UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Characterization and Control of Quantum Systems using Machine Learning and Information Theory

by **Akram Youssry Abdelaziz Mohamed**

Thesis submitted in fulfillment of the requirements of the degree of

Doctor of Philosophy (Engineering)

under the supervision of

A/Prof. Dr. Marco Tomamichel

A/Prof. Dr. Christopher Ferrie

A/Prof. Dr. Min-Hsiu Hsieh

University of Technology Sydney
Faculty of Engineering and Information Technology

November, 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Akram Youssry Abdelaziz Mohamed declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 09/02/2021

Acknowledgments

I would like to start by praising and thanking God for His unlimited blessings on me. I would like to express my gratitude to Dr. Marco Tomamichel my principal supervisor for opening the door for me to travel to Australia, join his group at UTS, and get to interact with all the researchers there. I also thank him for trusting me being the first Egyptian, Middle Eastern, and African student to join the group, and for supporting me from day one. Next, I would like to express my gratitude to my co-supervisor Dr. Christopher Ferrie, who influenced my research interests substantially. I would like to thank him for giving me the opportunity to join an AUSMURI collaborative project as a part-time research student and get to interact with many researchers across Australia. I acknowledge their continuous support and I have learnt a lot from them. I thank them for supporting my academic visits to other research groups, resulting in a multiple of successful collaborations. I would like to acknowledge my co-supervisor Dr. Min-Hsiu Hsieh for his support and guidance. I would like to thank my collaborators Dr. Alberto Peruzzo at RMIT, Dr. Robert Chapman at University of Innsbruck, and Dr. Gerardo Paz-Silva at Griffith University. I would like to acknowledge the support I got from UTS to pursue my research including the iHPC facility for executing the intensive computation needed in my research. I also acknowledge the Australian Research Training Program scholarship. I acknowledge all my friends and colleagues whom I enjoyed interacting with throughout my journey. I would like to mention particularly Maria Quadeer, Madhav Krishnan Vijayan, Arinta Auza, Elija Perrier, Mauricio Morales, Lirandë Pira, and Bin Cheng. I would like to thank my professors at Ain Shams University who taught me during my undergraduate and postgraduate studies communication systems engineering and fundamentals of quantum information science, as well as how to do research. Finally, I would like to acknowledge my parents for their continuous support throughout my life, and particularly during my travel. I thank my mother for the great sacrifices she made to make it possible for me to travel. I thank my father for assisting me logistically before and during my travel.

This is a THESIS BY COMPILATION. Parts of this thesis have been already published. The content have been edited to suit the formatting of the thesis and to maintain its coherence.

DECLARATION OF PUBLICATIONS INCLUDED IN THE THESIS

Chapter	:	3
Name	:	Efficient online quantum state estimation using a matrix-exponentiated gradient method
Authors	:	Akram Youssry, Christopher Ferrie, Marco Tomamichel
Contributions	:	AY implemented and tested the proposed algorithm. AY did the convergence proof with contributions from CF and MT. All the authors wrote and modified the paper. CF and MT supervised the project.
Status	:	Published in "New Journal of Physics"
Address	:	`https://doi.org/10.1088/1367-2630/ab0438`
Signatures	:	Akram Youssry
		Christopher Ferrie
		Marco Tomamichel

Chapter	:	4
Name	:	Modeling and control of a reconfigurable photonic circuit using deep learning
Authors	:	Akram Youssry, Robert J Chapman, Alberto Peruzzo, Christopher Ferrie, Marco Tomamichel
Contributions	:	RC and AP designed the photonic chip. RC developed the simulator for the chip. AY modified the simulator to include distortion effects. AY developed, implemented, and tested the proposed algorithm. AY wrote the paper, with contributions from RC. AP, CF, and MT supervised the project.
Status	:	Published in"Quantum Science and Technology"
Address	:	`https://doi.org/10.1088/2058-9565/ab60de`
Signatures	:	Akram Youssry
		Robert J Chapman
		Alberto Peruzzo
		Christopher Ferrie
		Marco Tomamichel

Chapter	:	5
Name	:	Characterization and control of open quantum systems beyond quantum noise spectroscopy
Authors	:	Akram Youssry, Gerardo A Paz-Silva, Christopher Ferrie
Contributions	:	AY designed the proposed architecture with feedback from all authors. AY implemented, trained, and tested the algorithm. GPS and CF equivalently supervised the project. All the authors wrote and modified the paper.
Status	:	Published in "npj Quantum Information"
Address	:	`https://doi.org/10.1038/s41534-020-00332-8`
Signatures	:	Akram Youssry
		Gerardo A Paz-Silva
		Christopher Ferrie

Contents

List of Figures

List of Tables

List of Abbreviations

CAD	Computer-Aided Design
CDD	Concatenated Dynamical Decoupling
CPMG	Carr–Purcell–Meiboom–Gill
CPTP	Completely-Positive Trace-Preserving
CRAB	Chopped Random Basis
DD	Dynamical Decoupling
GRAPE	Gradient Ascent Pulse Engineering
GRU	Gated-Recurrent Unit
GST	Gate Set Tomography
LS	Least-squares Estimation
LSTM	Long Short-Term Memory
MEG	Matrix Exponentiated Gradient
MLE	Maximum Likelihood Estimation
NISQ	Noisy Intermediate-Scale Quantum
NN	Neural Network
OC	Optimal Control
PDD	Periodic Dynamical Decoupling
PGD	Projected Gradient Descent
PSD	Power Spectral Density
QCVV	Quantum Characterization, Verification, and Verification

QIP Quantum Information Processing
QNS Quantum Noise Spectroscopy
QPT Quantum Process Tomography
QST Quantum State Tomography

RNN Recurrent Neural Network
RWA Reconfigurable Waveguide Array

SPAM State Preparation and Measurement

UDD Uhrig Dynamical Decoupling

Abstract

The tasks of characterization and control of quantum systems are becoming more challenging with the advancement of quantum technology. Standard methods that were successful for simple quantum systems are becoming inadequate for more complex engineered systems. Modelling assumptions and approximations (such as Markovianity) are not justifiable anymore. As a result, the usual models fail to fit experimental measurements. In this thesis, we use state-of-the-art machine learning methods, assisted by tools from information theory as needed, to develop new frameworks that try to address these challenges. We focus on three directions. The first is developing an efficient online quantum state estimation algorithm with provable convergence properties. The second is developing a deep learning framework for characterizing and controlling closed quantum systems. The final direction is upgrading that framework to be suitable for characterization and control of open quantum systems. This thesis opens the door for a novel way of utilizing machine learning techniques for applications in quantum information specially and physics in general.

Chapter 1

Introduction

1.1 Topic

Recently, there have been many developments in the field of quantum information processing (QIP) on experimental as well as theoretical aspects. These developments have been motivated by the wide range of prospective applications of quantum computing, quantum cryptography, and quantum communications. The emergence of Noisy Intermediate-Scale Quantum (NISQ) devices [1] has been a major technological breakthrough. These devices are capable of performing computations on an intermediate number of qubits with limited performance. This makes it difficult to execute quantum algorithm such as Shor's integer factorization or Grover's search algorithm. However, there are many potential near-term applications such as quantum machine learning, quantum simulations and quantum chemistry [1]. There is also experimental and theoretical evidence of the capability of such devices to have computational advantages compared to classical computers [1–3].

There are many challenges in designing and operating NISQ devices. On the physical side, the qubits suffer from noise, spatial correlations (crosstalk) and temporal correlations (See [4] for an example). Modelling these effects theoretically is difficult. Thus, many assumptions and approximations are made which might not be sufficiently accurate. On the classical hardware side, there are many experimental constraints such as limited power and bandwidth of control signals that can be generated. On the classical software side, the nature of quantum information makes it difficult to store or simulate on a classical computer without exponential computational resources. Therefore, a standard verification task such as reconstructing a quantum state given a set of physically acquired measurements becomes impossible for a device with just a 100 qubits. Additionally, standard optimal control techniques become unsuitable because they require either having an exact mathematical description of the quantum system (which is difficult to obtain without assump-

tions), or alternatively simulating the system dynamics (which is computationally infeasible). These kinds of challenges need to be addressed in order to realize the full potential of NISQ devices. As a result, there is a need for developing novel methods that facilitate the tasks of engineering a quantum system.

This thesis targets two main categories of audience. The first category is machine learning engineers who would like to contribute to advancing quantum technology. The second category is quantum engineers and experimentalists who focus on building quantum computers. In this thesis we try to bridge the gap between the two communities. The thesis will cover the fundamentals of modelling, characterization, and control of quantum systems as well as multiple novel contributions of utilizing machine learning for these purposes.

1.2 Stakeholders and Aims

Besides academic experimental research groups, there are many industrial organizations that could benefit from this thesis. Examples include gigantic organizations working on quantum technology such as Microsoft, Google, IBM, Intel, and NASA; as well as other smaller organizations that are emerging in a fast-growing economy.

For such organizations there are many aims that they would like to fulfil. The first aim is to develop a functional large-scale fault-tolerant quantum computer. The laws of quantum physics put many constraints on the extent to which we can control quantum systems. Additionally, noise has far more impact on quantum systems than classical systems, and at the same time more difficult to model and mitigate.

The second aim is to facilitate the design and operation of quantum computers through the development of suitable Computer-Aided Design (CAD) tools. One of the reasons that classical technology has progressed enormously is the use of such tools, many of which are even automated (such as tools for designing digital integrated circuits). For quantum technology, the situation is more complicated as any system would involve classical and quantum components. Additionally, for classical technology, the existence of well-defined well-separated abstraction layers facilitate the design and verification stages. So far, this has not been the case with quantum technology, with difficulty in even fundamentally defining such abstraction layers. For example, designing control pulses to implement a quantum gate depends on the physical implementation of the qubits (lower abstraction level), and would change significantly if quantum error correction is used (higher abstraction level). At the same time, selecting a universal set of quantum gates into which any quantum algorithm is compiled, would depend on the physical implementation of the qubits. For example, it is easier to implement a CZ for some systems rather than a CNOT gate, and the other way around for other systems. Finally, it can be the case that

designing the control pulses depends on the order of the target gates required, due to non-Markovian effects in the system. In this case, designing the control pulses (low abstraction level) requires the full knowledge of the target algorithm (highest abstraction level). Therefore, a different set of CAD tools, backed up by new design methodologies, is required for progressing quantum technology.

The third aim is to minimize the time-to-market. There is a high competition between companies to lead the new industry. The first to get to the market will have the greatest share of profits. Fulfilling this aim requires among other things reducing the time taken for the verification process. Standard techniques such as quantum tomography can take up few days on a high-performance computer for a system of few qubits. In many cases, the procedure has to be repeated iteratively (for example tuning the system to some operating point or controlling the dynamics of the system). This means that the whole design process would consume a huge amount of time, resulting in considerable delays.

The fourth aim is to minimize the production cost of a quantum computer. Although qubits are the building blocks of a quantum computer, it is still mandatory to have classical hardware at least to be able for control. Theoretically, it is possible to enhance the performance of many quantum systems if we can generate arbitrary control signals such as a train of ideal impulses, or an extremely high frequency sinusoidal. Practically, this might not be possible, or would require special hardware equipment (such as high-performance arbitrary waveform generators, or high-quality cables, etc.) which would contribute significantly to the overall cost. So, it might be more profitable if such requirements are relaxed, so cheaper hardware could be used.

