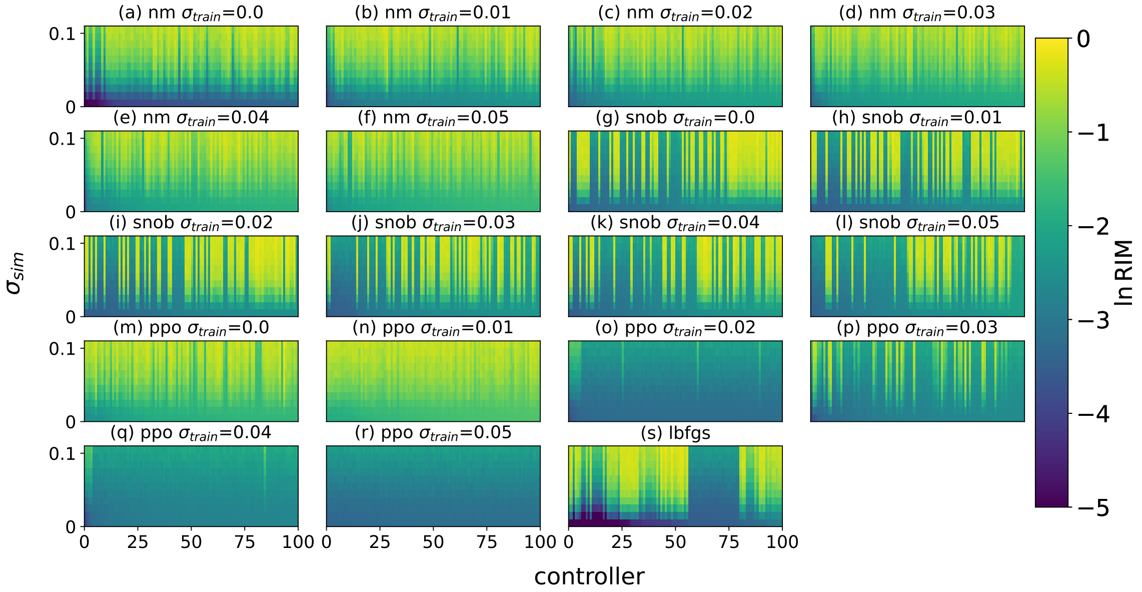


Figure A.6: Individual-controller comparison between (a)-(f) Nelder-Mead, (g)-(k) SNOBFit, (m)-(r) PPO with $\sigma_{\text{train}} = 0, 0.01, \ldots, 0.05$, using 100 controllers ranked by lowest infidelity for the case $M = 6$ and the spin transition from $|1\rangle$ to $|4\rangle$. (s) shows the L-BFGS result for $\sigma_{\text{train}} = 0$.

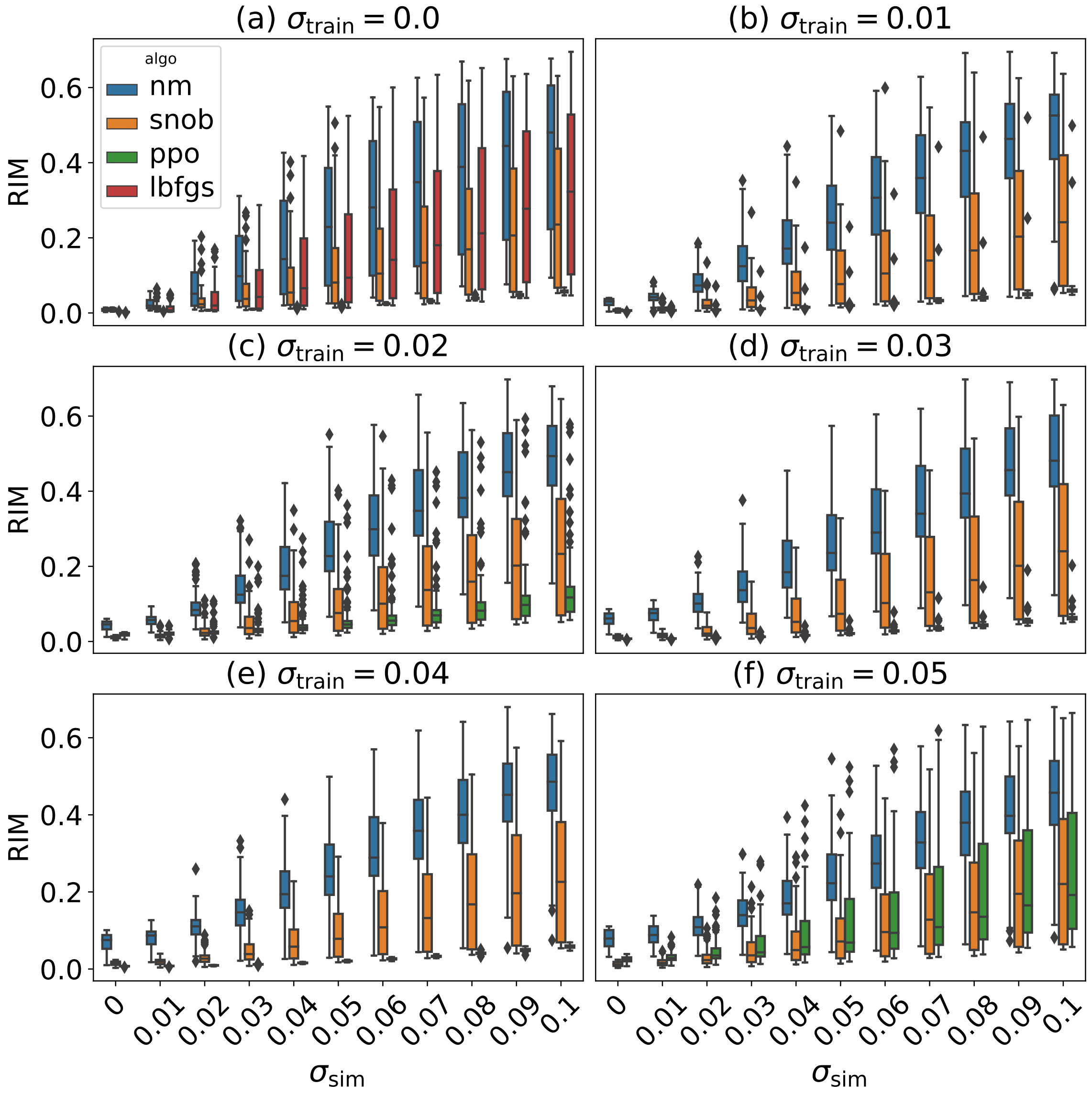


Figure A.7: Box plots of the RIM for the 100 controllers for $M = 5, |1\rangle$ to $|3\rangle$ shown in Fig. A.3 found by Nelder-Mead, SNOBFit, and PPO for various σ_{train} (a)-(f). For the case $\sigma_{\text{train}} = 0$ in (a), we also show L-BFGS box plots as a reference. On the distributional level, PPO controllers are generally the more robust of the three w.r.t. the RIM, but there is high variance across σ_{train} compared to the SNOBFit and Nelder-Mead controllers. The median SNOBFit RIM value per σ_{sim} is higher than L-BFGS, so it has a longer left tail. The Nelder-Mead controllers have the most weight on their right tails and are comparatively the worst.

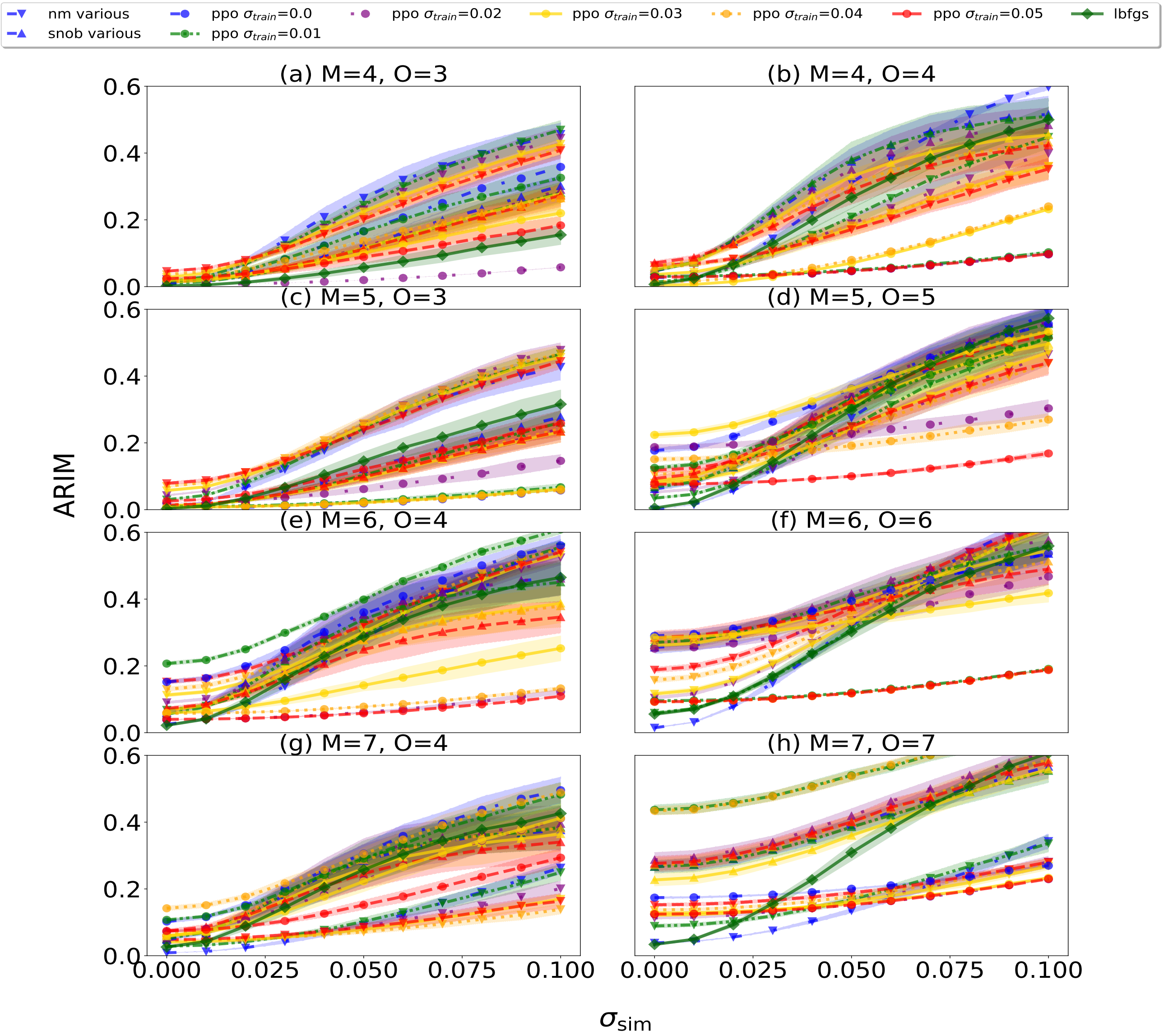


Figure A.8: ARIM as a function of σ_{sim} for $M = 4, 5, 6, 7$ where the left column contains end-to-middle transitions and the right column contains end-to-end transitions. The final state is denoted by O. The ARIM is computed from a distribution of RIM values for 100 controllers for each σ_{sim} for SNOBFit, Nelder-Mead, PPO and L-BFGS indicated by their marker shapes and line-styles. Both PPO and SNOBFit are run multiple times at $\sigma_{\mathrm{train}} = 0, 0.01, \ldots, 0.05$ which is indicated by the color of the ARIM curve. For all problems, PPO has higher variance with respect to σ_{train} than SNOBFit and Nelder-Mead. The latter pair's performance curves are more in line with the L-BFGS curve for $\sigma_{\mathrm{sim}} \geqslant 0.05$ and mostly worse for $\sigma_{\mathrm{sim}} \leqslant 0.05$. For most of the problems the best performing (lowest) curve across all problems is PPO at $\sigma_{\mathrm{train}} = 0.05$ (brown) except in (a) where it is PPO at $\sigma_{\mathrm{train}} = 0.02$ and in (g) where it is Nelder-Mead at $\sigma_{\mathrm{train}} \geqslant 0.04$. 95% confidence intervals (shading) are computed using non-parametric bootstrap resampling [Efr87] with 100 resamples.

Appendix B

Sample-efficient RL: additional results

B.1 Technical proofs

Lemma B.1. *H where $n = rank(H)$ admits a real decomposition $\{b_i\}_{i=1}^{n^2}$ in terms of the traceless hermitian basis elements $\{B_i\}_{i=1}^{n^2}$ of SUn.*

Proof: Note that by definition $H = H^\dagger$ is Hermitian. Then, it is easy to see that for b_i,

$$\begin{aligned} b_i^\dagger &= \mathrm{Tr}[B_i H(t)]^\dagger \\ &= \mathrm{Tr}\left[(B_i H(t))^\dagger\right] \quad \text{using trace linearity} \\ &= \mathrm{Tr}\left[H(t)^\dagger B_i^\dagger\right] \\ &= \mathrm{Tr}\left[B_i^\dagger H(t)^\dagger\right] \quad \text{using the trace's cyclic property} \\ &= \mathrm{Tr}[B_i H(t)]. \end{aligned}$$

□

B.2 Mapping complex linear ODEs to coupled real ODEs and step-size effects

The quantum control problem in Eqs. Eq. (2.10) and Eq. (2.19) involve ODEs (Eqs. Eq. (2.8), Eq. (2.13)) in the complex domain with a complex vector field map $f_\theta : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d$. For the unitary control problem we have a linear map $f_\theta(U(\mathbf{u}(t), t), t) = H_\theta(\mathbf{u}(t), t)U(\mathbf{u}(t), t)$

where H_θ is a Hermitian Hamiltonian that generates the ODE path of the propagator $U(t)$. We make use of the following isomorphism to map the complex ODE to two coupled real ODEs in $\mathbb{R}^{2d}$ by separating the propagator into its real and imaginary parts $U = U_{\text{real}} + iU_{\text{imag}}$ and mapping the Hamiltonian isomorphically $H(\mathbf{u}(t), t) \xrightarrow{\sim} \mathbb{1} \otimes H_{\text{real}}(\mathbf{u}(t), t) - i\sigma_y \otimes H_{\text{imag}}(\mathbf{u}(t), t)$, to get the following [Leu+17] coupled real ODE system,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} U_{\text{real}}(\mathbf{u}(t), t) \\ U_{\text{imag}}(\mathbf{u}(t), t) \end{pmatrix} = \begin{pmatrix} H_{\text{imag}}(\mathbf{u}(t), t) & H_{\text{real}}(\mathbf{u}(t), t) \\ -H_{\text{real}}(\mathbf{u}(t), t) & H_{\text{imag}}(\mathbf{u}(t), t) \end{pmatrix} \begin{pmatrix} U_{\text{real}}(\mathbf{u}(t), t) \\ U_{\text{imag}}(\mathbf{u}(t), t) \end{pmatrix}. \tag{B.1}$$

The mapping is analogous for the superoperator ODE in Eq. (2.13). Likewise, various other metrics, e.g., the fidelity $\mathcal{F}$, were analogously transformed. We made use of the real nature of the Pauli vector decomposition of H to keep track of both the time-independent learnable Hamiltonian and the time-dependent control Hamiltonian representations.

We use Heun's method [SM03] to implement a custom differentiable numerical ODE solver in `pytorch` [Pas+19], a popular automatic differentiation code library. The solver is able to evolve multiple ODEs under multiple generators in parallel using generalised matrix/tensor operations (ideally on a GPU to maximally leverage computational efficiency). The solver can be accessed in the `LearnableHamiltonian` module in our code [Kha22]. To determine the optimal tradeoff between accuracy of dynamical simulation, computed gradients and the size of the computation graph that is held in memory for automatic differentiation, we conduct experiments by simulating the dynamics of random n-qubit Hamiltonians from $n = 1$ to $n = 4$ at different precision or tolerance or step size of the ODE solver (see Fig. B.1).

Computational speed of the solver naturally trades off with the accuracy in the simulation and the computed gradients. We find that a step size of 10^{-2} is sufficiently accurate for forward dynamical simulation (no gradients are computed in this step) and a step size of 5×10^{-4} is required for the backward step when the gradients need to be computed to train the ODE model. The errors in the dynamical predictions (averaged over many thousands of data points) in both steps are reasonably small and are monitored. The ODE solvers in `scipy` [Vir+20] and the matrix exponential method for solving linear ODEs [Mac+11a] both have similar errors than our method for the step size 5×10^{-4} (likely the Bayes' optimal error for our numerical simulation).

The ability to be fast, but produce slightly less accurate predictions, improved the wall time of our algorithm. Specifically, a significantly large number of trajectories can be quickly sampled in the forward step to augment the RL policy's training data while the much slower backward step can be limited to a smaller number of trajectories that need to be predicted and are divided over multiple batches.

B.3 Comparison of fidelities for lindbladian dynamics

We study the agreement between three different fidelity measures of realised noisy gates on open systems with Lindblad decay and decoherence for the two-qubit transmon gate control problem. The fidelity measures are the diamond norm fidelity [BS10], the generalised state fidelity [FL11a], and the average gate fidelity [Uhl00]. The diamond norm fidelity, derived from the diamond norm or the completely bounded trace norm, is the most expensive to compute as it involves solving a convex optimization problem:

$$\begin{aligned}\mathcal{F}_\diamond(\mathbf{\Phi}(\mathbf{u}(t),t),\mathbf{\Phi}_{\text{target}}) &= 1-\|\mathbf{\Phi}(\mathbf{u}(t),t)-\mathbf{\Phi}_{\text{target}}\|_\diamond &\text{(B.2)}\\ &= 1-\max_\rho \|\mathbf{\Phi}(\mathbf{u}(t),t)(\rho)-\mathbf{\Phi}_{\text{target}}(\rho)\|_1 &\text{(B.3)}\end{aligned}$$

where the maximization is over the space of all density matrices ρ. This can be done by solving an equivalent semi-definite program [Wat09]. Note that $0.5 \leqslant \mathcal{F}_\diamond(\mathbf{\Phi}(\mathbf{u}(t),t) \leqslant 1$.

To study the sensitivities of the measures to dissipation and their agreement w.r.t. each other, we consider low, medium and high dissipation regimes. We evaluate 100 of our controllers found for the noisy shots setting of the two-qubit transmon in these regimes. The results are plotted in Fig. B.2. Here, `deca` and `deco` refer to inverse decoherence and decay rates $2/T_l^*, 2/T_l$ respectively, for the lth qubit and are measured in MHz. Note that we renormalize the trace of the realised operator $\mathbf{\Phi}(\mathbf{u}(t),t)$ during our experiments, as is standard practice. Due to the exhaustive nature of its computation, $\mathcal{F}_\diamond$ is the most sensitive to noise and loss of coherence out of all the measures. The generalised state fidelity is the least sensitive and the average gate fidelity falls in the middle. For very low to medium dissipation levels, e.g., $(0.05, 0.05)$, $(0.05, 0.1)$, or $(0.05, 0.2)$ for the pair $(\texttt{deca}, \texttt{deco})$, the generalised state fidelity is near perfect while the gate and diamond norm fidelities are more sensitive and closer to 0.9. For this reason, in Chapter 6.3.4, we chose to use the diamond norm fidelity to more accurately gauge controller performance – this was especially true for the low dissipation regime results.

As a side note, some controllers shown in Fig. B.2 are more robust to dissipation than others as revealed by the noisy variation across the controller index vs. fidelity plot. The controllers are not ordered, so the fidelity in the zero dissipation regime has some noise/variation as seen for $\texttt{deca}, \texttt{deco} = (0.05, 0.05)$. Across all the subfigures, the robustness is captured by all the fidelity measures where the variation magnitudes and positions are more or less aligned.

B.4 Leveraging the learned Hamiltonian for the two-qubit NV Center

Similar to the results found in Chapter 6.3.3, here we report the structural differences between the learned and target Hamiltonians for the two-qubit NV center.

The matrix difference between the true H_0 and learned Hamiltonian $H_0(\boldsymbol{\zeta})$ is

$$H - H_0(\boldsymbol{\zeta}) =$$
$$\begin{bmatrix} 0.0116 & 0.0013i & -0.0001 - 0.0002i & -0.0007 \\ -0.0013i & -0.0111 & -0.0001 + 0.0002i & 0.0003 + 0.0003i \\ -0.0001 + 0.0002i & 0.0001 + 0.0002i & -0.0108 & -0.0005 - 0.0002i \\ -0.0007 & 0.0003 - 0.0003i & -0.0005 + 0.0002i & -0.013 \end{bmatrix}.$$

Moreover, the non-linear relationship between the model prediction errors and the spectral norm error δ or the mean squared Pauli expectation value error is confirmed as before in Fig. B.3(a). Local and global trajectory differences under a random control pulse and the results of using GRAPE on RL controllers are shown in Fig. B.3(b) and (c) respectively. The learned Hamiltonian is able to improve the controller fidelities to greater than 0.999.

B.5 Three-qubit transmon control problem

In this section we discuss the issue of scalability of LH-MBSAC's performance related to the three-qubit transmon control problem in Chapter 6.3.5 in detail.

Working with two level systems, we extend the two-qubit transmon Hamiltonian to its three-qubit version $H^{(3)}_{\text{tra}}$. The system part generalizes trivially. For the control part $H^{(3)}_{\text{tra}_c}$, we generalize the cross resonance interaction presented in Ref. [She+16] to construct the following time-dependent part of the three-qubit transmon Hamiltonian,

$$\frac{H^{(3}_{\text{tra}_c}(t)}{\hbar} = \sum_{l=1}^{3} \Big(a_l(t)(Z_l X_{l+1} + X_{l+1} + Y_{l+1} + Z_l) \\ + b_l(t)(X_l Z_{l+1} + X_l + Y_l + Z_{l+1}) \Big) \qquad \text{(B.4)}$$

where $a_l(t), b_l(t)$ are the real drive amplitudes and X_l, Y_l, Z_l are the corresponding Pauli operators on the lth qubit.

To start, we mention our hyperparameter strategy. Only an initial hyperparameter search is performed for the two-qubit transmon control problem, and we were successfully able to transfer the same hyperparameters to all problems in the paper that were studied including the ones presented in Fig. 6.3.

It is a desirable property for the stabiltiy of RL algorithms to be robust to hyperparameter changes for different target problems, which we found to be the case. The search was only conducted for the model-free SAC since LH-MBSAC is just a model-based augmentation of the underlying SAC algorithm so there is no strong reason for the hyperparameters to fail to transfer.