1.3 Objectives and Significance

This thesis targets three main objectives. The first objective is to develop an online quantum state estimation algorithm suitable for quantum tomography as well as quantum control applications. The algorithm must produce an estimate that is accurate compared to standard methods. Additionally, there must be a guarantee on the convergence of the algorithm. This is one of the drawbacks of many standard techniques; that is, the lack of proofs of convergence. Finally, since this algorithm is an online estimation algorithm, there will be an issue of choosing the learning rate (or the step size). This choice will be taken into account while proving convergence. Achieving this objective will result in an efficient verification scheme, reducing the design time, which is the third aim of the stakeholders. Additionally, the software implementation of the algorithm could be a part of a CAD for quantum systems which is the second aim of the stakeholders.

The second objective of this thesis, is to develop a novel framework suitable for modelling a closed quantum system in practical experimental situations (such as the existence of unknown deterministic distortions on control pulses, or uncertain terms in the Hamiltonian). This framework must be suitable for performing quantum control as well, taking into consideration further constraints such as limited-amplitude control pulses. Finally, the experimental procedure must be practical, not involving physical quantities that are impossible to measure such as parasitic voltages in a chip. This objective will directly contribute to fulfilling the first, second, and fourth aims of the stakeholders. Being able to model and control a quantum system facilitates the design of a quantum computer. Like the first objective, the software implementation can also be a part of a CAD system. Finally, taking into consideration experimental constraints of available hardware makes it possible to use cheaper equipment and compensating for the performance degradation on the software side. This directly reduces the overall cost as exactly aimed.

The final objective of this thesis is to develop a novel framework suitable for characterizing open quantum systems beyond the standard quantum noise spectroscopy technique. The framework must work independent of any assumptions on noise and control which is the main limitation of the standard technique. Additionally, it must be suitable for performing usual tasks such as decoherence suppression, quantum control, and estimation of the power spectral density of the noise. Finally, the experimental procedures must be practical and possible to implement. It also needs to take any experimental constraints into consideration for designing control. This objective contributes to the first, second, and fourth aims of the stakeholders in similar way as the second objective. Since this objective focuses on noisy quantum systems and how to control them, this contributes more significantly to the first aim compared to the second objective.

1.4 Research Methods

1.4.1 Classical Machine Learning

For the first objective of this thesis, the Matrix-Exponentiated Gradient (MEG) method will be utilized to develop an online quantum state estimation algorithm. This method was introduced in classical machine learning literature in the context of kernel learning in support vector machines [5, 6]. The original proposal was to develop an estimation algorithm for real symmetric matrices. In this thesis, the MEG update rule as well as the related convergence identities will be ported to the quantum domain. This will be done by generalizing the mathematics to work with complex Hermitian matrices instead of real symmetric matrices. However, the

original statements about convergence and their proofs would have to be modified entirely to suit the objective of this problem which is quantum state estimation. Tools from quantum information theory will be utilized as will be discussed later in this section.

Additionally, techniques from classical machine learning will be utilized and adapted to model quantum systems. The problem of using standard "blackbox" architectures, is that they do not respect the mathematical constraints of quantum quantities such as Hermiticity, Unitarity, etc. As a consequence, the standard algorithms will not result in sound predictions (for example having Pauli observables that are outside the range $[-1, 1]$). Additionally, due to the abstract nature of these structures, it is not possible to extract useful physical quantities required for further applications. On the other hand, training a "whitebox" model derived from the physics of the system is not possible without assumptions, approximations, and idealizations. So, we will propose an approach that provides a mix between those two architectures. Standard blackbox layers (such as Gated-Recurrent Units (GRU) and Neural Networks (NN)) as well as novel customized structures whitebox layers (such as layers to compute Hamiltonians) will be merged together in a hybrid architecture. We will refer to this architecture as a "graybox". This novel approach will be utilized to model closed quantum systems as well as open quantum systems, satisfying the second and third objectives of this thesis. Moreover, it will be used to design control pulses for different purposes under realistic experimental conditions.

1.4.2 Quantum Information Theory

We will utilize many tools from quantum information theory particularly for proving the convergence of the MEG-based algorithm for the first objective of this thesis. This will include the use of information-theoretic quantities (such as the quantum relative entropy), distance measures (such as the trace distance), matrix inequalities (such as Golden-Thompson inequality and Jensen's inequality), methods and techniques (such as quantum unitary t-designs), and probabilistic notions (such as Markov's inequality and stochastic convergence). Additionally, the notion of quantum channels and quantum tomography will be needed as a foundation for the three objectives of this thesis.

1.4.3 Software Implementation

MATLAB will be used to implement the algorithm for the first objective. The algorithms developed in this thesis for the second and third objectives will be implemented in Python together with Tensorflow [7] and Keras [8]. As in any machine learning research, datasets are needed to train and assess the performance of any

novel algorithm. For this purpose, simulators for closed as well as open quantum systems will be implemented in Python and used to generate suitable datasets. The source code for all the methods will be available as open source. Since the computations are very intensive, the proposed methods will be executed on the UTS iHPC cluster facility.

1.5 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 covers the essential background including standard calculations presented in detail as needed for the rest of this thesis. Particularly, it provides an overview on modelling, simulating, characterizing, and controlling quantum systems. Starting Chapter 3, the novel contributions are presented consecutively with each chapter targeting one objective of this thesis. The first result in Chapter 3 is an online quantum state estimation algorithm with provable convergence. Chapter 4 presents a novel machine learning framework to model closed quantum systems. This framework is applied in a practical application which is modelling and controlling a photonic circuit. The framework is then generalized to open quantum systems in Chapter 5. The proposed method is applied to the case of spin qubits in the presence of a classical environment. Finally in Chapter 6, a summary and an outlook on the future work of this thesis is provided.

Chapter 2

Background

This chapter gives an extensive background needed for presenting the novel results in this thesis. The chapter starts with preliminaries from machine learning and signal processing in Section 2.1. Next, in Section 2.2 an overview is given on the different ways to mathematically model noisy quantum systems. After that, Section 2.3 provides an extensive analysis of a noisy qubit subjected to a classical environment, and the different models introduced earlier are applied and contrasted. The chapter then moves on introducing quantum tomography as a fundamental tool for characterizing quantum systems in Section 2.4. Finally, the chapter ends with a brief overview on quantum control in Section 2.5. This chapter is written with the purpose of trying to introduce concepts from machine learning and quantum information in an accessible way to readers across both communities, and thus can be skipped by advanced readers.

2.1 Preliminaries

2.1.1 Overview on Neural Networks and Gated-Recurrent Units

In this section we give a brief overview on some of the commonly used classical machine learning blackboxes that will be used later in this thesis. The first is the neural network, which is a non-linear modular structure composed of basic computational units called neurons. A neuron calculates a weighted average of its inputs and then applies a non-linear transformation, generating a single output. If we denote the set of inputs as $\mathbf{x} = [x_1, x_2, \cdots x_n]^T$, then the output would be

$$y = f\left(w_0 + \sum_{i=1}^{n} w_i x_i\right) \tag{2.1}$$

$$= f\left(\mathbf{W}\mathbf{x} + w_0\right) \tag{2.2}$$

where w_i are called the weights of the neuron, w_0 is called the bias and the generally non-linear function $f(\cdot)$ is called the activation function. The most commonly-used activations are the linear $f(x) = x$, the sigmoid $f(x) = \frac{1}{1+e^{-x}}$, and hyperbolic tangent $f(x) = \tanh(x)$. The nice property about those particular functions is that their gradients are easy to evaluate (1, $f(x)(1 - f(x))$, and $1 - f^2(x)$ respectively). Nonetheless other functions can be used. The weights and the bias are chosen through the training process to generate some desired output. For instance, if the neuron output is denoted by y, and the desired output is y_d, then by constructing the loss function

$$L = (y - y_d)^2, \tag{2.3}$$

we can use the steepest descent method to find the optimal weights as

$$w_i^{(t+1)} = w_i^{(t)} - \eta \frac{\partial L}{\partial w_i^{(t)}}, \tag{2.4}$$

where $w_i^{(t)}$ is the i^{th} weight at iteration t and η is the learning rate. So, starting with random weights $w_i^{(0)}$ the iterations are applied until convergence. Now, this single neuron generates one output only. If we want to generate multiple outputs then we can construct a layer of neurons each acting on the same input. In many applications, this structure might not be complex enough to model our data. So, we can connect multiple layers where the output of one layer is fed as input to the next layer. The last layer is called the output layer, and the number of neurons there should match the number of desired outputs. The other layers are called hidden layers and they can have arbitrary number of neurons. One can also derive an update equation (using the chain rule) which is referred to as the backpropagation. There exist many variants that enhance the basic update rule. NNs turn out to be very useful many applications such as classification and regression.

The second type of blackboxes that will be used in this thesis is the Recurrent Neural Network (RNN). This is a structure that allows processing sequences. Besides its input $\mathbf{x}_t$ and output $\mathbf{y}_t$ at time instant t, it has an internal hidden state denoted by $\mathbf{h}_t$. At each time instant, the RNN processes the inputs to update the hidden state from the previous time instant, as well as generates the new output. So, it works like a feedback system. Generally, the new hidden state $\mathbf{h}_{t+1}$ and the output $\mathbf{y}_t$ can both depend on $\mathbf{x}_t$ and $\mathbf{h}_t$. The relations between different variables would depend on some weights which are adjusted during training to produce some desired output. Based on this idea, there are many "update rules" resulting in various kinds of structures. In this thesis, we make use of the Gated Recurrent Unit (GRU) [9]. It consists of the following structure. First, there is a "reset gate", which is a single-

layer neural network with sigmoid activation that operates on the concatenation of current input at time instant t and the previous hidden state $\mathbf{h}_t$, to produce the output $\mathbf{r}_t$. In other words,

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r\right), \tag{2.5}$$

where $\mathbf{W}_r$, $\mathbf{U}_r$, and $\mathbf{b}_r$ are the weights and the bias of the neural network, and $\sigma(\cdot)$ is the sigmoid function. The second component of a GRU is the update gate, which is similarly a single-layer neural network with sigmoid activation,

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z\right). \tag{2.6}$$

After computing the outputs from the reset and update gates we can now calculate $\tilde{\mathbf{h}}_t$ which represents the new information we need to add to our hidden state,

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \left(\mathbf{r_t} \odot \mathbf{h}_{t-1}\right) + \mathbf{b}_h\right), \tag{2.7}$$

where $\odot$ is the Hadamard product (i.e. element-wise multiplication $(A \odot B)_{ij} = (A)_{ij}(B)_{ij}$). The final step is to calculate the new hidden state which would be a weighted average of the existing state $\mathbf{h}_t$ and the new information $\tilde{\mathbf{h}}_t$, using the output of the update gate $\mathbf{z}_t$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t. \tag{2.8}$$

The output of the GRU at time t is simply $\mathbf{y}_t = \mathbf{h}_t$. The training will involve updating all the weight matrices and bias vectors, such that we obtain a target sequence of vectors $\mathbf{y}_t$ at every time instant t. The GRU is a special class of a more general structure called the Long Short-Term Memory (LSTM) [10]. In an LSTM, there is a third gate that calculates the output given the hidden state rather than just an identity gate as in the case of a GRU. The name LSTM comes from the fact that the hidden state gets updated at every time step without neither completely neglecting the new information $\tilde{\mathbf{h}}_t$ nor forgetting completely the old information $\mathbf{h}_{t-1}$. In this sense, it retains both a long and a short-term memory. These recurrent networks turn out to be very successful in application of time series analysis and natural language processing. However, they are generic enough for any application that involves sequence processing.