However, for the three-qubit transmon control problem, we encountered issues and had to repeat the search. This was extensive, and what we focused on are: more initial exploration data, using bigger layer sizes for the policy and value function neural networks, changing the learning and update rates for the policy and value functions, amongst other things. An extremely thorough search is difficult since the problem is more computationally challenging, and it is hard to determine when to terminate the training during a trial run that necessarily needs to be premature during the hyperparameter search. Please see the accompanying code for the list of hyperparameters we searched over using Bayesian optimization in `tune_hypers.py` along with some results in the `hyper_tests` folder [Kha22].

Furthermore, we make observations that make this issue seem less like a hyperparameter issue and more like an optimization landscape problem:

1. The values and the gradients for policy and value functions that saturate are both stuck in suboptimal extrema and ultimately we get stuck at a prematurely optimized reward function. This is illustrated in Fig. B.4. Essentially, SAC gets stuck in a loop mining medium level fidelities and its policy outputs saturate on the extremes of the control amplitudes. It is already detailed in Chapter 6.3.5

that RL pulses are biased towards maintaining high intermediate fidelities due to the nature of the MDP used in this thesis. Fig. 6.10 example pulses found by RL vs. GRAPE for the two-qubit transmon, confirming this.

2. Since we have the model Hamiltonian, we insert it into GRAPE initialized with the highest fidelity SAC controller values, and it also gets stuck (at slightly better fidelities).

Despite these issues, the system Hamiltonian is still learned. It can be inserted into GRAPE with uniform random initialization of control pulse parameters to achieve fidelities of over 0.999.

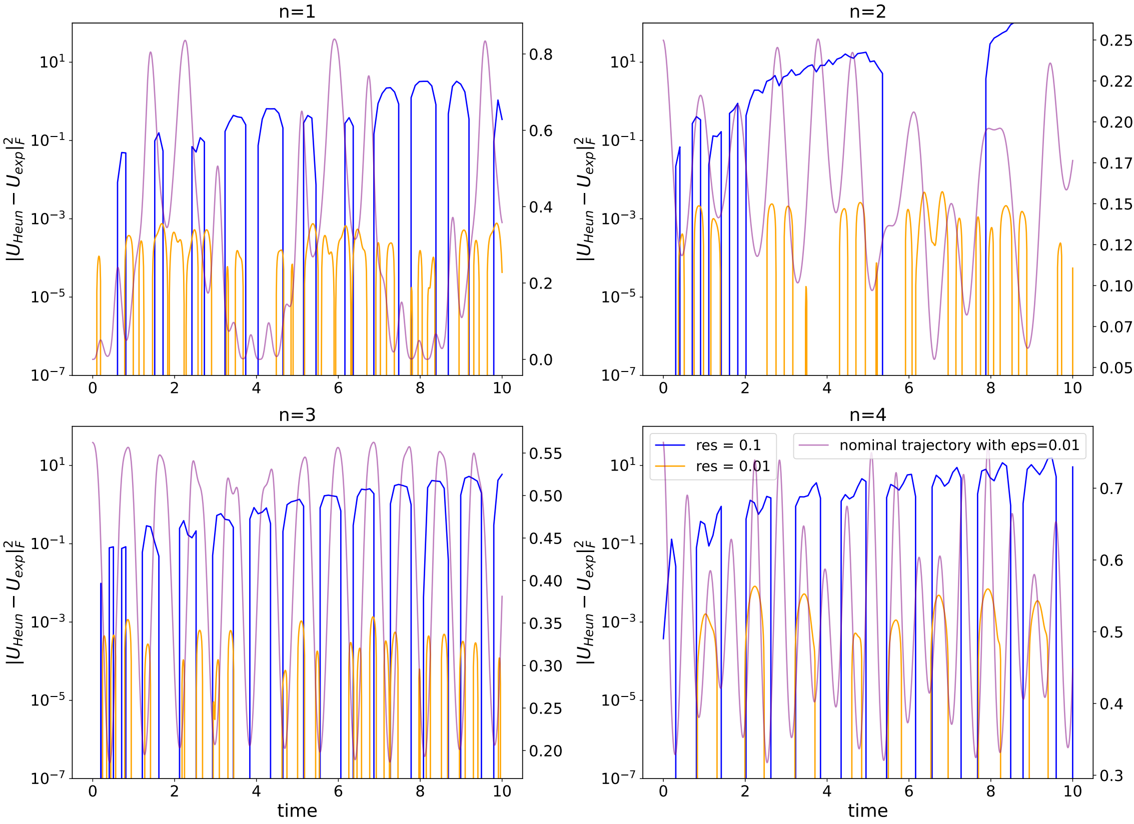


Figure B.1: Frobenius norm of the prediction error of the Heun ODE solver [SM03] compared to the matrix exponential method. The number of qubits n is shown on top of each subfigure. The random time-dependent sinusoidal Hamiltonians are as follows: for $n = 1$, $H = -2.32\sigma_z \cos 2.19t - 0.01\mathbb{1} \sin 3.62t + 1.79\sigma_x \cos 4.89t + 3.04\sigma_y \cos 2.69t$; for $n = 2$, $H = 1.01\sigma_z\mathbb{1} \cos 1.44t + 4.51\mathbb{1}\mathbb{1} \sin 4.55t - 2.7\sigma_y\sigma_z \sin 1.07t + 0.48\sigma_x\sigma_z \cos 2.26t$; for $n = 3$, $H = -1.28\mathbb{1}\sigma_x\mathbb{1} \cos 2.62t - 0.23\sigma_y\sigma_z\sigma_y \sin 3.75t - 1.34\mathbb{1}\sigma_y\sigma_x \sin 3.35t + 3.38\sigma_x\sigma_x\sigma_z \cos 2.34t$; for $n = 4$, $H = -0.41\mathbb{1}\sigma_z\sigma_z\sigma_x \sin 2.86t + 2.19\sigma_y\mathbb{1}\sigma_x\sigma_z \sin 1.38t - 0.87\sigma_y\sigma_x\sigma_x\sigma_z \sin 2.26t + 4.06\sigma_x\sigma_x\sigma_z\mathbb{1} \sin 1.76t$ where the shorthand used is $\mathbb{1}\sigma_x\mathbb{1} \equiv \mathbb{1} \otimes \sigma_x \otimes \mathbb{1}$. Trace fidelities w.r.t. the generalized CNOT (NOT or X-gate for $n = 1$, CNOT for $n = 2$, CCNOT for $n = 3$ and so on) are shown in the twin axis on the right. It can be seen that the step size of 10^{-1} leads to quick accumulation of error seen in the sharp peaks but a step size of 10^{-2} is more stable with more than $O(10^3)$ times less prediction error.

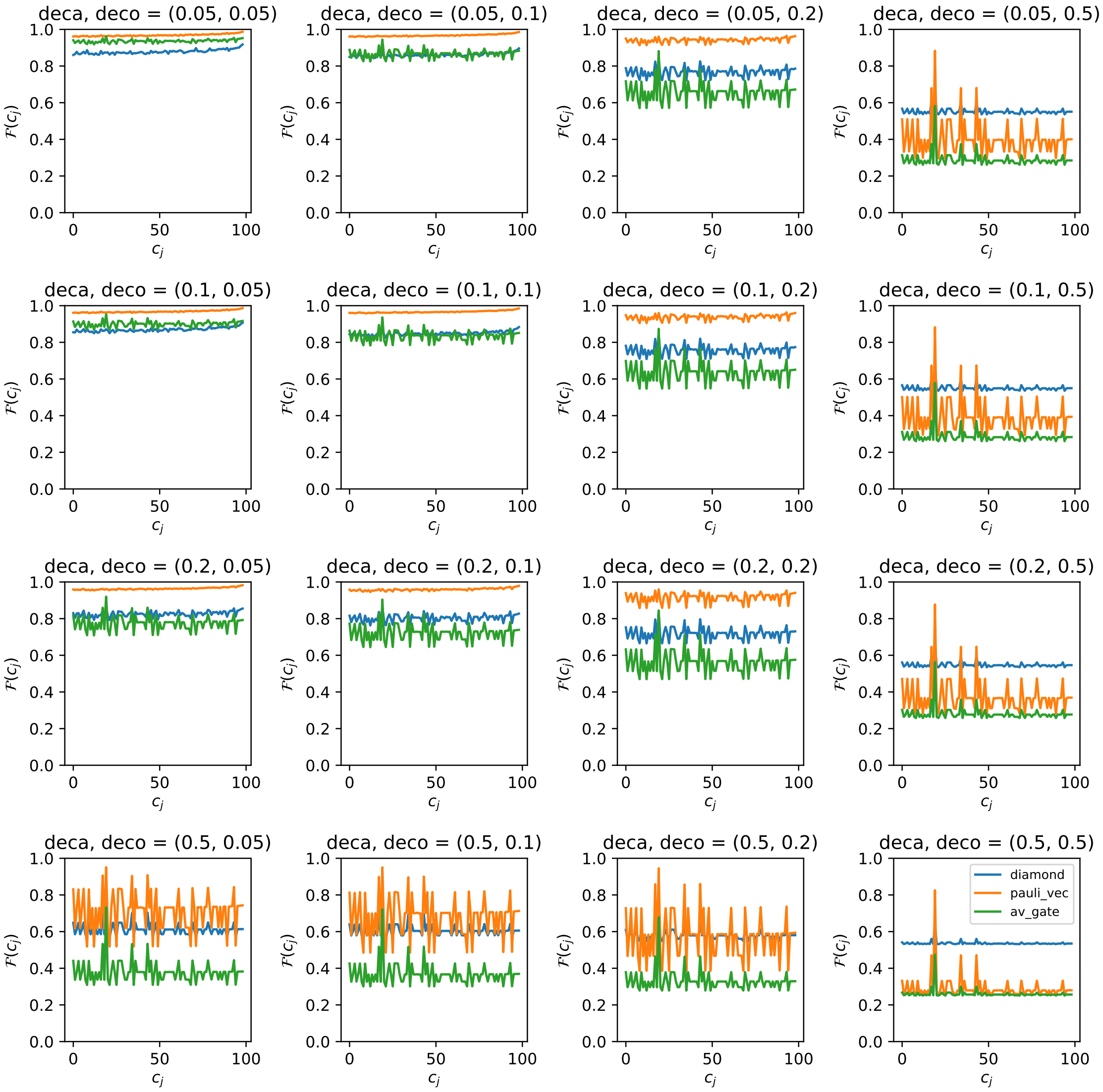


Figure B.2: How much the fidelity measures relate to one another as the dissipation strength varies in terms of the decoherence and the decay coefficients in Eq. (2.7) for the Lindbladian operators l_d. Here, deca, deco refer to inverse decay and decoherence rates $2/T_l^*, 2/T_l$ respectively, for the lth qubit and are measured in MHz. The x-axis refers to a controller c_j obtained for the two-qubit transmon gate control problem with shots noise where the target is the CNOT gate. The controllers are in random order w.r.t. the fidelity but the ordering is preserved across each subfigure. The number of shots is 10^6 and diamond, pauli_vec, av_gate refer to the diamond norm fidelity [BS10], the generalised state fidelity [FL11a] and the average gate fidelity [Uhl00].

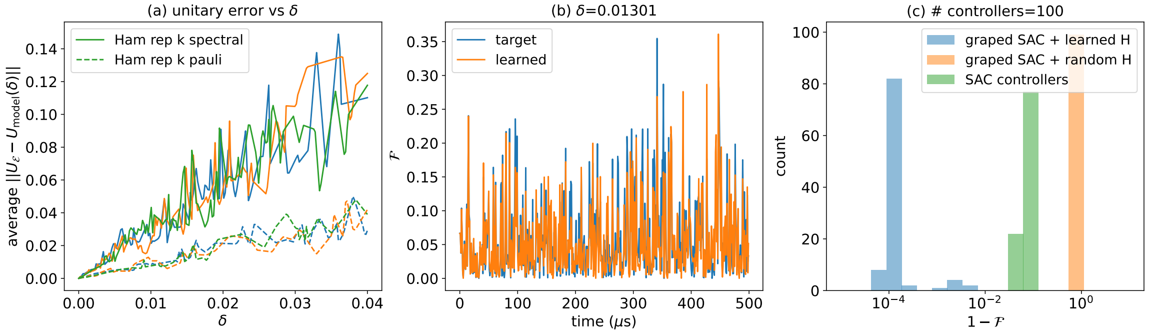


Figure B.3: (a) The non-linear relationship between the prediction error $\left\|U_\mathcal{E} - U_{\mathbf{M}_\zeta}\right\|$ and Hamiltonian spectral norm error or mean squared Pauli expectation value error δ for the two-qubit NV center Hamiltonian. For the same 1,000 random control pulses, we evaluate the average unitary prediction error of $\mathbf{M}_\zeta$ with increasing δ for three different uniform randomly sampled two-qubit Hamiltonians $H_0(\boldsymbol{\zeta})$. (b) Local and global unitary trajectories: $\mathcal{F}$ as a function of a random control pulse with either the learned $H_0(\boldsymbol{\zeta})$ or true H_0. The learned trajectories and global trajectory overlap less with increasing time with the spectral norm error of $\delta = 0.01301$ and a global phase factor $\mathrm{Tr}[H - H_0(\boldsymbol{\zeta})]$ of ~ 0.01. (c) The learned $H_0(\boldsymbol{\zeta})$ can be leveraged using GRAPE to further optimize the fidelities of LH-MBSAC's controllers. Repeating the procedure in Chapter 6.3.3, yields fidelities of greater than 0.999.

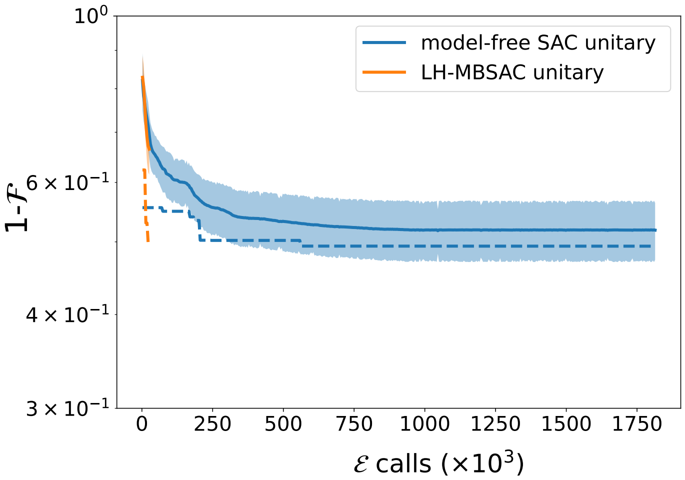


Figure B.4: Noiseless unitary sample complexity for the three-qubit transmon where the target gate is the Toffoli gate. Since LH-MBSAC is based on SAC, the latter's training curves are obtained first to see if it viably solves the problem, and it was trained for much longer, i.e., in the order of millions of samples as seen in Fig. B.4. Mean (solid) and maximum fidelities (dashed) saturate as the policy and value function gradients and outputs saturate due to the agent getting stuck in a suboptimal extremum of the optimization landscape.

Bibliography

[Ach+23] R. Acharya et al. "Suppressing quantum errors by scaling a surface code logical qubit". In: *Nature* 614.7949 (2023), pp. 676–681.

[Aci+18] A. Acin et al. "The quantum technologies roadmap: a European community view". In: *New Journal of Physics* 20.8 (2018), p. 080201.

[Agr10] A. Agresti. *Analysis of ordinal categorical data.* Vol. 656. John Wiley & Sons, 2010.

[Ahm+18] A. Ahmadov et al. "Analytical solutions of the Schrödinger equation for the Manning-Rosen plus Hulthén potential within SUSY quantum mechanics". In: *Journal of Physics: Conference Series.* Vol. 965. 1. IOP Publishing. 2018, p. 012001.

[Alt+03] J. B. Altepeter et al. "Ancilla-Assisted Quantum Process Tomography". In: *Phys. Rev. Lett.* 90 (19 May 2003), p. 193601.

[Ani+21] M. S. Anis et al. "Qiskit: An open-source framework for quantum computing". In: *Qiskit/qiskit* (2021).

[AGS21] Q. Ansel, S. J. Glaser, and D. Sugny. "Selective and robust time-optimal rotations of spin systems". In: *J. Phys. A: Mathematical and Theoretical* 54.8 (2021), p. 085204.

[Ans+21] A. Anshu et al. "Sample-efficient learning of interacting quantum systems". In: *Nature Physics* 17.8 (2021), pp. 931–935. DOI: 10.1038/s41567-021-01232-0.

[ACB17] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein generative adversarial networks". In: *Int. Conf. Machine Learning.* 2017, pp. 214–223.

[Aru+19] F. Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510.

[AO20] A. Asl and M. L. Overton. *Behavior of Limited Memory BFGS when Applied to Nonsmooth Functions and their Nesterov Smoothings.* 2020. arXiv: 2006.11336.

[Ass+98] A. Assion et al. "Control of Chemical Reactions by Feedback-Optimized Phase-Shaped Femtosecond Laser Pulses". In: *Science* 282.5390 (1998), pp. 919–922.

[ATL15] K. Azuma, K. Tamaki, and H.-K. Lo. “All-photonic quantum repeaters”. In: *Nature communications* 6.1 (2015), p. 6787.

[Bal+21] H. Ball et al. “Software tools for quantum control: improving quantum computer performance through noise and error suppression”. In: *Quantum Science and Technology* 6.4 (Sept. 2021), p. 044011.

[Bar+98] C. J. Bardeen et al. “Quantum control of population transfer in green fluorescent protein by using chirped femtosecond pulses”. In: *Journal of the American Chemical Society* 120.50 (1998), pp. 13023–13027.

[Bar+14] R. Barends et al. “Superconducting quantum circuits at the surface code threshold for fault tolerance”. In: *Nature* 508.7497 (2014), pp. 500–503.

[BMF21] R. E. Barfknecht, T. Mendes-Santos, and L. Fallani. “Engineering entanglement Hamiltonians with strongly interacting cold atoms in optical traps”. In: *Phys. Rev. Research* 3 (1 Feb. 2021), p. 013112.

[BD12] E. Barnes and S. Das Sarma. “Analytically Solvable Driven Time-Dependent Two-Level Quantum Systems”. In: *Phys. Rev. Lett.* 109 (6 2012), p. 060401.

[BWS15] E. Barnes, X. Wang, and S. D. Sarma. “Robust quantum control using smooth pulses and topological winding”. In: *Scientific Reports* 5.1 (2015), pp. 1–6.

[Bar+22] E. Barnes et al. “Dynamically corrected gates from geometric space curves”. In: *Quantum Science and Technology* 7.2 (2022), p. 023001.

[BK02] H. Barnum and E. Knill. “Reversing quantum dynamics with near-optimal quantum and classical fidelity”. In: *J. Math. Phys.* 43.5 (2002), pp. 2097–2106. DOI: 10.1063/1.1459754.

[Bea+18] S. J. Beale et al. “Quantum Error Correction Decoheres Noise”. In: *Phys. Rev. Lett.* 121 (19 Nov. 2018), p. 190501.

[BNT12] M. Bellec, G. M. Nikolopoulos, and S. Tzortzakis. “Faithful communication Hamiltonian in photonic lattices”. In: *Opt. Lett.* 37.21 (Nov. 2012), pp. 4504–4506.

[Bel52] R. Bellman. *On the theory of dynamic programming.* Tech. rep. 8. 1952, pp. 716–719.

[Bel+11] V. Beltrani et al. “Exploring the top and bottom of the quantum control landscape”. In: *The Journal of chemical physics* 134.19 (2011), p. 194106. DOI: 10.1063/1.3589404.

[BS10] G. Benenti and G. Strini. “Computing the distance between quantum channels: usefulness of the Fano representation”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 43.21 (2010), p. 215508.

[BK08] R. A. Bertlmann and P. Krammer. “Bloch vectors for qudits”. In: *Journal of Physics A: Mathematical and Theoretical* 41.23 (2008), p. 235303.

[Ber19] D. Bertsekas. *Reinforcement learning and optimal control.* Athena Scientific, 2019.

[BT96] D. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming.* Athena Scientific, 1996.

[BDN12] I. Bloch, J. Dalibard, and S. Nascimbene. "Quantum simulations with ultracold quantum gases". In: *Nature Physics* 8.4 (2012), pp. 267–276.

[Blo18] I. Bloch. "Quantum simulations come of age". In: *Nature Physics* 14.12 (2018), pp. 1159–1161.

[Blu+21a] R. Blumel et al. *Power-optimal, stabilized entangling gate between trapped-ion qubits.* 2021. arXiv: 1905.09292.

[Blu+21b] D. Bluvstein et al. "Controlling quantum many-body dynamics in driven Rydberg atom arrays". In: *Science* 371.6536 (2021), pp. 1355–1359.

[Blu+21c] D. Bluvstein et al. "Controlling quantum many-body dynamics in driven Rydberg atom arrays". In: *Science* 371.6536 (2021), pp. 1355–1359.

[Bos07] S. Bose. "Quantum communication through spin chain dynamics: an introductory overview". In: *Contemporary Physics* 48.1 (2007), pp. 13–30.

[Bra92] S. L. Braunstein. "Quantum limits on precision measurements of phase". In: *Phys. Rev. Lett.* 69 (25 1992), pp. 3598–3601. DOI: 10.1103/PhysRevLett.69.3598.

[BC94] S. L. Braunstein and C. M. Caves. "Statistical distance and the geometry of quantum states". In: *Phys. Rev. Lett.* 72 (22 1994), pp. 3439–3443. DOI: 10.1103/PhysRevLett.72.3439.

[BP+02] H.-P. Breuer, F. Petruccione, et al. *The theory of open quantum systems.* Oxford University Press on Demand, 2002.

[Bri+98] H.-J. Briegel et al. "Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication". In: *Phys. Rev. Lett.* 81 (26 1998), pp. 5932–5935.

[BCR10] C. Brif, R. Chakrabarti, and H. Rabitz. "Control of quantum phenomena: past, present and future". In: *New J. Phys.* 12.7 (2010), p. 075008.

[Bro+21] M. M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.* 2021.

[Bro+15] M. Brownnutt et al. "Ion-trap measurements of electric-field noise near surfaces". In: *Rev. Mod. Phys.* 87 (4 Dec. 2015), pp. 1419–1482.

[Buc+18] J. Buckman et al. "Sample-efficient reinforcement learning with stochastic ensemble value expansion". In: *Advances in neural information processing systems* 31 (2018).

[Buk+18] M. Bukov et al. "Reinforcement Learning in Different Phases of Quantum Control". In: *Phys. Rev. X* 8 (3 Sept. 2018), p. 031086.

[Bur+22] D. Burgarth et al. "One bound to rule them all: from Adiabatic to Zeno". In: *Quantum* 6 (2022), p. 737.

[Bur07] D. K. Burgarth. “Quantum state transfer with spin chains”. PhD thesis. 2007.