Although ML techniques have proven to be successful for an endless amount of applications in all domains, one has to understand the limitations and challenges of those methods. The first challenge is that the training procedure requires the preparation of large datasets, over which the loss function is optimized. This is needed

so that the ML models are able to generalize to new examples, and not just predict the outputs corresponding to the training examples. For some applications (such as image recognition), this can be easily done because of the availability of thousands of images. For other applications (such as characterization of devices), this requires physically performing a large number of experiments which might be very time-consuming to the extent of becoming infeasible. The second challenge is then how to choose the size of the dataset. There is no systematic way to answer this question, as it depends on the application itself and how accurate we want our predictions. It also depends on the model complexity (such as number of nodes in an NN) and the number of the inputs and outputs of the model. The general rule of thumb is that the higher these numbers are, the larger the dataset is needed. The third challenge is that even if we are able to construct a large dataset, there is no guarantee that this will work for every possible input. This is very similar to non-linear curve fitting. If the points are concentrated at some interval, then the prediction of the fitted curve will most likely be accurate on that interval but might not be as accurate for points outside the interval. Finally, a major part of the design process of ML algorithms is based on heuristics rather than a systematic approach. Usually, there are many hyperparameters that can be tuned which includes the model structure, the loss function, the training algorithm, and the learning rate. Although there are emerging techniques that automate the choice of these hyperparameters, they are all computationally intensive and might not be feasible to implement.

Regarding the nature of the applications we are presenting in this thesis, namely characterization and control of quantum devices, these challenges are still manageable to address for small-scale devices. In Chapters 4 and 5, we will discuss the utilization of ML techniques for these applications as well as the associated limitations and challenges.

2.1.2 Cumulants of Gaussian Random Variables

In this section, we present for the sake of completeness the calculations of the statistics of a particular random variable that appears frequently in the context of open quantum systems and quantum control. We are interested in the random variable of the form

$$\Phi(T) = \int_0^T \beta(\tau) y(\tau) d\tau, \tag{2.9}$$

where $y(t)$ is any deterministic signal, and $\beta(t)$ is a Gaussian random process. Recall that a random process $\beta(t)$ has a Gaussian distribution if at each time instant t, $\beta(t)$ is a random variable drawn from a Gaussian distribution. This means that the joint

distribution over any finite set of indices is also Gaussian, as well as any linear combination of $\beta(t)$ over different time instants. This implies that the random variable $\Phi(T)$ is also Gaussian distributed. Next, recall the cumulant generating function $\chi(t)$ of a random variable X can be defined in terms of the moment generating function $M_X(t)$ as

$$\begin{aligned}\chi(t) &= \log M_X(t) && (2.10)\\ &= \log \mathbb{E}\left\{e^{tX}\right\} && (2.11)\\ &= \log \sum_{k=0}^{\infty} \frac{t^k}{k!} m_k, && (2.12)\end{aligned}$$

where $m_k = \mathbb{E}\left\{X^k\right\}$ is k^{th} moment of X. The cumulants C_k are defined by expanding the cumulant generating function as a Taylor series:

$$\chi(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} C_k, \tag{2.13}$$

where

$$C_k = \chi(t)^{(k)}|_{t=0}. \tag{2.14}$$

For example, the first three cumulants can be calculated as follows.

$$\begin{aligned}C_0 &= \chi(0) && (2.15)\\ &= \log \mathbb{E}\left\{e^{tX}\right\}|_{t=0} && (2.16)\\ &= \log 1 && (2.17)\\ &= 0. && (2.18)\end{aligned}$$

This is why the cumulant expansion usually starts at $k = 1$.

$$\begin{aligned}C_1 &= \frac{d}{dt} \log \mathbb{E}\left\{e^{tX}\right\}|_{t=0} && (2.19)\\ &= \frac{\mathbb{E}\left\{X e^{tX}\right\}}{\mathbb{E}\left\{e^{tX}\right\}}|_{t=0} && (2.20)\\ &= \mathbb{E}\left\{X\right\}. && (2.21)\end{aligned}$$

$$\begin{aligned}C_2 &= \frac{d^2}{dt^2} \log \mathbb{E}\left\{e^{tX}\right\}|_{t=0} && (2.22)\\ &= \frac{d}{dt} \frac{\mathbb{E}\left\{X e^{tX}\right\}}{\mathbb{E}\left\{e^{tX}\right\}}|_{t=0} && (2.23)\end{aligned}$$

$$= \frac{\mathbb{E}\{e^{tX}\}\mathbb{E}\{X^2e^{tX}\} - \mathbb{E}\{Xe^{tX}\}\mathbb{E}\{Xe^{tX}\}}{(\mathbb{E}\{e^{tX}\})^2}|_{t=0} \tag{2.24}$$

$$= \mathbb{E}\{X^2\} - \mathbb{E}\{X\}^2 \tag{2.25}$$

$$= \text{var}\{X\}. \tag{2.26}$$

Now, if X has a Gaussian distribution with mean μ and variance σ^2, then

$$M_X(t) = \mathbb{E}\{e^{tX}\} \tag{2.27}$$

$$= \int_{-\infty}^{\infty} f_X(x)e^{tx}dx \tag{2.28}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} e^{tx}dx \tag{2.29}$$

$$= e^{\frac{1}{2}\sigma^2t^2+\mu t} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-(\mu+\frac{1}{2}\sigma^2t))^2}{2\sigma^2}} dx \tag{2.30}$$

$$= e^{\frac{1}{2}\sigma^2t^2+\mu t} \tag{2.31}$$

Therefore, the cumulant generating function in this case is

$$\chi(t) = \log M_X(t) \tag{2.32}$$

$$= \log e^{\frac{1}{2}\sigma^2t^2+\mu t} \tag{2.33}$$

$$= \mu t + \frac{1}{2}\sigma^2t^2, \tag{2.34}$$

and so the cumulants are simply: $C_1 = \mu, C_2 = \sigma^2, C_{k>0} = 0$.

Going back to the random variable of interest $\Phi(T)$, assuming that $\beta(t)$ has zero mean, then

$$C_1 = \mathbb{E}\{\Phi(T)\} \tag{2.35}$$

$$= \mathbb{E}\left\{\int_0^T \beta(\tau)y(\tau)d\tau\right\} \tag{2.36}$$

$$= \int_0^T \mathbb{E}\{\beta(\tau)\}y(\tau)d\tau \tag{2.37}$$

$$= 0, \tag{2.38}$$

and,

$$C_2 = \mathbb{E}\{\Phi(T)^2\} \tag{2.39}$$

$$= \mathbb{E}\left\{\int_0^T \int_0^T \beta(\tau_2)\beta(\tau_1)y(\tau_2)y(\tau_1)d\tau_2d\tau_1\right\} \tag{2.40}$$

$$= \int_0^T \int_0^T \mathbb{E}\{\beta(\tau_2)\beta(\tau_1)\}y(\tau_2)y(\tau_1)d\tau_2d\tau_1 \tag{2.41}$$

$$= \int_0^T \int_0^T R_\beta(\tau_1, \tau_2) y(\tau_2) y(\tau_1) d\tau_2 d\tau_1. \tag{2.42}$$

Assuming the process $\beta(t)$ is a stationary process (i.e. statistics do not depend on time), and that $y(\tau) = 0$ for $\tau > T$ or $\tau < 0$, then:

$$\begin{aligned}
C_2 &= \int_0^T \int_0^T R_\beta(\tau_1, \tau_2) y(\tau_2) y(\tau_1) d\tau_2 d\tau_1 && (2.43)\\
&= \int_0^T \int_0^T R_\beta(\tau_1 - \tau_2) y(\tau_2) y(\tau_1) d\tau_2 d\tau_1 && (2.44)\\
&= \int_0^T d\tau_1 y(\tau_1) \int_0^T R_\beta(\tau_1 - \tau_2) y(\tau_2) d\tau_2 && (2.45)\\
&= \int_{-\infty}^{\infty} d\tau_1 y(\tau_1) \int_{-\infty}^{\infty} R_\beta(\tau_1 - \tau_2) y(\tau_2) d\tau_2 && (2.46)\\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\tau_1 y(\tau_1) \int_{-\infty}^{\infty} S(\omega) Y(\omega) e^{+i\omega\tau_1} d\omega && (2.47)\\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) Y(\omega) \int_{-\infty}^{\infty} y(\tau_1) e^{+i\omega\tau_1} d\tau_1 d\omega && (2.48)\\
&= \int_{-\infty}^{\infty} S(\omega) Y(\omega) Y^*(\omega) d\omega && (2.49)\\
&= \int_{-\infty}^{\infty} S(\omega) |Y(\omega)|^2 d\omega, && (2.50)
\end{aligned}$$

where $S(\omega)$ is the power spectral density of the random process $\beta(t)$, and $Y(\omega)$ is the Fourier transform of the function $y(\tau)$ (we take the definition of the Fourier transform to be symmetric with the $1/\sqrt{2\pi}$ factor). Now, we can write down the expression of the cumulant generating function for the process $\Phi(t)$.

$$\begin{aligned}
\mathbb{E}\left\{e^{v\Phi(T)}\right\} &= e^{\chi(v)} && (2.51)\\
&= e^{\sum_{k=1}^{\infty} \frac{v^k}{k!} C_k} && (2.52)\\
&= e^{C_1 v + \frac{v^2}{2!} C_2} && (2.53)\\
&= e^{\frac{v^2}{2} \int_{-\infty}^{\infty} S(\omega) |Y(\omega)|^2 d\omega}, && (2.54)
\end{aligned}$$

In particular when $v = -i$, we get

$$\begin{aligned}
\mathbb{E}\left\{e^{-i\Phi(T)}\right\} &= \mathbb{E}\left\{e^{-i \int_0^T \beta(\tau) y(\tau) d\tau}\right\}\\
&= e^{-\frac{1}{2} \int_{-\infty}^{\infty} S(\omega) |Y(\omega)|^2 d\omega}. && (2.55)
\end{aligned}$$

This expression is of extreme importance as it is the foundation of dynamical decoupling and quantum noise spectroscopy. This will be discussed later in this chapter. To emphasize again the assumptions are: **$\boldsymbol{\beta(t)}$ is zero-mean, stationary, Gaus-**

sian random process, and $y(t)$ has support on $[0, T]$. If those assumptions are not satisfied, then it is more challenging to find a closed-form expression for this expectation. Particularly, if the process $\beta(t)$ is not Gaussian distributed, then the cumulant expansion is infinite. In this case it is truncated for calculation purposes, limiting the accuracy of any quantum calculations. Now, if $y(t) = 1$ for $0 < t < T$, and 0 otherwise, then

$$\begin{aligned} Y(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y(\tau) e^{-i\omega\tau} d\tau && (2.56) \\ &= \frac{1}{\sqrt{2\pi}} \int_{0}^{T} e^{-i\omega\tau} d\tau && (2.57) \\ &= \frac{1}{\sqrt{2\pi}} e^{-i\omega T/2} \frac{T \sin(\omega T/2)}{\omega T/2} && (2.58) \\ &= \frac{T}{\sqrt{2\pi}} e^{-i\omega T/2} \mathrm{sinc}(\omega T/2), && (2.59) \end{aligned}$$

and the expectation reduces to

$$\begin{aligned} \mathbb{E}\left\{e^{-i\Phi(T)}\right\} &= \mathbb{E}\left\{e^{-i\int_0^T \beta(\tau)d\tau}\right\} \\ &= e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty} S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega}. && (2.60) \end{aligned}$$

2.2 Modelling Quantum Systems

In this chapter, we denote the quantum system of interest S, and associate it with a Hilbert space $\mathcal{H}_S$. We focus only on finite-dimensional systems. The space of linear operators on S is denoted by $L(S)$. The space of positive semidefinite operators (Hermitian operators with non-negative eigenvalues) is defined as

$$P(S) = \{A^\dagger A : A \in L(S)\}, \tag{2.61}$$

the space of density operators $D(S)$ is defined as

$$D(S) = \{\rho \in P(S) : \mathrm{tr}(\rho) = 1\}. \tag{2.62}$$

A quantum system is completely characterized by its state $\rho \in D(S)$. The process of estimating the state of the system given a set of measurements is referred to as quantum state tomography. An isolated system with no unwanted interactions with its environment is referred to as a closed system. In this case, the state of the system evolves in time via the Liouville–von Neumann equation (setting $\hbar = 1$)

$$\dot{\rho} = -i[H_S, \rho], \tag{2.63}$$

where H_S is the system Hamiltonian which encodes all the physical forces acting on the system. The solution is

$$\rho(t) = U(t)\rho(0)U(t)^\dagger, \tag{2.64}$$

where the evolution matrix $U(t) = e^{-iH_S t}$ if H_S is time-independent. On the other hand, if H_S is time-dependent, then the evolution is given by

$$U(t) = \mathcal{T}_+ e^{-i\int_0^t H_S(s)ds} \tag{2.65}$$

$$:= \lim_{M\to\infty} e^{-iH_S(t_M)\Delta T} e^{-iH_S(t_{M-1})\Delta T} \cdots e^{-iH_S(t_0)\Delta T}, \tag{2.66}$$

where $\mathcal{T}_+$ is the time-ordering operator, $t_k = k\Delta T$ and $\Delta T = \frac{T}{M}$.

In general, many quantum systems are very hard to completely isolate, and thus would interact non-trivially with the environment. This imposes challenges regarding engineering quantum systems for purposes such as quantum computation, quantum sensing, quantum communications, etc. Modelling such "open" quantum systems is not as straightforward as the case for closed systems. In this chapter, we emphasize on three approaches for modelling open quantum systems. The first approach is information-theoretic where the noise is modelled as a quantum channel. The second approach is physics-driven through the use of master equations and is only valid under very specific assumptions. The final approach is also physics-driven and is based on studying the effective dynamics of the observables rather than the dynamics of the state. These three approaches will be presented next.

2.2.1 Noise as a Quantum Channel

The first method to model an open quantum system is using an information-theoretic approach, where the noise effectively acts a quantum channel acting on the system. We define the identity map as $\mathcal{I}(\rho) = \rho$. A quantum channel is described by a Completely-Positive Trace-Preserving (CPTP) map $\mathcal{E} : L(S) \to L(S)$. Complete-Positivity refers to the condition that for any auxiliary system S',

$$\forall \rho \in L(S \otimes S') : (\mathcal{E} \otimes \mathcal{I}_{S'})(\rho) \geq 0. \tag{2.67}$$

That is, applying the channel to a part of a composite system produces a new physically valid composite state (with non-negative eigenvalues). Trace-Preserving refers to the condition that,

$$\forall \rho \in L(S) : \mathrm{tr}(\mathcal{E}(\rho_S)) = \mathrm{tr}(\rho_S). \tag{2.68}$$

These constraints are required to preserve the mathematical properties of the quantum state after applying the channel. We can also consider the environment E which is everything outside the system of interest having unwanted interactions with it. In this case, the composite system and environment become closed, and so their joint evolution is unitary. The state of the system alone after applying the channel can be extracted by taking the partial trace over the environment,

$$\mathcal{E}(\rho) = \mathrm{tr}_E \left(U(\rho_S \otimes \rho_E)U^\dagger \right), \tag{2.69}$$

where, ρ_S is the initial state of the system, ρ_E is the initial state of the environment, and U is some unitary gate acting on the composite system. Equivalently, this can be expressed using the Kraus representation:

$$\mathcal{E}(\rho_S) = \sum_k E_k \rho_S E_k^\dagger, \tag{2.70}$$

where the Kraus operators E_k are also called superoperators (since they act on $L(S)$ rather than $\mathcal{H}_S$). These operators must satisfy the normalization condition that $\sum_k E_k^\dagger E_k = 1$. Finally, a quantum channel can be described using the Choi-Jamiolkowski isomorphism (which is conceptually similar to the idea of an impulse response of a classical channel). Introducing an identical system S' to S, and choosing a set of orthonormal basis $\{|i\rangle\}_i$ for each of them, we can define the un-normalized maximally entangled state as $|\Psi\rangle = \sum_i |i\rangle_S |i\rangle_{S'}$. The Choi-Jamiolkowski state J of a quantum map $\mathcal{E}$ can be defined as follows.

$$J = (\mathcal{E} \otimes \mathcal{I})(|\Psi\rangle \langle\Psi|), \tag{2.71}$$

The result of applying the channel to any state ρ_S can then be expressed as

$$\mathcal{E}(\rho_S) = \mathrm{tr}_{S'} \left(J(I \otimes \rho_S^\intercal) \right). \tag{2.72}$$

The complete-positivity of the channel $\mathcal{E}$ is equivalent to the positive definiteness of J, i.e. $J \geq 0$. The trace-preserving condition of $\mathcal{E}$ is equivalent to the condition that $\mathrm{tr}_{S'}(J) = I_S$. The representations in Equations 2.69, 2.70, and 2.72 are all equivalent for CPTP maps, one representation can be calculated from another.

2.2.2 The Lindblad Master Equation

The information-theoretic modelling with its way of abstracting out the exact physics of the system is suitable for many applications including quantum information theory, quantum error correction, and quantum tomography. However, it is not suffi-

cient for quantum control applications, where we need to have more access to the actual physics of the system. The purpose of quantum control is to find pulse sequences that drive the system to a desired target. Thus, we need to know exactly how the pulses affect the dynamics of the system, and so we need a continuous-time model rather than the discrete-time quantum channel model. If the system is closed, we can just use the Liouville–von Neumann equation. When the system is open, it becomes more complicated to model the dynamics. In this section, we describe one approach which is the Lindblad master equation. The equation describes the evolution of the system's state ρ_S, subjected to a Hamiltonian H_S, in the presence of an interaction with the environment encoded by a set of dissipators $\mathcal{D}_k$ as

$$\dot{\rho}_S = \mathcal{L}[\rho_S] = -i[H_S, \rho_S] + \sum_k \mathcal{D}_k[\rho_S] \tag{2.73}$$

$\mathcal{L}$ is called the Lindbladian, and the terms $\mathcal{D}_k$ take the form:

$$\mathcal{D}_k[\rho_S] = L_k \rho_S L_k^\dagger - \frac{1}{2}\left(L_k^\dagger L_k \rho_S + \rho_S L_k^\dagger L_k\right), \tag{2.74}$$

with each of the L_k being a superoperator. For few quantum systems, we can derive exactly this form. However, for most quantum systems, this is only possible under very non-rigorous and non-justifiable approximations such as the "Born-Markov" approximation, or some more justifiable approximations such as the "Rotating Wave Approximation (RWA)" which is only valid in certain experimental conditions. Other more rigorous approaches are the "weak-coupling limit" and "strong-coupling limit" which may not be valid in actual experimental setups. Sometimes it is possible to derive an effective Lindbladian by fitting measured data under free evolution (absence of control). The problem is that this effective model does not take into account the effect of controls. In other words, the control would change the Lindbladian. For these reasons, we avoid working with master equations in this thesis. Nonetheless, we present the idea for the sake of completeness as a major part of the literature on quantum control relies on this formulation. In what follows, we give a non-formal derivation of the Lindblad master equation. Starting from the Kraus representation of a quantum channel, looking at the state of the system after an infinitesimal increment in time,

$$\rho_S(t + \delta t) = \sum_k E_k(\delta t)\rho_S(t)E_k^\dagger(\delta t). \tag{2.75}$$

By taking the limit as $\delta t \to 0$, then the infinitesimal change in ρ up to first order in δt is

$$\rho_S(t+\delta t) = \rho_S(t) + \delta t \delta\rho_S + O(\delta t^2) \tag{2.76}$$

Assuming the Kraus operators can be expanded as power series in $\sqrt{\delta t}$, in the form $E_k = E_k^{(0)} + \sqrt{\delta t}E_k^{(1)} + \delta t E_k^{(2)} + \cdots$. We can express E_0 arbitrarily utilizing the system Hamiltonian H_S in the form $E_0 = I + \delta t(-iH_S + K) + O(\delta t^2)$, and $E_k = \sqrt{\delta t}L_k + O(\delta_t)$. We need to ensure that we get an infinitesimal change in ρ_S to the first order in δt, and also ensure that we keep the normalization of the Kraus operators. This will be fixed by appropriately choosing K as will be shown later. Getting back to the evolution equation and expanding it in terms of the chosen representation of the Kraus operator

$$\begin{aligned}
\rho_S(t+\delta t) &= \sum_k E_k(\delta t)\rho_S(t)E_k^\dagger(\delta t) && (2.77)\\
&= E_0(\delta t)\rho_S(t)E_0^\dagger(\delta t) + \sum_{k>0} E_k(\delta t)\rho_S(t)E_k^\dagger(\delta t) && (2.78)\\
&= \rho_S(t) + \delta t(-iH_S + K)\rho(t) + \rho(t)\delta t(iH_S + K^\dagger)\\
&+ \delta t\sum_{k>0} L_k\rho_S(t)L_k^\dagger + O(\delta t^2) && (2.79)\\
&= \rho_S(t) + \delta t\left(-i\left[H_S, \rho_S(t)\right] + K\rho_S(t) + \rho_S(t)K^\dagger + \sum_{k>0} L_k\rho_S(t)L_k^\dagger\right)\\
&+ O(\delta t^2). && (2.80)
\end{aligned}$$

Now, for $\rho_S(t+\delta t)$ to be Hermitian, K must be Hermitian. So,

$$\begin{aligned}
\rho_S(t+\delta t) &= \rho_S(t) + \delta t\left(-i\left[H, \rho_S(t)\right] + K\rho_S(t) + \rho_S(t)K + \sum_{k>0} L_k\rho_S(t)L_k^\dagger\right)\\
&+ O(\delta t^2). && (2.81)
\end{aligned}$$

Now, we need to fix K so that normalization of the Kraus operators is preserved.

$$\begin{aligned}
\sum_k E_k^\dagger E_k &= E_0^\dagger E_0^\dagger + \sum_{k>0} E_k^\dagger E_k\\
&= (I + \delta t(iH + K))(I + \delta t(-iH + K)) + \delta t\sum_{k>0} L_k^\dagger L_k + O(\delta t^2)\\
&= I + \delta t(-iH + K) + \delta t(iH + K) + \delta t\sum_{k>0} L_k^\dagger L_k + O(\delta t^2)\\
&= I + 2\delta t K + \delta t\sum_{k>0} L_k^\dagger L_k + O(\delta t^2) && (2.82)
\end{aligned}$$

So, for $\sum_k E_k^\dagger E_k = I$, neglecting higher orders of δt

$$K = -\frac{1}{2}\sum_{k>0} L_k^\dagger L_k. \tag{2.83}$$

Substituting back this result we get

$$\begin{aligned}\rho_S(t+\delta t) = \rho_S(t) + \delta t\left(-i\left[H,\rho_S(t)\right] - \frac{1}{2}\left(\sum_{k>0} L_k^\dagger L_k \rho(t) + \sum_{k>0}\rho(t) L_k^\dagger L_k\right)\right) \\ + \delta t\sum_{k>0} L_k \rho_S(t) L_k^\dagger + O(\delta t^2)\end{aligned} \tag{2.84}$$

Now, if we take the limit as $\delta t \to 0$, neglecting higher order terms in δt we finally get the Lindblad master equation,

$$\frac{d\rho(t)}{dt} = \mathcal{L}[\rho], \tag{2.85}$$

where

$$\mathcal{L}[\rho_S] = -i\left[H,\rho(t)\right] + \sum_{k=1}\left(L_k\rho(t)L_k^\dagger - \frac{1}{2}L_k^\dagger L_k\rho(t) - \frac{1}{2}\rho(t)L_k^\dagger L_k\right). \tag{2.86}$$

As a final note, In this non-formal proof we assumed Markovian dynamics, which means that $\rho_S(t+\delta t)$ depends only on $\rho_S(t)$, and not earlier times (hence the first-order differential equations). In general, this may not be the case. For non-Markovian dynamics, not only does the information dissipate from the system to the environment, but it can also flow back from the environment to the system. It is as if the environment has a memory that stores the information from the system.