[BDB21] D. Buterakos, S. Das Sarma, and E. Barnes. “Geometrical Formalism for Dynamically Corrected Gates in Multiqubit Systems”. In: *PRX Quantum* 2 (1 Mar. 2021), p. 010341.

[BGN00] R. H. Byrd, J. C. Gilbert, and J. Nocedal. “A trust region method based on interior point techniques for nonlinear programming”. In: *Mathematical Programming* 89.1 (2000), pp. 149–185.

[CS96] A. R. Calderbank and P. W. Shor. “Good quantum error-correcting codes exist”. In: *Phys. Rev. A* 54 (2 Aug. 1996), pp. 1098–1105.

[CH88] C. Carroll and F. T. Hioe. “Driven three-state model and its analytic solutions”. In: *Journal of mathematical physics* 29.2 (1988), pp. 487–509.

[Cav81] C. M. Caves. “Quantum-mechanical noise in an interferometer”. In: *Phys. Rev. D* 23 (8 1981), pp. 1693–1708. DOI: `10.1103/PhysRevD.23.1693`.

[Cet+20] M. Cetina et al. *Quantum Gates on Individually-Addressed Atomic Qubits Subject to Noisy Transverse Motion.* 2020. arXiv: `2007.06768`.

[Che+13] C. Chen et al. “Fidelity-based probabilistic Q-learning for control of quantum systems”. In: *IEEE transactions on neural networks and learning systems* 25.5 (2013), pp. 920–933.

[Che+14] C. Chen et al. “Sampling-based learning control of inhomogeneous quantum ensembles”. In: *Phys. Rev. A* 89 (2 Feb. 2014), p. 023402.

[Che+18] R. T. Q. Chen et al. “Neural Ordinary Differential Equations”. In: *Advances in Neural Information Processing Systems.* Vol. 31. Curran Associates, Inc., 2018.

[Che+22] S. Chen et al. *The Complexity of NISQ.* 2022. arXiv: `2210.07234 [quant-ph]`.

[CX20] T. Chen and Z.-Y. Xue. “High-Fidelity and Robust Geometric Quantum Gates that Outperform Dynamical Ones”. In: *Phys. Rev. Applied* 14 (6 Dec. 2020), p. 064009.

[Che+16] Z. Chen et al. “Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit”. In: *Phys. Rev. Lett.* 116 (2 Jan. 2016), p. 020501. DOI: `10.1103/PhysRevLett.116.020501`.

[CS91] R. Y. Chiang and M. G. Safonov. “Design of H∞ controller for a lightly damped system using a bilinear pole shifting transform”. In: *1991 American Control Conference.* 1991, pp. 1927–1928.

[Chi+19] C. S. Chiu et al. “String patterns in the doped Hubbard model”. In: *Science* 365.6450 (2019), pp. 251–256.

[Cho75] M.-D. Choi. “Completely positive linear maps on complex matrices”. In: *Linear algebra and its applications* 10.3 (1975), pp. 285–290.

[Chr+04] M. Christandl et al. “Perfect State Transfer in Quantum Spin Networks”. In: *Phys. Rev. Lett.* 92 (18 May 2004), p. 187902.

[Chu+18] K. Chua et al. “Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[CZ95] J. I. Cirac and P. Zoller. “Quantum computations with cold trapped ions”. In: *Phys. Rev. Lett.* 74 (1995), pp. 4091–4094.

[CKW22] M. CLOUÂTRÉ, M. J. Khojasteh, and M. Z. WIN. “Model-predictive quantum control via Hamiltonian learning”. In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–23.

[CL55] E. A. Coddington and N. Levinson. *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955.

[Col11] T. L. S. Collaboration. “A gravitational wave observatory operating beyond the quantum shot-noise limit”. In: *Nature Physics* 7.12 (2011), pp. 962–965.

[Cro18] A. Cross. “The IBM Q experience and QISKit open-source quantum computing software”. In: *APS March meeting abstracts*. Vol. 2018. 2018, pp. L58–003.

[Dal+22] A. J. Daley et al. “Practical quantum advantage in quantum simulation”. In: *Nature* 607.7920 (2022), pp. 667–676.

[DMS22] M. Dalgaard, F. Motzoi, and J. Sherson. “Predicting quantum dynamical cost landscapes with deep learning”. In: *Phys. Rev. A* 105 (1 Jan. 2022), p. 012402.

[Dal+20a] M. Dalgaard et al. “Global optimization of quantum dynamics with AlphaZero deep exploration”. In: *npj Quantum Information* 6.1 (2020), p. 6.

[Dal+20b] M. Dalgaard et al. “Global optimization of quantum dynamics with AlphaZero deep exploration”. In: *npj Quantum Information* 6.1 (2020), pp. 1–9.

[Dal+20c] M. Dalgaard et al. “Global optimization of quantum dynamics with AlphaZero deep exploration”. In: *npj Quantum Information* 6.1 (2020), pp. 1–9.

[DS13] P. De Fouquieres and S. G. Schirmer. “A closer look at quantum control landscapes and their implication for control optimization”. In: *Infinite dimensional analysis, quantum probability and related topics* 16.03 (2013), p. 1350021.

[DR98] M. Demiralp and H. Rabitz. “Assessing optimality and robustness of control over quantum dynamics”. In: *Phys. Rev. A* 57 (4 Apr. 1998), pp. 2420–2425.

[Deu85] D. Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[Din+23] L. Ding et al. *High-Fidelity, Frequency-Flexible Two-Qubit Fluxonium Gates with a Transmon Coupler.* 2023. arXiv: `2304.06087`.

[DP10a] D. Dong and I. R. Petersen. “Quantum control theory and applications: a survey”. In: *IET control theory & applications* 4.12 (2010), pp. 2651–2671.

[DP10b] D. Dong and I. R. Petersen. “Quantum control theory and applications: a survey”. In: *IET control theory & applications* 4.12 (2010), pp. 2651–2671.

[Don+13] D. Dong et al. “Sampling-based learning control for quantum systems with Hamiltonian uncertainties”. In: (2013), pp. 1924–1929.

[Don+21] W. Dong et al. “Doubly Geometric Quantum Control”. In: *PRX Quantum* 2 (3 Aug. 2021), p. 030333.

[Dor87] P. Dorato. “A historical review of robust control”. In: *IEEE Control Systems Magazine* 7.2 (1987), pp. 44–47. DOI: `10.1109/MCS.1987.1105273`.

[DB11] R. C. Dorf and R. H. Bishop. *Modern Control Systems Solution Manual.* Pearson Studium, London, 2011.

[DCM11] P. Doria, T. Calarco, and S. Montangero. “Optimal Control Technique for Many-Body Quantum Dynamics”. In: *Phys. Rev. Lett.* 106 (19 May 2011), p. 190501.

[Doy82] J. Doyle. “Analysis of feedback systems with structured uncertainties”. In: *IEE Proc. D Control Theory and Applications.* Vol. 129 (6). 1982, pp. 242–250.

[DLG20] G. Dridi, K. Liu, and S. Guérin. “Optimal Robust Quantum Control by Inverse Geometric Optimization”. In: *Phys. Rev. Lett.* 125 (25 Dec. 2020), p. 250403.

[Duf02] M. O. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes.* University of Massachusetts Amherst, 2002.

[DDT19] E. Dupont, A. Doucet, and Y. W. Teh. “Augmented Neural ODEs”. In: *Advances in Neural Information Processing Systems.* Vol. 32. Curran Associates, Inc., 2019.

[DOS23] A. Dutkiewicz, T. E. O’Brien, and T. Schuster. *The advantage of quantum control in many-body Hamiltonian learning.* 2023. arXiv: `2304.07172 [quant-ph]`.

[DKW56] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. “Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator”. In: *Annals of Mathematical Statistics* 27.3 (1956), pp. 642–669.

[Efr87] B. Efron. “Better bootstrap confidence intervals”. In: *J. American Statistical Association* 82.397 (1987), pp. 171–185.

[Ega21] L. N. Egan. “Scaling Quantum Computers with Long Chains of Trapped Ions”. PhD thesis. University of Maryland, 2021.

[EW14] D. J. Egger and F. K. Wilhelm. “Adaptive Hybrid Optimal Quantum Control for Imprecisely Characterized Systems”. In: *Phys. Rev. Lett.* 112 (24 June 2014), p. 240503.

[EAŻ05] J. Emerson, R. Alicki, and K. Życzkowski. “Scalable noise estimation with random unitary operators”. In: *Journal of Optics B: Quantum and Semiclassical Optics* 7.10 (2005), S347.

[EBL18] S. Endo, S. C. Benjamin, and Y. Li. “Practical Quantum Error Mitigation for Near-Future Applications”. In: *Phys. Rev. X* 8 (3 July 2018), p. 031027.

[Eri+19] D. Eriksson et al. “Scalable global optimization via local bayesian optimization”. In: *Advances in neural information processing systems* 32 (2019).

[EHF19] T. J. Evans, R. Harper, and S. T. Flammia. “Scalable Bayesian Hamiltonian learning”. In: *arXiv preprint arXiv:1912.07636* (2019).

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: `1411.4028`.

[Faw+22] A. Fawzi et al. “Discovering faster matrix multiplication algorithms with reinforcement learning”. In: *Nature* 610.7930 (2022), pp. 47–53.

[Fel60] A. A. Feldbaum. “Dual control theory. i & ii”. In: *Avtomatika i Telemekhanika* 21.9 (1960), pp. 1240–1249.

[Fey18] R. P. Feynman. “Simulating physics with computers”. In: *Feynman and computation*. CRC Press, 2018, pp. 133–153.

[FL11a] S. T. Flammia and Y.-K. Liu. “Direct Fidelity Estimation from Few Pauli Measurements”. In: *Phys. Rev. Lett.* 106 (23 June 2011), p. 230501.

[FL11b] S. T. Flammia and Y.-K. Liu. “Direct Fidelity Estimation from Few Pauli Measurements”. In: *Phys. Rev. Lett.* 106 (23 June 2011), p. 230501.

[FDS12] F. F. Floether, P. De Fouquieres, and S. G. Schirmer. “Robust quantum gates for open systems via optimal control: Markovian versus non-Markovian dynamics”. In: *New Journal of Physics* 14.7 (2012), p. 073023.

[Fou+11] P. de Fouquieres et al. “Second order gradient ascent pulse engineering”. In: *Journal of Magnetic Resonance* 212.2 (2011), pp. 412–417.

[Fra+17a] F. Frank et al. "Autonomous calibration of single spin qubit operations". In: *npj Quantum Information* 3.1 (2017), p. 48.

[Fra+17b] F. Frank et al. "Autonomous calibration of single spin qubit operations". In: *npj Quantum Information* 3.1 (2017), pp. 1–5.

[Fra+16] S. van Frank et al. "Optimal control of complex atomic quantum systems". In: *Scientific reports* 6.1 (2016), p. 34187.

[Fra18] P. I. Frazier. "A tutorial on Bayesian optimization". In: *arXiv preprint arXiv:1807.02811* (2018).

[FHM18] S. Fujimoto, H. Hoof, and D. Meger. "Addressing function approximation error in actor-critic methods". In: *International conference on machine learning.* PMLR. 2018, pp. 1587–1596.

[Gau+88] U. Gaubatz et al. "Population switching between vibrational levels in molecular beams". In: *Chemical physics letters* 149.5-6 (1988), pp. 463–468.

[GW21] X. Ge and R.-B. Wu. "Risk-sensitive optimization for robust quantum controls". In: *Phys. Rev. A* 104 (1 July 2021), p. 012422.