2.2.3 Effective Observable Dynamics

The third approach for modelling open quantum systems, is through modelling the effective dynamics of the observables using a modified interaction picture. At the end, a quantum state is not directly measurable in experiment but can only be reconstructed from measured observables. We will start with the usual interaction picture, then see why it will not be suitable for our purpose, and finally present the modified interaction picture that will yield the desired form.

As it is known the Schrödinger picture is where the states are time-dependent while the operator are time-independent, which is the more famous picture. The Heisenberg picture is the opposite, so the states are constants while the operators are time-dependent. Finally, there is the interaction (Dirac) picture, where both states and operators are time-dependent. The three pictures are equivalent to each other

in the sense that they all yield the same expectation values of quantum observables, which is what can be measured physically. We can also move from one picture to another. The use of particular picture depends on the application, and one picture can make calculations easier more than another picture.

The most general form of the Hamiltonian for an open quantum system can be written in the form

$$H(t) = H_{\text{ctrl}}(t) + H_E(t) + H_{SE}(t), \tag{2.87}$$

where H_E is the Hamiltonian acting on the environment, H_{SE} is the coupling Hamiltonian acting on both system and environment, and $H_{\text{ctrl}}(t)$ is in the control Hamiltonian acting on the system including also the drifting Hamiltonian which acts as the free evolution in the absence of control. In general, the coupling Hamiltonian can be stochastic, in which case for each possible realization we get a different evolution "trajectory". In an actual experiment, it is only possible to measure expectations of observables, by repeating a prepare-evolve-measure experiment multiple times and averaging the results. Therefore, averaging over these realizations is implicit when calculating analytically any observables. Now, starting with the system in the initial state $\rho_S(0)$, the expectation of some observable O acting on the system at time $t = T$ is given by

$$\mathbb{E}\{O(T)\} = \left\langle \text{tr}\left[U(t)(\rho_S(0) \otimes \rho_E)U^\dagger(t)(O \otimes I)\right]\right\rangle_c, \tag{2.88}$$

where ρ_E is the initial state of the environment, the expectation $\langle\cdot\rangle_c$ is taken with respect to the noise realizations, and $U(t)$ is the total unitary which can be expressed as the time-ordered evolution of the total Hamiltonian $H(t)$

$$U(t) = \mathcal{T}_+ e^{-i\int_0^t H(s)ds}. \tag{2.89}$$

The problem with this form is that $U(t)$ depends on both the noise and the control. We are interested to find a form where we can separate both so that we can design then protocols for dynamical decoupling, quantum control and quantum noise spectroscopy. So, we are going to move to the interaction picture to help us separate the noise and control. This is a standard procedure. We start by separating the Hamiltonian into two parts, $H(t) = H_0(t) + H_1(t)$, $H_0(t) = H_{\text{ctrl}}$ which includes the free evolution term and the control terms, and $H_1(t) = H_E + H_{SE}$, has all the noise (environment) terms. Next, we define the unitary

$$U_0(t) = \mathcal{T}_+ e^{-i\int_0^t H_0(s)ds}, \tag{2.90}$$

and use it to transform the states and operators as follows. First, the composite state becomes

$$\rho_I(t) = U_0^\dagger(t)\rho_{SE}(t)U_0(t) \tag{2.91}$$
$$= U_0^\dagger(t)U(t)\rho_{SE}(0)U^\dagger(t)U_0(t). \tag{2.92}$$

The operator $H_0(t)$ does not change between the two pictures (because $U_0(t)$ commutes with $H_0(t)$, conjugating $H_0(t)$ by $U_0(t)$ has no effect). Next, we transform $H_1(t)$ to become $H_I(t)$ as

$$H_I(t) = U_0^\dagger(t)H_1(t)U_0(t), \tag{2.93}$$

and consequently, the interaction unitary is

$$U_I(t) = \mathcal{T}_+ e^{-i\int_0^t H_I(s)ds}. \tag{2.94}$$

On the other hand, we know that the time-evolution of the state in the interaction picture is given by

$$\frac{d}{dt}\rho_I(t) = [H_I(t), \rho_I(t)], \tag{2.95}$$

which is equivalent to

$$\rho_I(t) = U_I(t)\rho_I(0)U_I^\dagger(t). \tag{2.96}$$

By comparing this form with that in Equation 2.92 (and noticing that $\rho_I(0) = \rho_{SE}(0)$) we find that

$$U(t) = U_0(t)U_I(t) \tag{2.97}$$

which means we separated the noise and control parts. Thus, the expectation becomes

$$\mathbb{E}\{O(t)\} = \left\langle \mathrm{tr}\left[U(t)(\rho_S(0)\otimes\rho_E)U^\dagger(t)(O\otimes I)\right]\right\rangle_c \tag{2.98}$$
$$= \left\langle \mathrm{tr}\left[U_0(t)U_I(t)(\rho_S(0)\otimes\rho_E){U_I}^\dagger(t)U_0^\dagger(t)(O\otimes I)\right]\right\rangle_c \tag{2.99}$$

Now, the problem with that form is that the initial state gets conjugated by the noise unitary first. This will result in the dependence of the classical expectation on the initial quantum state and will not facilitate expressing the optimal control problem. This is why the conventional interaction picture does not solve the problem completely. The form we need is the one where the state is conjugated with the

control unitary first. So, we are going to modify the interaction picture as follows.

$$
\begin{aligned}
U(t) &= U_0(t)U_I(t) && (2.100)\\
&= U_0(t)U_I(t){U_0}^\dagger(t)U_0(t) && (2.101)\\
&= U_0(t)\mathcal{T}_+ e^{-i\int_0^t H_I(s)ds}{U_0}^\dagger(t)U_0(t) && (2.102)\\
&= U_0(t)\Bigg(1 - i\int_0^t H_I(t_1)dt_1 \\
&+ \frac{(-i)^2}{2!}\int_0^t\int_0^t H_I(t_1)H_I(t_2)dt_1dt_2 + \cdots \Bigg){U_0}^\dagger(t)U_0(t) && (2.103)\\
&= \Bigg(1 - i\int_0^t U_0(t)H_I(t_1)U_0^\dagger(t)dt_1 \\
&+ \frac{(-i)^2}{2!}\int_0^t\int_0^t U_0(t)H_I(t_1)\left(U_0^\dagger(t)U_0(t)\right)H_I(t_2)U_0^\dagger(t)dt_1dt_2 + \cdots \Bigg)U_0(t) && (2.104)\\
&= \left(\mathcal{T}_+ e^{-i\int_0^t U_0(t)H_I(s)U_0^\dagger(t)ds}\right)U_0(t) && (2.105)\\
&:= \tilde{U}_I(t)U_0(t). && (2.106)
\end{aligned}
$$

Notice, the second line is just multiplying the identity from left. In the fifth and the sixth lines we multiplied $U_0(t)$ from the left to all terms in the infinite series and $U_0^\dagger(t)$ from the right. We also resolved the identity between each H_I term. This means effectively the H_I terms gets conjugated by $U_0(t)$ inside the time-ordered exponential as in the last line. So, now we can finally express the expectation value of the observable as

$$
\begin{aligned}
\mathbb{E}\{O(t)\} &= \left\langle \mathrm{tr}_{SE}\left[U(t)(\rho_S(0)\otimes\rho_E)U^\dagger(t)O\right]\right\rangle_c && (2.107)\\
&= \left\langle \mathrm{tr}_{SE}\left[\tilde{U}_I(t)U_0(t)(\rho(0)_S\otimes\rho_E)U_0^\dagger(t)\tilde{U_I}^\dagger(t)O\right]\right\rangle_c && (2.108)\\
&= \left\langle \mathrm{tr}_{SE}\left[\tilde{U_I}^\dagger(t)O\tilde{U}_I(t)U_0(t)(\rho_S(0)\otimes\rho_E)U_0^\dagger(t)\right]\right\rangle_c && (2.109)\\
&= \left\langle \mathrm{tr}_{SE}\left[\tilde{U_I}^\dagger(t)O\tilde{U}_I(t)(U_0(t)\rho_S(0)U_0^\dagger(t)\otimes\rho_E)OO^{-1}\right]\right\rangle_c && (2.110)\\
&= \mathrm{tr}_S\left\langle\left[\mathrm{tr}_E\left[O^{-1}\tilde{U_I}^\dagger(t)O\tilde{U}_I(t)\rho_E\right]\right\rangle_c U_0(t)\rho_S(0)U_0^\dagger(t)O\right] && (2.111)\\
&= \mathrm{tr}_S\left[\left\langle O^{-1}\tilde{U_I}^\dagger(t)O\tilde{U}_I(t)\right\rangle U_0(t)\rho(0)U_0^\dagger(t)O\right] && (2.112)\\
&\equiv \mathrm{tr}_S\left[V_O U_0(t)\rho_S(0)U_0^\dagger(t)O\right]. && (2.113)
\end{aligned}
$$

In the third line we applied the cyclic property of the trace twice. In the fourth line we added an identity $I = OO^{-1}$ at the end. In the fifth line, we applied the cyclic property again, and moved the classical expectation inside and it acts only on

the first part that depends on the noise (i.e. the $\tilde{U}_I$). Note that we have also used that $\mathrm{tr}_{SE}[V_{SE}(A_S \otimes A_E)] = \mathrm{tr}_S[A_S \,\mathrm{tr}_E[V_{SE} A_E]]$.

Now, this is exactly the form we want, because we can recover the closed system dynamics ($H_1 = 0$, and thus $U_I = I$, and so $V_O = I$). The initial state is now conjugated with U_0 which is just the control part of the Hamiltonian which we have access to. Thus, we can formulate different quantum control problems utilizing this form, and the V_O operator encodes everything about the noise and its interaction with the control, independent on the initial state of the quantum system. It represents all the non-unitary effects that occur due to the noise. Thus, knowing the V_O operator completely determines the deviation of open system dynamics from the corresponding closed system dynamics. This form is valid for any quantum system (not necessarily qubits) as long as it is possible to write the interaction picture (which is the case for finite-dimensional quantum systems). Finally, calculating those operators analytically is very difficult and requires again assumptions and approximations similar to that used in deriving a master equation. However, as will be discussed later in Chapter 5, combined with machine learning techniques we show that it is possible to estimate these operators for a quantum system experimentally given some measurement data.

2.2.4 Other Modelling Approaches

There are other methods to model open quantum systems. There are variations of the master equation such as the Bloch-Redfield master equation which relates the evolution of a quantum state to the power spectrum of the noise affecting the system. Similar to Lindblad master equation, deriving the equation depends on some assumptions and approximations. Another information-theoretic framework is the process tensor (also known as process matrix) in [11, 12] which is based on the idea of quantum combs [13]. The process tensor can be thought of a map between a sequence of control operations and the evolved state of the system, taking into consideration interaction with the environment. The framework is general requiring no assumptions on the noise, and thus is suitable for modelling non-Markovian systems. It is also suitable if the control sequence is correlated (for example when doing a feedback control). Finally, it has many foundational applications such as quantum causality [14].

2.3 Application: Modelling a Noisy Qubit

We conclude the discussion on modelling quantum systems by giving an extensive analysis of a noisy qubit undergoing dephasing, and how to apply the different

modelling approaches discussed in the previous section.

2.3.1 Physical Model

Assume a qubit with the Hamiltonian

$$H(t) = \frac{1}{2}\left(\Omega\sigma_z + f(t)\sigma_x\right) + \frac{1}{2}\beta(t)\sigma_z. \tag{2.114}$$

The first term consists of the drifting (free evolution) which happens to be a rotation around z-axis with frequency Ω, and control with a pulse sequence $f(t)$ along x-axis. The second term represents the noise, modelled by a random process $\beta(t)$, and thus we have no control over this term. This term effectively represents a random rotation of the qubit around z-axis. The Hamiltonian of the system is stochastic. This means that every time the qubit is initialized to the same state and under the same control, the evolved final state will be random with some distribution related to the distribution of the noise. In order to be able to find analytical closed-from solution, we will assume that $\beta(t)$ is a stationary, zero-mean, Gaussian process.

2.3.2 Free Evolution

Now, we will analyse the system in more detail. First, we will start with the free evolution case (i.e. we do not apply any controls, $f(t) = 0$). In this case, the two terms in the Hamiltonian commute, and so the evolution matrix of the system obtained by solving Schrödinger's equation over any time interval $[t_0, t_1]$ is given by

$$U_f(t_1, t_0) = e^{-i\int_{t_0}^{t_1} H d\tau} \tag{2.115}$$

$$= e^{-i\frac{1}{2}\int_{t_0}^{t_1}(\Omega+\beta(\tau))d\tau\sigma_z} \tag{2.116}$$

$$= e^{-i\frac{1}{2}\left(\Omega(t_0-t_1)+\int_{t_0}^{t_1}\beta(\tau)d\tau\right)\sigma_z}. \tag{2.117}$$

Assume we start with a general initial state at time $t = 0$,

$$\rho_S(0) = \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix}, \tag{2.118}$$

then the evolved state at time $t = T$ given a realization of the noise process is

$$\rho_S(T)|_{\beta(t)} = U_f(T,0)\rho_S(0)U_f(T,0)^\dagger \tag{2.119}$$

$$= e^{-i\frac{1}{2}\phi(T,0)\sigma_z}\rho(0)e^{\frac{1}{2}i\phi(T,0)\sigma_z} \tag{2.120}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\phi(T,0)} & 0 \\ 0 & e^{i\frac{1}{2}\phi(T,0)} \end{pmatrix} \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} \begin{pmatrix} e^{i\frac{1}{2}\phi(T,0)} & 0 \\ 0 & e^{-i\frac{1}{2}\phi(T,0)} \end{pmatrix} \tag{2.121}$$

$$= \begin{pmatrix} \rho_{00} & e^{-i\phi(T,0)}\rho_{01} \\ e^{i\phi(T,0)}\rho_{10} & \rho_{11} \end{pmatrix}, \tag{2.122}$$

where $\phi(t_1, t_0) = \Omega(t_1 - t_0) + \int_{t_0}^{t_1} \beta(\tau)d\tau$. This corresponds to a single trajectory of the evolution. However, all other trajectories are possible evolutions, thus we need to take an average over all possible realization of noise. In experiment we only have access to expectation of observables. Therefore, we can write the final evolved state of the system as

$$\rho_S(T) = \mathbb{E}_\beta \left\{ \rho_S(T)|_{\beta(t)} \right\} \tag{2.123}$$

$$= \begin{pmatrix} \rho_{00} & \mathbb{E}_\beta \left\{ e^{-i\phi(T,0)} \right\} \rho_{01} \\ \mathbb{E}_\beta \left\{ e^{i\phi(T,0)} \right\} \rho_{10} & \rho_{11} \end{pmatrix} \tag{2.124}$$

$$= \begin{pmatrix} \rho_{00} & \Gamma(T)\rho_{01} \\ \Gamma(T)^*\rho_{10} & \rho_{11} \end{pmatrix}, \tag{2.125}$$

and

$$\Gamma(T) = \langle e^{-i\phi(T,0)} \rangle_\beta \tag{2.126}$$

$$= \langle e^{-i\left(\Omega T + \int_0^T \beta(\tau)d\tau\right)} \rangle_\beta \tag{2.127}$$

$$= e^{-i\Omega T} \langle e^{-i\int_0^T \beta(\tau)d\tau} \rangle_\beta \tag{2.128}$$

$$= e^{-i\Omega T} e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty} S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega}. \tag{2.129}$$

where $S(\omega)$ is the power spectral density of the noise process $\beta(t)$. The last step utilizes the result of the calculation in 2.60. Now, define the coherence function at time $t = T$ to be

$$W(T) := \left| \frac{\langle \rho_{01}(T) \rangle_\beta}{\langle \rho_{01}(0) \rangle_\beta} \right| \tag{2.130}$$

$$= |\Gamma(T)|. \tag{2.131}$$

This quantity is physically very significant and can be measured experimentally. Consider doing a measurement along x-axis, the average expectation of the observable at time $t = T$ is given by

$$\langle X(T) \rangle = \langle \mathrm{tr} \left(\rho_S(T)|_\beta X \right) \rangle_\beta \tag{2.132}$$

$$= \left\langle \mathrm{tr} \begin{pmatrix} \rho_{00} & e^{-i\phi(T,0)}\rho_{01} \\ e^{i\phi(T,0)}\rho_{10} & \rho_{11} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle_\beta \tag{2.133}$$

$$= \left\langle \mathrm{tr} \begin{pmatrix} e^{-i\phi(T,0)}\rho_{01} & \rho_{00} \\ \rho_{11} & e^{i\phi(T,0)}\rho_{10} \end{pmatrix} \right\rangle_\beta \tag{2.134}$$

$$= \langle e^{-i\phi(T,0)}\rho_{01} + e^{i\phi(T,0)}\rho_{10}\rangle_\beta \tag{2.135}$$

$$= 2\Re\Big(\langle e^{-i\phi(T,0)}\rangle_\beta \rho_{01}\Big) \tag{2.136}$$

$$= 2\Re(W(T)\rho_{01}) \tag{2.137}$$

$$= W(T)\langle X(0)\rangle \tag{2.138}$$

This means that the value of the observable would vary with time due to the presence of the noise. If $W(T)$ decays as T increases, then the observable would decay till it reaches zero for large T. This is what is referred to as dephasing or decoherence. The state of the qubit loses its off-diagonal elements which encode the "quantumness", and eventually the state becomes diagonal (the completely-mixed state or a classical state). The decay rate then determines the lifetime of the qubit. This is an important aspect when designing quantum memories. Notice, that the dephasing effect is basis-dependent. If we start with a state like $|0\rangle$, then we will not observe decoherence under the given Hamiltonian as $\forall t, \rho_{01} = 0$. For this state however, if the stochastic term in the Hamiltonian was in the form $\beta_x(t)\sigma_x$, then once again there is decoherence. Moreover, in the general case of multi-axis noise (i.e. having the stochastic part of the Hamiltonian in the form $\sum_{k=1}^{3}\beta_k(t)\sigma_k$), we will observe decoherence for all states. Measuring $W(T)$ experimentally is easy. We need to prepare a state where we can observe the decoherence (for instance for the given Hamiltonian choose the eigenstate of the Pauli-X operator $|+\rangle$). In this case, $\langle X(0)\rangle = 1$. Then we let the system evolve to time T, and measure the expectation of the X observable ($\langle X(T)\rangle$), and this value would correspond to $W(T)$. For the free evolution case, the coherence is given by

$$W(T) = e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty} S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega} \tag{2.139}$$

The low frequency components of the PSD of the noise affects the coherence more than the high frequency components. A worst case scenario would a coloured $1/f^\alpha$ noise (which exists for many physical systems). In the limit as $f \to 0$, we see that $S(0) \to \infty$, and thus $W(T) \to 0$, which means losing coherence rapidly. However, this is not the end of the story. We will see later how to overcome this problem using dynamical decoupling.

We can also represent the free evolution dynamics as a quantum channel. Fixing the time at $t = T$, and using the Kraus operators:

$$E_0 = \sqrt{1-p}R_z(\theta) \tag{2.140}$$

$$E_1 = \sqrt{p}R_z(\theta)\sigma_z = \sqrt{p}R_Z(\theta+\pi), \tag{2.141}$$

where

$$p = \frac{1 - W(T)}{2} \tag{2.142}$$

$$\theta = \Omega T \tag{2.143}$$

$$R_z(\alpha) = e^{-i\frac{\alpha}{2}\sigma_z}, \tag{2.144}$$

we can write the evolved state in the form $\mathcal{E}(\rho) = (1-p)E_0\rho E_0^\dagger + pE_1\rho E_1^\dagger$. We can know proceed to calculate the Choi-Jamiolkowski state as follows. First, we calculate the output of the map for the basis states,

$$\mathcal{E}(|0\rangle\langle 0|) = |0\rangle\langle 0| \tag{2.145}$$

$$\mathcal{E}(|0\rangle\langle 1|) = \Gamma(T)|0\rangle\langle 1| \tag{2.146}$$

$$\mathcal{E}(|1\rangle\langle 0|) = \Gamma(T)^*|1\rangle\langle 0| \tag{2.147}$$

$$\mathcal{E}(|1\rangle\langle 1|) = |1\rangle\langle 1|. \tag{2.148}$$

Next, using the definition of the Choi-Jamiolkowski state we get

$$J = (\mathcal{E}\otimes I)(|\Psi\rangle\langle\Psi|) \tag{2.149}$$

$$= (\mathcal{E}\otimes I)(|0\rangle|0\rangle + |1\rangle|1\rangle)(\langle 0|\langle 0| + \langle 1|\langle 1|) \tag{2.150}$$

$$\begin{aligned} &= (\mathcal{E}\otimes I)(|0\rangle\langle 0|\otimes|0\rangle\langle 0|) + (\mathcal{E}\otimes I)(|0\rangle\langle 1|\otimes|0\rangle\langle 1|) \\ &+ (\mathcal{E}\otimes I)(|1\rangle\langle 0|\otimes|1\rangle\langle 0|) + (\mathcal{E}\otimes I)(|1\rangle\langle 1|\otimes|1\rangle\langle 1|) \end{aligned} \tag{2.151}$$

$$= |0\rangle\langle 0|\otimes|0\rangle\langle 0| + \Gamma(T)|0\rangle\langle 1|\otimes|0\rangle\langle 1| + \Gamma(T)^*|1\rangle\langle 0|\otimes|1\rangle\langle 0| + |1\rangle\langle 1|\otimes|1\rangle\langle 1| \tag{2.152}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \Gamma(T) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Gamma(T)^* & 0 & 0 & 1 \end{pmatrix}. \tag{2.153}$$

We can verify the calculation by doing the inverse transformation as follows

$$\mathcal{E}(\rho) = \mathrm{tr}_{S'}(J(I\otimes\rho^\intercal)) \tag{2.154}$$

$$= \mathrm{tr}_{S'}\begin{pmatrix} 1 & 0 & 0 & \Gamma(T) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Gamma(T)^* & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \rho_{00} & \rho_{10} & 0 & 0 \\ \rho_{01} & \rho_{11} & 0 & 0 \\ 0 & 0 & \rho_{00} & \rho_{10} \\ 0 & 0 & \rho_{01} & \rho_{11} \end{pmatrix} \tag{2.155}$$

$$= \mathrm{tr}_{S'} \begin{pmatrix} \rho_{00} & \rho_{10} & \Gamma(T)\rho_{01} & \Gamma(T)\rho_{11} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Gamma(T)^*\rho_{00} & \Gamma(T)^*\rho_{10} & \rho_{01} & \rho_{11} \end{pmatrix} \tag{2.156}$$

$$= \begin{pmatrix} \rho_{00} & \Gamma(T)\rho_{01} \\ \Gamma(T)^*\rho_{10} & \rho_{11} \end{pmatrix}, \tag{2.157}$$

which is exactly the evolution we obtained earlier.