[GLM11] V. Giovannetti, S. Lloyd, and L. Maccone. "Advances in quantum metrology". In: *Nature photonics* 5.4 (2011), pp. 222–229.

[Gla+15] S. J. Glaser et al. "Training Schrödinger's cat: Quantum optimal control: Strategic report on current status, visions and goals for research in Europe". In: *The European Physical Journal D* 69 (2015), pp. 1–24.

[GCM22] M. H. Goerz, S. C. Carrasco, and V. S. Malinovsky. *Quantum Optimal Control via Semi-Automatic Differentiation.* 2022.

[Gol+22] A. J. Goldschmidt et al. *Model predictive control for robust quantum state preparation.* 2022.

[Got97] D. Gottesman. *Stabilizer codes and quantum error correction.* California Institute of Technology, 1997.

[Got09] D. Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation.* 2009. arXiv: `0904.2557`.

[GL14] T. Graß and M. Lewenstein. "Trapped-ion quantum simulation of tunable-range Heisenberg chains". In: *EPJ Quantum Technology* 1.8 (2014), pp. 1–20. DOI: `10.1140/epjqt8`.

[Gre+13] T. J. Green et al. "Arbitrary quantum control of qubits in the presence of universal noise". In: *New J. Phys.* 15.9 (Sept. 2013), p. 095004. DOI: `10.1088/1367-2630/15/9/095004`.

[GB17] C. Gross and I. Bloch. "Quantum simulations with ultracold atoms in optical lattices". In: *Science* 357.6355 (2017), pp. 995–1001.

[GSD12] A. Guez, D. Silver, and P. Dayan. "Efficient Bayes-adaptive reinforcement learning using sample-based search". In: *Advances in neural information processing systems* 25 (2012).

[HKT22] J. Haah, R. Kothari, and E. Tang. "Optimal learning of quantum Hamiltonians from high-temperature Gibbs states". In: *IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS).* 2022, pp. 135–146. DOI: `10.1109/FOCS54457.2022.00020`.

[Haa+18] T. Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning.* PMLR. 2018, pp. 1861–1870.

[HS17] M. Hausknecht and P. Stone. "Deep recurrent Q-learning for partially observable MDPs". In: *AAAI Fall Symposium Series.* 2017. arXiv: `1507.06527`.

[HZS20] S. S. Hegde, J. Zhang, and D. Suter. "Efficient Quantum Gates for Individual Nuclear Spin Qubits by Indirect Control". In: *Phys. Rev. Lett.* 124 (22 June 2020), p. 220501.

[Hoc+14] D. Hocker et al. "Characterization of control noise effects in optimal quantum unitary dynamics". In: *Phys. Rev. A* 90 (6 Dec. 2014), p. 062309.

[Hol+20] E. T. Holland et al. "Optimal control for the quantum simulation of nuclear dynamics". In: *Phys. Rev. A* 101 (6 June 2020), p. 062307.

[Hou+12] S. C. Hou et al. "Optimal Lyapunov-based quantum control for quantum systems". In: *Phys. Rev. A* 86 (2 Aug. 2012), p. 022321.

[How98] R. Howard. "The Gronwall Inequality". In: *Lecture Notes* (1998).

[HJM19] T. A. Howell, B. E. Jackson, and Z. Manchester. "ALTRO: A fast solver for constrained trajectory optimization". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2019, pp. 7674–7679.

[HKP20] H.-Y. Huang, R. Kueng, and J. Preskill. "Predicting many properties of a quantum system from very few measurements". In: *Nature Physics* 16.10 (2020), pp. 1050–1057.

[Hua+22a] H.-Y. Huang et al. "Learning many-body Hamiltonians with Heisenberg-limited scaling". In: *arXiv preprint arXiv:2210.03030* (2022).

[Hua+22b] H.-Y. Huang et al. "Quantum advantage in learning from experiments". In: *Science* 376.6598 (2022), pp. 1182–1186.

[HN08] W. Huyer and A. Neumaier. "SNOBFIT - Stable Noisy Optimization by Branch and Fit". In: *ACM Trans. Math. Softw.* 35.2 (July 2008).

[Jam72] A. Jamiolkowski. "Linear transformations which preserve trace and positive semidefiniteness of operators". In: *Reports on Mathematical Physics* 3.4 (1972), pp. 275–278.

[Jan+19] M. Janner et al. "When to Trust Your Model: Model-Based Policy Optimization". In: *Advances in Neural Information Processing Systems.* Vol. 32. Curran Associates, Inc., 2019.

[Jin+13] J. Jing et al. “Inverse engineering control in open quantum systems”. In: *Phys. Rev. A* 88 (5 Nov. 2013), p. 053422.

[JSL17] E. Jonckheere, S. Schirmer, and F. Langbein. “Structured singular value analysis for spintronics network information transfer control”. In: *IEEE Trans. Automatic Control* 62.12 (Dec. 2017), pp. 6568–6574. arXiv: 1706.03247.

[JSR14] E. Jonckheere, A. Shabani, and A. Rezakhani. “Indirect control invariance of Decoherence Splitting Manifold”. In: *IEEE Conf. Decision and Control*. Dec. 2014, pp. 5794–5801.

[JLS14] E. Jonckheere, F. C. Langbein, and S. Schirmer. “Quantum networks: anti-core of spin chains”. In: *Quantum Information Processing* 13.7 (2014), pp. 1607–1637.

[JSL18] E. Jonckheere, S. Schirmer, and F. Langbein. “Jonckheere-Terpstra test for nonclassical error versus log-sensitivity relationship of quantum spin network controllers”. In: *Int. J. Robust and Nonlinear Control* 28.6 (2018), pp. 2383–2403.

[Jon+09] J. A. Jones et al. “Magnetic Field Sensing Beyond the Standard Quantum Limit Using 10-Spin NOON States”. In: *Science* 324.5931 (2009), pp. 1166–1168.

[JR92] R. S. Judson and H. Rabitz. “Teaching lasers to control molecules”. In: *Phys. Rev. Lett.* 68 (10 Mar. 1992), pp. 1500–1503.

[Kab+14] C. Kabytayev et al. “Robustness of composite pulses to time-dependent control noise”. In: *Phys. Rev. A* 90 (1 July 2014), p. 012316.

[KL02] S. Kakade and J. Langford. “Approximately optimal approximate reinforcement learning”. In: *Proceedings of the Nineteenth International Conference on Machine Learning*. 2002, pp. 267–274.

[Kau+19] R. Kaubruegger et al. “Variational Spin-Squeezing Algorithms on Programmable Quantum Sensors”. In: *Phys. Rev. Lett.* 123 (26 Dec. 2019), p. 260505.

[Kel+14a] J. Kelly et al. “Optimal Quantum Control Using Randomized Benchmarking”. In: *Phys. Rev. Lett.* 112 (24 June 2014), p. 240504.

[Kel+14b] J. Kelly et al. “Optimal Quantum Control Using Randomized Benchmarking”. In: *Phys. Rev. Lett.* 112 (24 June 2014), p. 240504.

[Kel+14c] J. Kelly et al. “Optimal Quantum Control Using Randomized Benchmarking”. In: *Phys. Rev. Lett.* 112 (24 June 2014), p. 240504.

[Ken62] M. G. Kendall. *Rank Correlation Methods: 3d Ed.* C. Griffin, 1962.

[Kha+21] I. Khalid et al. “Reinforcement learning vs. gradient-based optimisation for robust energy landscape control of spin-1/2 quantum networks”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 4133–4139.

[Kha22] I. Khalid. 2022. URL: https://github.com/erg0dic/transmon_public.

[Kha+23a] I. Khalid et al. *Sample-efficient Model-based Reinforcement Learning for Quantum Control*. 2023. arXiv: 2304.09718.

[Kha+23b] I. Khalid et al. "Statistically characterizing robustness and fidelity of quantum controls and quantum control algorithms". In: *Phys. Rev. A* 107 (3 2023), p. 032606.

[Kha+05a] N. Khaneja et al. "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms". In: *Journal of magnetic resonance* 172.2 (2005), pp. 296–305.

[Kha+05b] N. Khaneja et al. "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms". In: *J. Magn. Res.* 172.2 (2005), pp. 296–305.

[KV09] K. Khodjasteh and L. Viola. "Dynamically Error-Corrected Gates for Universal Quantum Computation". In: *Phys. Rev. Lett.* 102 (8 2009), p. 080501.

[Kid22] P. Kidger. "On neural differential equations". PhD thesis. University of Oxford, 2022.

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[KB17] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980.

[KW19] D. P. Kingma and M. Welling. "An introduction to variational autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[Kni+08] E. Knill et al. "Randomized benchmarking of quantum gates". In: *Phys. Rev. A* 77 (1 Jan. 2008), p. 012307.

[Koc+22] C. P. Koch et al. *Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe*. 2022.

[KBC21] A. Koswara, V. Bhutoria, and R. Chakrabarti. "Robust control of quantum dynamics under input and parameter uncertainty". In: *Phys. Rev. A* 104 (5 Nov. 2021), p. 053118.

[Kra+19] P. Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied Physics Reviews* 6.2 (2019), p. 021318. DOI: 10.1063/1.5089550.

[Kud+22] M. Kudra et al. "Robust Preparation of Wigner-Negative States with Optimized SNAP-Displacement Sequences". In: *PRX Quantum* 3 (3 July 2022), p. 030301.

[KWM00] T. D. Kühner, S. R. White, and H. Monien. “One-dimensional Bose-Hubbard model with nearest-neighbor interaction”. In: *Phys. Rev. B* 61 (18 May 2000), pp. 12474–12489.

[Kuk+89] J. R. Kuklinski et al. “Adiabatic population transfer in a three-level system driven by delayed laser pulses”. In: *Phys. Rev. A* 40 (11 Dec. 1989), pp. 6741–6744.

[KR09] I. Kuprov and C. T. Rodgers. “Derivatives of spin dynamics simulations”. In: *The Journal of Chemical Physics* 131.23 (2009). DOI: 10.1063/1.3267086.

[LSJ15a] F. C. Langbein, S. Schirmer, and E. Jonckheere. “Time optimal information transfer in spintronics networks”. In: *IEEE Conf. Decision and Control* (2015), pp. 6454–6459.

[LSJ15b] F. C. Langbein, S. Schirmer, and E. Jonckheere. “Time optimal information transfer in spintronics networks”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 6454–6459.

[LKD02] H. Lee, P. Kok, and J. P. Dowling. “A quantum Rosetta stone for interferometry”. In: *Journal of Modern Optics* 49.14-15 (2002), pp. 2325–2338.

[Lee+18] J.-S. Lee et al. “Quantum plasmonic sensing using single photons”. In: *Opt. Express* 26.22 (2018), pp. 29272–29282.

[Len+22] Y. L. Len et al. “Quantum metrology with imperfect measurements”. In: *Nature Communications* 13.1 (2022), p. 6971.

[Leu+17] N. Leung et al. “Speedup for quantum optimal control from automatic differentiation based on graphics processing units”. In: *Phys. Rev. A* 95 (4 Apr. 2017), p. 042318.

[LK09] J.-S. Li and N. Khaneja. “Ensemble control of Bloch equations”. In: *IEEE Transactions on Automatic Control* 54.3 (2009), pp. 528–536.

[LB17] Y. Li and S. C. Benjamin. “Efficient Variational Quantum Simulator Incorporating Active Error Minimization”. In: *Phys. Rev. X* 7 (2 June 2017), p. 021050.