Next, we show how to express the dynamics using the effective observables picture. Defining $H_0 = \frac{1}{2}\Omega\sigma_z$, and $H_1 = \frac{1}{2}\beta(t)\sigma_z$, then at time $t = T$, we can evaluate the control unitary as

$$U_0(T) = \mathcal{T}_+ e^{-i\int_0^t H_0(s)ds} \tag{2.158}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\Omega T} & 0 \\ 0 & e^{i\frac{1}{2}\Omega T} \end{pmatrix}. \tag{2.159}$$

We previously showed that the full evolution unitary is given by

$$U(T) = \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T(\Omega+\beta(\tau))d\tau} & 0 \\ 0 & e^{-i\frac{1}{2}\int_0^T(\Omega+\beta(\tau))d\tau} \end{pmatrix}, \tag{2.160}$$

thus, the interaction unitary can be calculated as

$$U_I(T) = U_0(T)^\dagger U(T) \tag{2.161}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T\beta(\tau)d\tau} & 0 \\ 0 & e^{i\frac{1}{2}\int_0^T\beta(\tau)d\tau} \end{pmatrix}, \tag{2.162}$$

and

$$\tilde{U}_I(T) = U_0(T)U_I(T)U_0^\dagger(T) \tag{2.163}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T\beta(\tau)d\tau} & 0 \\ 0 & e^{i\frac{1}{2}\int_0^T\beta(\tau)d\tau} \end{pmatrix}. \tag{2.164}$$

Now, we can evaluate the $V_O = \langle O^{-1}\tilde{U}_I(T)^\dagger O\tilde{U}_I(T)\rangle_c$ operators for $O \in \{X, Y, Z\}$.

$$V_X = \langle X\tilde{U}_I(T)^\dagger X\tilde{U}_I(T)\rangle_c \tag{2.165}$$

$$= \begin{pmatrix} \langle e^{-i\int_0^T\beta(\tau)d\tau}\rangle_c & 0 \\ 0 & \langle e^{i\int_0^T\beta(\tau)d\tau}\rangle_c \end{pmatrix} \tag{2.166}$$

$$= \begin{pmatrix} e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty}S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega} & 0 \\ 0 & e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty}S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega} \end{pmatrix}. \tag{2.167}$$

Similarly,

$$V_Y = \langle Y\tilde{U}_I(T)^\dagger Y\tilde{U}_I(T)\rangle_c \tag{2.168}$$

$$= \begin{pmatrix} e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty} S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega} & 0 \\ 0 & e^{-\frac{T^2}{4\pi}\int_{-\infty}^{\infty} S(\omega)\mathrm{sinc}^2(\omega T/2)d\omega} \end{pmatrix}. \tag{2.169}$$

Finally, by noticing that $\tilde{U}_I(T)$ commutes with Z, we get

$$V_Z = \langle Z\tilde{U}_I(T)^\dagger Z\tilde{U}_I(T)\rangle_c \tag{2.170}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.171}$$

2.3.2.1 Markovian Evolution

Consider now the special case when the noise is white (i.e. $S(\omega) = \sigma^2$). This means that the noise at different time instants is uncorrelated. In this case, we find that

$$\Gamma(T) = e^{-i\Omega T} e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega} \tag{2.172}$$

$$= e^{-i\Omega T} e^{-\frac{\sigma^2}{2}\int_{-\infty}^{\infty} |Y(\omega)|^2 d\omega} \tag{2.173}$$

$$= e^{-i\Omega T} e^{-\frac{2\pi\sigma^2}{2}\int_{-\infty}^{\infty} |y(\tau))|^2 d\tau} \tag{2.174}$$

$$= e^{-i\Omega T} e^{-\pi\sigma^2\int_0^T d\tau} \tag{2.175}$$

$$= e^{-i\Omega T} e^{-T\pi\sigma^2}, \tag{2.176}$$

where the third line follows from Parseval's theorem. The coherence in this case is

$$W(T) = e^{-T\pi\sigma^2}. \tag{2.177}$$

The observation here is that the evolution dynamics are Markovian. The fact that there exist no correlations in the noise means that the environment is not storing information from previous time instants, i.e. it has no memory. Mathematically, the form of $W(T)$ being an exponential of the first-order polynomial of the evolution time T, allows writing a first-order differential equation for ρ_S in the form of the Lindblad master equation. This can be shown as follows.

$$\dot{\rho}_S = \frac{d}{dt}\begin{pmatrix} \rho_{00} & e^{-T(\pi\sigma^2+i\Omega)}\rho_{01} \\ e^{-T(\pi\sigma^2-i\Omega)}\rho 10 & \rho_{11} \end{pmatrix} \tag{2.178}$$

$$= \begin{pmatrix} 0 & (\pi\sigma^2 - i\Omega)e^{-t(\pi\sigma^2+i\Omega)}\rho_{01} \\ (\pi\sigma^2 + i\Omega)e^{-t(\pi\sigma^2-i\Omega)}\rho 10 & 0 \end{pmatrix} \tag{2.179}$$

$$= \begin{pmatrix} 0 & (\pi\sigma^2 - i\Omega)\rho_{01}(t) \\ (\pi\sigma^2 + i\Omega)\rho_{10}(t) & 0 \end{pmatrix} \tag{2.180}$$

$$= \begin{pmatrix} 0 & -i\Omega\rho_{01}(t) \\ i\Omega\rho_{10}(t) & 0 \end{pmatrix} + \begin{pmatrix} 0 & \pi\sigma^2\rho_{01}(t) \\ \pi\sigma^2\rho_{10}(t) & 0 \end{pmatrix} \tag{2.181}$$

$$= \frac{-1}{2}\begin{pmatrix} i\Omega\rho_{00}(t) & i\Omega\rho_{01}(t) \\ -i\Omega\rho_{10}(t) & -i\Omega\rho_{11}(t) \end{pmatrix} - \frac{-1}{2}\begin{pmatrix} i\Omega\rho_{00}(t) & -i\Omega\rho_{01}(t) \\ i\Omega\rho_{10}(t) & -i\Omega\rho_{11}(t) \end{pmatrix}$$
$$+ \begin{pmatrix} \pi\sigma^2\rho_{00}(t) & -\pi\sigma^2\rho_{01}(t) \\ -\pi\sigma^2\rho_{10}(t) & \pi\sigma^2\rho_{11}(t) \end{pmatrix} - \begin{pmatrix} \pi\sigma^2\rho_{00}(t) & \pi\sigma^2\rho_{01}(t) \\ \pi\sigma^2\rho_{10}(t) & \pi\sigma^2\rho_{11}(t) \end{pmatrix} \tag{2.182}$$

$$= \frac{-i\Omega}{2}\begin{pmatrix} \rho_{00}(t) & \rho_{01}(t) \\ -\rho_{10}(t) & -\rho_{11}(t) \end{pmatrix} - \frac{-i\Omega}{2}\begin{pmatrix} \rho_{00}(t) & -\rho_{01}(t) \\ \rho_{10}(t) & -\rho_{11}(t) \end{pmatrix}$$
$$+ \frac{1}{2}\pi\sigma^2\begin{pmatrix} \rho_{00}(t) & -\rho_{01}(t) \\ -\rho_{10}(t) & \rho_{11}(t) \end{pmatrix} - \frac{1}{2}\pi\sigma^2\begin{pmatrix} \rho_{00}(t) & \rho_{01}(t) \\ \rho_{10}(t) & \rho_{11}(t) \end{pmatrix} \tag{2.183}$$

$$= -i\left(\frac{1}{2}\Omega\sigma_z\rho(t) - \rho(t)\frac{1}{2}\Omega\sigma_z\right) + \pi\sigma^2\sigma_z\rho(t)\sigma_z - \pi\sigma^2\rho(t) \tag{2.184}$$

$$= -i\left[\frac{1}{2}\Omega\sigma_z, \rho(t)\right] + \frac{1}{2}\pi\sigma^2\left(\sigma_z\rho(t)\sigma_z - \frac{1}{2}\left(\sigma_z\sigma_z\rho(t) + \rho(t)\sigma_z\sigma_z\right)\right) \tag{2.185}$$

$$\equiv -i[H, \rho(t)] + L_1\rho(t)L_1^\dagger - \frac{1}{2}\left(L_1L_1^\dagger\rho(t) + \rho(t)L_1L_1^\dagger\right), \tag{2.186}$$

where $H = \frac{1}{2}\Omega\sigma_z$, and $L_k = \sqrt{\frac{\pi\sigma^2}{2}}\sigma_z$, which is exactly the Lindblad form $\dot{\rho}_S = \mathcal{L}[\rho]$.

2.3.2.2 Non-Markovian Evolution

Now, if the noise is coloured (i.e. the noise at different time instants is correlated), then the situation changes. For simplicity, assume that the noise has a Lorentzian PSD, $S(\omega) = \frac{1}{1+\omega^2}$, then the overlap integral can be evaluated in closed-form as

$$\Gamma(T) = e^{-i\Omega T}e^{-\frac{1}{2}(T+e^{-T}-1)}, \tag{2.187}$$

which results in a coherence in the form

$$W(T) = e^{-\frac{1}{2}(T+e^{-T}-1)} \tag{2.188}$$

$$= e^{-\frac{1}{2}T^2+O(T^3)} \tag{2.189}$$

In this case, the evolution dynamics are non-Markovian. The environment is effectively storing information from previous time instants which is modelled through the correlation of the noise. Additionally, the exponent of the coherence function is second-order in T and thus writing a Lindblad master equation is mathematically impossible.