[Lic16] A. Lichnerowicz. *Elements of tensor calculus*. Courier Dover Publications, 2016.

[LCW98] D. A. Lidar, I. L. Chuang, and K. B. Whaley. “Decoherence-Free Subspaces for Quantum Computation”. In: *Phys. Rev. Lett.* 81 (12 Sept. 1998), pp. 2594–2597.

[LSM61] E. Lieb, T. Schultz, and D. Mattis. “Two soluble models of an antiferromagnetic chain”. In: *Annals of Physics* 16.3 (1961), pp. 407–466.

[Lil+15] T. P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015.

[Loh96] W.-L. Loh. "On Latin hypercube sampling". In: *Annals of Statistics* 24.5 (1996), pp. 2058–2080.

[Lov+13] N. B. Lovett et al. "Differential Evolution for Many-Particle Adaptive Quantum Metrology". In: *Phys. Rev. Lett.* 110 (22 May 2013), p. 220501.

[Lud+15] A. D. Ludlow et al. "Optical atomic clocks". In: *Rev. Mod. Phys.* 87 (2 June 2015), pp. 637–701.

[Lyk+22] D. Lykov et al. *Sampling Frequency Thresholds for Quantum Advantage of Quantum Approximate Optimization Algorithm.* 2022. arXiv: 2206.03579 [quant-ph].

[Mac+11a] S. Machnes et al. "Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework". In: *Phys. Rev. A* 84 (2 Aug. 2011), p. 022305.

[Mac+11b] S. Machnes et al. "Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework". In: *Phys. Rev. A* 84 (2 Aug. 2011), p. 022305.

[Mac+18] S. Machnes et al. "Tunable, Flexible, and Efficient Optimization of Control Pulses for Practical Qubits". In: *Phys. Rev. Lett.* 120 (15 Apr. 2018), p. 150401.

[MGE11] E. Magesan, J. M. Gambetta, and J. Emerson. "Scalable and Robust Randomized Benchmarking of Quantum Processes". In: *Phys. Rev. Lett.* 106 (18 May 2011), p. 180504. DOI: 10.1103/PhysRevLett.106.180504.

[MG20] E. Magesan and J. M. Gambetta. "Effective Hamiltonian models of the cross-resonance gate". In: *Phys. Rev. A* 101 (5 May 2020), p. 052308.

[Man+23] D. J. Mankowitz et al. "Faster sorting algorithms discovered using deep reinforcement learning". In: *Nature* 618.7964 (2023), pp. 257–263.

[Moe+23] T. M. Moerland et al. "Model-based reinforcement learning: A survey". In: *Foundations and Trends® in Machine Learning* 16.1 (2023), pp. 1–118.

[MRT18] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning.* MIT press, 2018.

[MR12] K. W. Moore and H. Rabitz. "Exploring constrained quantum control landscapes". In: *The Journal of chemical physics* 137.13 (2012), p. 134113.

[Mue+22] M. Mueller et al. "One decade of quantum optimal control in the chopped random basis". In: *Reports on Progress in Physics* (2022).

[Muk+20] R. Mukherjee et al. "Preparation of ordered states in ultra-cold gases using bayesian optimization". In: *New Journal of Physics* 22.7 (2020), p. 075001.

[NM65] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313.

[NP20] G. Neu and C. Pike-Burke. “A unifying view of optimism in episodic reinforcement learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1392–1403.

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[Niu+19a] M. Y. Niu et al. “Universal quantum control through deep reinforcement learning”. In: *npj Quantum Information* 5.1 (2019), p. 33.

[Niu+19b] M. Y. Niu et al. “Universal quantum control through deep reinforcement learning”. In: *npj Quantum Information* 5.1 (2019), pp. 1–8.

[Niu+19c] M. Y. Niu et al. “Universal quantum control through deep reinforcement learning”. In: *npj Quantum Information* 5.1 (2019), pp. 1–8.

[NY98] J. Nocedal and Y. Yuan. “Combining trust region and line search techniques”. In: *Advances in Nonlinear Programming.* 1998, pp. 153–175.

[ONe+22a] S. O’Neil et al. *Time Domain Sensitivity of the Tracking Error.* 2022. eprint: `2210.15783`.

[ONe+22b] S. O’Neil et al. *Time Domain Sensitivity of the Tracking Error.* 2022. arXiv: `2210.15783`.

[ONe+23] S. P. O’Neil et al. “Analyzing and Unifying Robustness Measures for Excitation Transfer Control in Spin Networks”. In: *IEEE Control Systems Letters* 7 (2023), pp. 1783–1788.

[Owe19] A. B. Owen. *Monte Carlo Book: the Quasi-Monte Carlo parts.* 2019.

[Par+16] D. K. Park et al. “Randomized benchmarking of quantum gates implemented by electron spin resonance”. In: *Journal of Magnetic Resonance* 267 (2016), pp. 68–78.

[Pas+19] A. Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[Per+13] A. Perez-Leija et al. “Coherent quantum transport in photonic lattices”. In: *Phys. Rev. A* 87 (1 Jan. 2013), p. 012309.

[Per+14] A. Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature communications* 5.1 (2014), p. 4213.

[Pez+18] L. Pezzè et al. “Quantum metrology with nonclassical states of atomic ensembles”. In: *Rev. Mod. Phys.* 90 (3 2018), p. 035005.

[Pir+20] S. Pirandola et al. “Advances in quantum cryptography”. In: *Adv. Opt. Photon.* 12.4 (2020), pp. 1012–1236.

[Pog+21] I. Pogorelov et al. “Compact Ion-Trap Quantum Computing Demonstrator”. In: *PRX Quantum* 2 (2 June 2021), p. 020343.

[Pok+18] B. Pokharel et al. “Demonstration of Fidelity Improvement Using Dynamical Decoupling with Superconducting Qubits”. In: *Phys. Rev. Lett.* 121 (22 2018), p. 220502.

[Pon87] L. S. Pontryagin. *Mathematical theory of optimal processes.* CRC press, 1987.

[PPM22] R. Porotti, V. Peano, and F. Marquardt. *Gradient Ascent Pulse Engineering with Feedback.* 2022.

[Por+19] R. Porotti et al. “Coherent transport of quantum states by deep reinforcement learning”. In: *Communications Physics* 2.1 (2019), p. 61.

[Pre18] J. Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.

[PCM22] F. Preti, T. Calarco, and F. Motzoi. “Continuous Quantum Gate Sets and Pulse-Class Meta-Optimization”. In: *PRX Quantum* 3 (4 2022), p. 040311.

[Pro+17] T. Proctor et al. “What Randomized Benchmarking Actually Measures”. In: *Phys. Rev. Lett.* 119 (13 Sept. 2017), p. 130502. DOI: `10.1103/PhysRevLett.119.130502`.

[Pro+22] T. Propson et al. “Robust Quantum Optimal Control with Trajectory Optimization”. In: *Phys. Rev. Applied* 17 (1 Jan. 2022), p. 014036. DOI: `10.1103/PhysRevApplied.17.014036`.

[Rab+06] H. Rabitz et al. “Topology of optimally controlled quantum mechanical transition probability landscapes”. In: *Phys. Rev. A* 74 (1 July 2006), p. 012721.

[RHR05] H. Rabitz, M. Hsieh, and C. Rosenthal. “Landscape for optimal control of quantum-mechanical unitary transformations”. In: *Phys. Rev. A* 72 (5 Nov. 2005), p. 052337.

[Raf+05] E. J. Rafols et al. “Using Predictive Representations to Improve Generalization in Reinforcement Learning.” In: *IJCAI.* 2005, pp. 835–840.

[RTC17] A. Ramdas, N. G. Trillos, and M. Cuturi. “On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests”. In: *Entropy* 19.2 (2017).

[RNK12] D. M. Reich, M. Ndong, and C. P. Koch. “Monotonically convergent optimization in quantum control using Krotov’s method”. In: *The Journal of Chemical Physics* 136.10 (2012), p. 104103.

[Rip07] B. D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, 2007.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (1978), pp. 120–126.

[RR96] R. Rojas and R. Rojas. “The backpropagation algorithm”. In: Springer, 1996, pp. 149–182.

[RBR16] P. Rooney, A. M. Bloch, and C. Rangan. “Flag-based control of quantum purity for $n = 2$ systems”. In: *Phys. Rev. A* 93 (6 June 2016), p. 063424.

[RRZ88] T. S. Rose, M. J. Rosker, and A. H. Zewail. “Femtosecond real-time observation of wave packet oscillations (resonance) in dissociation reactions”. In: *The Journal of Chemical Physics* 88.10 (1988), pp. 6672–6673.

[Rus+12] A. Ruschhaupt et al. “Optimally robust shortcuts to population inversion in two-level quantum systems”. In: *New Journal of Physics* 14.9 (Sept. 2012), p. 093040.

[SLH81] M. G. Safonov, A. J. Laub, and G. L. Hartmann. “Feedback properties of multivariable systems: The role and use of the return difference matrix”. In: *IEEE Trans. Automatic Control* AC-26.1 (Feb. 1981), pp. 47–65.

[Saf12] M. G. Safonov. “Origins of robust control: Early history and future speculations”. In: *Annual Reviews in Control* 36.2 (2012), pp. 173–181.

[SM20] F. Sauvage and F. Mintert. “Optimal Quantum Control with Poor Statistics”. In: *PRX Quantum* 1 (2 Dec. 2020), p. 020322.

[SM22] F. Sauvage and F. Mintert. “Optimal Control of Families of Quantum Gates”. In: *Phys. Rev. Lett.* 129 (5 July 2022), p. 050507.

[Sch+20] F. Schäfer et al. “A differentiable programming method for quantum control”. In: *Machine Learning: Science and Technology* 1.3 (Aug. 2020), p. 035009.

[Sch+21a] F. Schäfer et al. “Control of stochastic quantum dynamics by differentiable programming”. In: *Machine Learning: Science and Technology* 2.3 (Apr. 2021), p. 035004.

[SFS01] S. G. Schirmer, H. Fu, and A. I. Solomon. “Complete controllability of quantum systems”. In: *Phys. Rev. A* 63 (6 May 2001), p. 063410.

[Sch+21b] S. G. Schirmer et al. “Robustness of Quantum Systems Subject to Decoherence: Structured Singular Value Analysis?” In: *IEEE Conf. Decision and Control.* 2021, pp. 4158–4163.

[SJL17] S. G. Schirmer, E. A. Jonckheere, and F. C. Langbein. “Design of feedback control laws for information transfer in spintronics networks”. In: *IEEE Transactions on Automatic Control* 63.8 (2017), pp. 2523–2536.

[SJL18] S. G. Schirmer, E. A. Jonckheere, and F. C. Langbein. “Design of Feedback Control Laws for Information Transfer in Spintronics Networks”. In: *IEEE Trans. Automatic Control* 63.8 (2018), pp. 2523–2536. DOI: 10.1109/TAC.2017.2777187.

[Sch+22a] S. G. Schirmer et al. “Robust Control Performance for Open Quantum Systems”. In: *IEEE Trans. Automatic Control* (2022). in press. arXiv: 2008.13691.

[Sch+22b] S. G. Schirmer et al. “Robust Control Performance for Open Quantum Systems”. In: *IEEE Transactions on Automatic Control* 67.11 (2022), pp. 6012–6024.

[Sch+15] J. Schulman et al. “Trust region policy optimization”. In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.

[Sch+17] J. Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347.

[Sem+21] G. Semeghini et al. “Probing topological spin liquids on a programmable quantum simulator”. In: *Science* 374.6572 (2021), pp. 1242–1247.

[She+16] S. Sheldon et al. “Procedure for systematically tuning up cross-talk in the cross-resonance gate”. In: *Phys. Rev. A* 93 (6 June 2016), p. 060302. DOI: 10.1103/PhysRevA.93.060302. URL: https://link.aps.org/doi/10.1103/PhysRevA.93.060302.

[SHR06] Z. Shen, M. Hsieh, and H. Rabitz. “Quantum optimal control: Hessian analysis of the control landscape”. In: *J. Chem. Phys.* 124.20 (2006), p. 204106. DOI: 10.1063/1.2198836.

[Shi+21] H.-J. M. Shi et al. *A Noise-Tolerant Quasi-Newton Algorithm for Unconstrained Optimization*. 2021. arXiv: 2010.04352.

[Sho94] P. W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[SP00] P. W. Shor and J. Preskill. “Simple Proof of Security of the BB84 Quantum Key Distribution Protocol”. In: *Phys. Rev. Lett.* 85 (2 July 2000), pp. 441–444.

[Sil+14] D. Silver et al. “Deterministic policy gradient algorithms”. In: *International conference on machine learning*. Pmlr. 2014, pp. 387–395.

[Sil+18] D. Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419 (2018), pp. 1140–1144.

[Siv+22a] V. V. Sivak et al. “Model-Free Quantum Control with Reinforcement Learning”. In: *Phys. Rev. X* 12 (1 Mar. 2022), p. 011059.

[Siv+22b] V. V. Sivak et al. “Model-Free Quantum Control with Reinforcement Learning”. In: *Phys. Rev. X* 12 (1 Mar. 2022), p. 011059.

[SZZ04] P. Solinas, P. Zanardi, and N. Zanghi. “Robustness of non-Abelian holonomic quantum gates against parametric noise”. In: *Phys. Rev. A* 70 (4 Oct. 2004), p. 042316.

[SM99] A. Sørensen and K. Mølmer. "Quantum Computation with Ions in Thermal Motion". In: *Phys. Rev. Lett.* 82 (9 Mar. 1999), pp. 1971–1974.

[SM00] A. Sørensen and K. Mølmer. "Entanglement and quantum computation with ions in thermal motion". In: *Phys. Rev. A* 62 (2 2000), p. 022311.

[SHW87] F. Spano, M. Haner, and W. Warren. "Spectroscopic demonstration of picosecond, phase-shifted laser multiple-pulse sequences". In: *Chemical physics letters* 135.1-2 (1987), pp. 97–102.

[Spe04] C. Spearman. "The Proof and Measurement of Association between Two Things". In: *American J. Psychology* 15 (1904), pp. 72–101.

[Sri+21] R. Srinivas et al. "High-fidelity laser-free universal control of trapped ion qubits". In: *Nature* 597.7875 (2021), pp. 209–213.

[Ste96a] A. M. Steane. "Error Correcting Codes in Quantum Theory". In: *Phys. Rev. Lett.* 77 (5 July 1996), pp. 793–797.

[Ste96b] A. M. Steane. "Multiple-particle interference and quantum error correction". In: *Proc. Royal Soc. London. Series A: Mathematical, Physical and Engineering Sciences* 452 (1954 1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136.

[SS08] M. Stefanovic and M. G. Safonov. "Safe adaptive switching control: Stability and convergence". In: *IEEE Transactions on Automatic Control* 53.9 (2008), pp. 2012–2021.

[Ste87] M. Stein. "Large Sample Properties of Simulations Using Latin Hypercube Sampling". In: *Technometrics* 29.2 (1987), pp. 143–151.

[SM03] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis.* Cambridge university press, 2003.

[SB18a] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 2018.

[Sut88] R. S. Sutton. "Learning to predict by the methods of temporal differences". In: *Machine learning* 3 (1988), pp. 9–44.

[Sut91] R. S. Sutton. "Dyna, an integrated architecture for learning, planning, and reacting". In: *ACM Sigart Bulletin* 2.4 (1991), pp. 160–163.

[SB18b] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[TR85] D. J. Tannor and S. A. Rice. "Control of selectivity of chemical reaction via control of wave packet evolution". In: *The Journal of chemical physics* 83.10 (1985), pp. 5013–5018.

[TB16] M. A. Taylor and W. P. Bowen. "Quantum metrology and its application in biology". In: *Physics Reports* 615 (2016), pp. 1–59.

[TBG17] K. Temme, S. Bravyi, and J. M. Gambetta. "Error Mitigation for Short-Depth Quantum Circuits". In: *Phys. Rev. Lett.* 119 (18 Nov. 2017), p. 180509.

[Tho14] P. Thomas. “Bias in natural actor-critic algorithms”. In: *Int. Conf. Machine Learning.* 2014, pp. 441–448.

[Tur19] G. Turinici. “Stochastic learning control of inhomogeneous quantum ensembles”. In: *Phys. Rev. A* 100 (5 Nov. 2019), p. 053403.

[UO30] G. E. Uhlenbeck and L. S. Ornstein. “On the Theory of the Brownian Motion”. In: *Phys. Rev.* 36 (5 Sept. 1930), pp. 823–841.

[Uhl00] A. Uhlmann. “Fidelity and concurrence of conjugated states”. In: *Phys. Rev. A* 62 (3 Aug. 2000), p. 032307. DOI: `10.1103/PhysRevA.62.032307`.

[Val+19] A. Valenti et al. “Hamiltonian learning for quantum error correction”. In: *Phys. Rev. Res.* 1 (3 Nov. 2019), p. 033092.

[VHA19] H. P. Van Hasselt, M. Hessel, and J. Aslanides. “When to use parametric models in reinforcement learning?” In: *Advances in Neural Information Processing Systems* 32 (2019).

[Vil09] C. Villani. *Optimal transport: old and new.* Vol. 338. Springer, 2009.

[VZ12] L. Vinet and A. Zhedanov. “Almost perfect state transfer in quantum spin chains”. In: *Phys. Rev. A* 86 (5 Nov. 2012), p. 052319.

[VKL99] L. Viola, E. Knill, and S. Lloyd. “Dynamical Decoupling of Open Quantum Systems”. In: *Phys. Rev. Lett.* 82 (12 Mar. 1999), pp. 2417–2421.

[Vir+20] P. Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3 (2020), pp. 261–272.

[Vit+17] N. V. Vitanov et al. “Stimulated Raman adiabatic passage in physics, chemistry, and beyond”. In: *Rev. Mod. Phys.* 89 (1 Mar. 2017), p. 015006.

[Wal+15] J. Wallman et al. “Estimating the coherence of noise”. In: *New Journal of Physics* 17.11 (2015), p. 113020.

[WBE16] J. J. Wallman, M. Barnhill, and J. Emerson. “Robust characterization of leakage errors”. In: *New Journal of Physics* 18.4 (2016), p. 043021.

[WF14] J. J. Wallman and S. T. Flammia. “Randomized benchmarking with confidence”. In: *New Journal of Physics* 16.10 (2014), p. 103032.

[Wan+19] S. Wang et al. “Beating the Fundamental Rate-Distance Limit in a Proof-of-Principle Quantum Key Distribution System”. In: *Phys. Rev. X* 9 (2 June 2019), p. 021046.

[WAU20] Z. T. Wang, Y. Ashida, and M. Ueda. “Deep Reinforcement Learning Control of Quantum Cartpoles”. In: *Phys. Rev. Lett.* 125 (10 Sept. 2020), p. 100401.

[Wan+16] Z. Wang et al. “Bayesian optimization in a billion dimensions via random embeddings”. In: *Journal of Artificial Intelligence Research* 55 (2016), pp. 361–387.

[WD92] C. J. Watkins and P. Dayan. “Q-learning”. In: *Machine learning* 8 (1992), pp. 279–292.

[Wat09] J. Watrous. “Semidefinite programs for completely bounded norms”. In: *arXiv preprint arXiv:0901.4709* (2009).

[WA18] C. A. Weidner and D. Z. Anderson. “Experimental Demonstration of Shaken-Lattice Interferometry”. In: *Phys. Rev. Lett.* 120 (26 June 2018), p. 263201.

[Wei+22] C. A. Weidner et al. “Applying classical control techniques to quantum systems: entanglement versus stability margin and other limitations”. In: *2022 IEEE 61st Conference on Decision and Control (CDC).* 2022, pp. 5813–5818.

[Wei+86] J. Weiner et al. “Nonlinear spectroscopy of InGaAs/InAlAs multiple quantum well structures”. In: *Applied physics letters* 49.9 (1986), pp. 531–533.

[Wit+21] N. Wittler et al. “Integrated Tool Set for Control, Calibration, and Characterization of Quantum Devices Applied to Superconducting Qubits”. In: *Physical Review Applied* 15.3 (Mar. 2021).

[WBC11] C. J. Wood, J. D. Biamonte, and D. G. Cory. “Tensor networks and graphical calculus for open quantum systems”. In: *arXiv preprint arXiv:1111.6950* (2011).

[Wu+19a] R.-B. Wu et al. “Learning robust and high-precision quantum controls”. In: *Physical Review A* 99.4 (2019), p. 042327.

[Wu+19b] R.-B. Wu et al. “Learning robust and high-precision quantum controls”. In: *Phys. Rev. A* 99 (4 Apr. 2019), p. 042327.

[Xia+15] T. Xia et al. “Randomized Benchmarking of Single-Qubit Gates in a 2D Array of Neutral-Atom Qubits”. In: *Phys. Rev. Lett.* 114 (10 Mar. 2015), p. 100503. DOI: 10.1103/PhysRevLett.114.100503.

[Yan+19] H. Yan et al. “On robustness of neural ordinary differential equations”. In: *arXiv preprint arXiv:1910.05513* (2019).

[Yan+20] X.-d. Yang et al. “Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses”. In: *Phys. Rev. A* 102 (6 Dec. 2020), p. 062605.

[You10] L. Younes. *Shapes and Diffeomorphisms.* Vol. 171. Springer, 2010.

[ZSS14] E. Zahedinejad, S. Schirmer, and B. C. Sanders. “Evolutionary algorithms for hard quantum control”. In: *Phys. Rev. A* 90 (3 Sept. 2014), p. 032310.

[Zam63] G. Zames. “Functional Analysis Applied to Nonlinear Feedback Systems”. In: *IEEE Transactions on Circuit Theory* 10.3 (1963), pp. 392–404. DOI: 10.1109/TCT.1963.1082162.

[Zew88] A. H. Zewail. “Laser femtochemistry”. In: *Science* 242.4886 (1988), pp. 1645–1653.

[Zha+17] J. Zhang et al. “Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator”. In: *Nature* 551.7682 (2017), pp. 601–604.

[Zha+05] J. Zhang et al. “Simulation of Heisenberg XY interactions and realization of a perfect state transfer in spin chains using liquid nuclear magnetic resonance”. In: *Phys. Rev. A* 72 (1 July 2005), p. 012331.

[ZD98] K. Zhou and J. C. Doyle. *Essentials of robust control.* Vol. 104. Upper Saddle River, NJ: Prentice Hall, 1998.

[Zhu+97] C. Zhu et al. “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization”. In: *ACM Trans. Math. Softw.* 23.4 (1997), pp. 550–560.

[Zie+08] B. D. Ziebart et al. “Maximum entropy inverse reinforcement learning.” In: *Aaai.* Vol. 8. Chicago, IL, USA. 2008, pp. 1433–1438.