2.3.3 Controlled-Evolution and Dynamical Decoupling

Now, let's see how the model changes when we apply control. Assume we apply a train of n-ideal instantaneous Dirac delta pulses along the x-axis. The pulses are applied at time-instants $\{t_i : 0 < t_i < T\}_{i=1}^n$, and each with power π. In other words,

$$f(t) = \sum_{i=1}^{n} \pi\delta(t - t_i), \tag{2.190}$$

The effect of one of these pulses is to introduce the evolution

$$U_x = e^{-i\frac{\pi}{2}\sigma_x} = -i\sigma_x. \tag{2.191}$$

The total evolution over the time interval $[0, T]$ is

$$\begin{aligned} U(T,0) &= U_f(T,t_n)U_x \cdots U_x U_f(t_1,t_0)U_x U_f(t_0,0) && (2.192)\\ &= e^{-i\frac{1}{2}\phi(T,t_n)\sigma_z}(-i\sigma_x)\cdots(-i\sigma_x)e^{-i\frac{1}{2}\phi(t_1,t_0)\sigma_z}(-i\sigma_x)e^{-i\frac{1}{2}\phi(t_0,0)\sigma_z}. && (2.193)\end{aligned}$$

Let's find the matrix elements of this unitary $U_{rs} = \langle r| U(T,0) |s\rangle$ in the computational basis $\{|0\rangle, |1\rangle\}$, define $t_0 = 0$ and $t_{n+1} = T$,

$$\begin{aligned} U_x U_f(t_{k+1},t_k) |s\rangle &= (-i\sigma_x)e^{-i\frac{1}{2}\phi(t_{k+1},t_k)\sigma_z} |s\rangle && (2.194)\\ &= (-i)e^{-i\frac{1}{2}\phi(t_{k+1},t_k)(-1)^s} |\bar{s}\rangle . && (2.195)\end{aligned}$$

If we have even number of pulses we get

$$U(T,0) |s\rangle = (-i)^n e^{-i\frac{1}{2}\sum_{k=0}^{n}\phi(t_{k+1},t_k)(-1)^{(k+s) \bmod 2}} |s\rangle , \tag{2.196}$$

while if we have odd number of pulses we get

$$U(T,0) |s\rangle = (-i)^n e^{-i\frac{1}{2}\sum_{k=0}^{n}\phi(t_{k+1},t_k)(-1)^{(k+s) \bmod 2}} |\bar{s}\rangle , \tag{2.197}$$

and so generally we get

$$U(T,0) |s\rangle = (-i)^n e^{-i\frac{1}{2}\sum_{k=0}^{n}\phi(t_{k+1},t_k)(-1)^{(k+s) \bmod 2}} |(n \bmod 2) \oplus s\rangle . \tag{2.198}$$

So,

$$U_{rs} = (-i)^n e^{-i\frac{1}{2}\sum_{k=0}^{n}\phi(t_{k+1},t_k)(-1)^{(k+s) \bmod 2}} \langle r|(n \bmod 2) \oplus s\rangle . \tag{2.199}$$

Now if we define the switching function $y(\tau)$ that switches between +1 and -1 whenever the π pulse is applied, i.e.

$$y(\tau) = \sum_{k=0}^{n} (-1)^k u(\tau - t_k) u(t_{k+1} - \tau), \tag{2.200}$$

where $u(t)$ is the unit step function, and define

$$\tilde{\phi}(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} (\Omega + \beta(\tau))\, y(\tau) d\tau, \tag{2.201}$$

then we can write the matrix elements of the evolution matrix as

$$U_{rs} = (-i)^n e^{-i\frac{1}{2}(-1)^{s \bmod 2} \sum_{k=0}^{n} \tilde{\phi}(t_{k+1}, t_k)} \langle r | (n \bmod 2) \oplus s \rangle \tag{2.202}$$

$$= (-i)^n e^{-i\frac{1}{2}(-1)^{s \bmod 2} \tilde{\phi}(T,0)} \langle r | (n \bmod 2) \oplus s \rangle . \tag{2.203}$$

Neglecting the global phase shift, for even n the matrix can be written as

$$U(T,0)|_{n \text{ even}} = \begin{pmatrix} e^{-i\frac{1}{2}\tilde{\phi}(T,0)} & 0 \\ 0 & e^{i\frac{1}{2}\tilde{\phi}(T,0)} \end{pmatrix}, \tag{2.204}$$

and the evolved state becomes

$$\rho_S(T)|_{n \text{ even}, \beta(t)} = \begin{pmatrix} \rho_{00} & e^{-i\tilde{\phi}(t)}\rho_{01} \\ e^{i\tilde{\phi}(t)}\rho_{10} & \rho_{11} \end{pmatrix}. \tag{2.205}$$

On the other hand for odd n, the evolution matrix can be written as

$$U(T,0)|_{n \text{ odd}} = \begin{pmatrix} 0 & e^{i\frac{1}{2}\tilde{\phi}(T,0)} \\ e^{-i\frac{1}{2}\tilde{\phi}(T,0)} & 0 \end{pmatrix}, \tag{2.206}$$

and evolved state becomes

$$\rho_S(T)|_{n \text{ odd}, \beta(t)} = \begin{pmatrix} \rho_{11} & e^{i\tilde{\phi}(t)}\rho_{10} \\ e^{-i\tilde{\phi}(t)}\rho_{01} & \rho_{00} \end{pmatrix} \tag{2.207}$$

$$= \sigma_x \rho_S(T)|_{n \text{ even}, \beta(t)} \sigma_x. \tag{2.208}$$

Therefore, to obtain the same evolved state at time $t = T$, We either apply an even number of pulses or apply an odd number of pulses and at the end of evolution time apply an X-gate. In what follows, we restrict the control to this setting. The

averaged evolved state becomes

$$\rho_S(T) = \begin{pmatrix} \rho_{00} & \Gamma_c(T)\rho_{01} \\ \Gamma_c(T)^*\rho_{10} & \rho_{11} \end{pmatrix}, \tag{2.209}$$

with

$$\Gamma_c(T) = \langle e^{-i\int_0^T (\Omega+\beta(\tau))y(\tau)d\tau} \rangle_\beta \tag{2.210}$$

$$= e^{-i\Omega\int_0^T y(\tau)d\tau} e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega}. \tag{2.211}$$

The last line follows from 2.60. $|Y(\omega)|^2$ is called the filter function, where $Y(\omega)$ is the Fourier transform of the switching function $y(\tau)$. This means the location of the pulses and the evolution duration T affects the filter function. This result is quite significant. If we calculate the coherence function, we get

$$W(T) = |\Gamma_c| \tag{2.212}$$

$$= e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega}. \tag{2.213}$$

This implies that if we want to suppress decoherence, then we need the value of the integral in the exponent to be as close as possible to zero, so that the exponential evaluates to 1. This means for different noise processes, we can engineer the pulses such that the filter function has minimal overlap with the power spectral density of the noise. This technique is referred to as Dynamical Decoupling (DD) and is proven both theoretically (for finite dimensional environments) and experimentally to be very successful. One of the oldest and most famous DD control sequences is the CPMG sequence [15]. It is characterized by applying the π-pulses at time instants $t_k = \delta_k T$, where

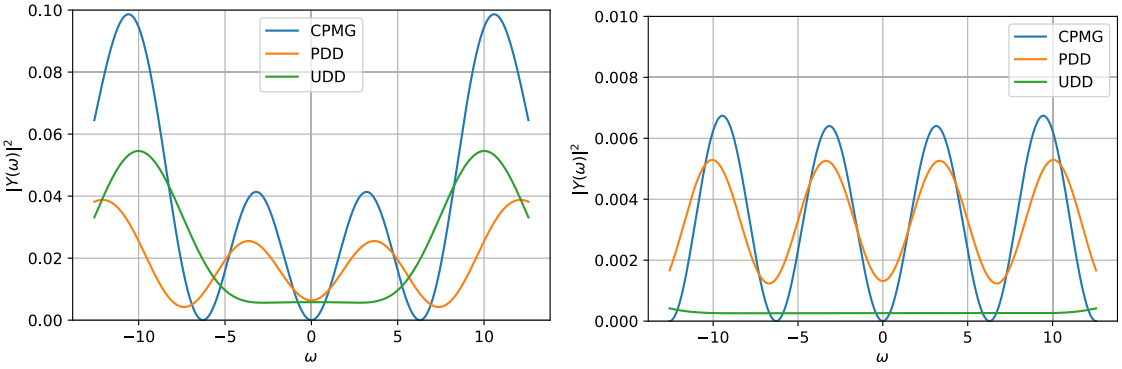
$$\delta_k = \frac{1}{n}\left(k - \frac{1}{2}\right) \tag{2.214}$$

The special case $n = 1$ corresponds to spin echo. Another technique is the Periodic Dynamical Decoupling (PDD) sequence [16] described by

$$\delta_k = \frac{k}{n+1} \tag{2.215}$$

Practically, CPMG outperforms PDD for small n, but both perform equally for large n [17]. There is also the Uhrig Dynamical Decoupling (UDD) sequence [18] which is optimal for a noise with Gaussian PSD. It is described by

$$\delta_k = \sin^2\left(\frac{\pi k}{2(n+1)}\right). \tag{2.216}$$



(a) $n = 4$ (b) $n = 10$

Figure 2.1: The plot of the filter function for CPMG, PDD, and UDD sequences where the evolution time is chosen to be $T = 1$, and the number of pulses is (a) $n = 4$ and (b) $n = 10$.

In general, these control sequences will have switching functions with Fourier transform in the form

$$Y(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y(\tau) e^{-i\omega\tau} d\tau \tag{2.217}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{0}^{t} y(\tau) e^{-i\omega\tau} d\tau \tag{2.218}$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{n+1} \int_{t_k}^{t_{k+1}} (-1)^k e^{-i\omega\tau} d\tau \tag{2.219}$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{n+1} (-1)^k \frac{e^{-i\omega t_{k+1}} - e^{-i\omega t_k}}{-i\omega} \tag{2.220}$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{n+1} (-1)^k \frac{e^{-i\omega t_k} - e^{-i\omega t_{k+1}}}{i\omega} \tag{2.221}$$

$$= \frac{1}{\sqrt{2\pi}} \frac{(1)(1 - e^{-\omega t_1}) + (-1)(e^{-\omega t_1} - e^{-\omega t_2}) + (1)(e^{-\omega t_2} + e^{-\omega t_3}) + \cdots}{i\omega} \tag{2.222}$$

$$= \frac{1}{\sqrt{2\pi}} \frac{1 + (-1)^{n+1} e^{-i\omega t} + 2\sum_{k=1}^{n} (-1)^k e^{-i\omega t_k}}{i\omega}. \tag{2.223}$$

Figure 2.1 shows a plot of the filter function for an example of these sequences. The plot shows that as $\omega \to 0$, the filter function almost vanishes, and the more we increase the number of pulses n, the more it approaches 0. Finally there is also the Concatenated Dynamical Decoupling (CDD) sequence [19] which is defined recursively by concatenating basic pulse sequences together to form higher-order sequences.

If the noise is white, then no matter how we design the pulses, the coherence

will not be suppressed. This is reflected by the fact that the PSD of the noise is a constant, and thus it will overlap with any filter function no matter how we choose it. This technique works only if the noise is coloured. This result can also be applied in reverse to learn the power spectrum of colored noise, which is referred to as quantum noise spectroscopy and will be discussed later.

A final important thing to note is that a qubit with short lifetime (as measured by the coherence for the free evolution) is not the end of the story. In fact, the PSD of the noise may be shaped in such a way that facilitates the design of the control pulses to maximize the coherence of the controlled evolution. Moreover, this controlled qubit will outperform another qubit suffering from noise with a PSD that makes designing the control difficult to implement experimentally (for example due to limited bandwidth of some electronics). Therefore, the lifetime measures of qubits (such as T_1 and T_2) do not provide the full story and should not be a base of comparison between different realizations of qubits. The assessment must take into consideration the PSD of the noise, and whether the experimental constraints can still support the optimal control design.

We can also derive the effective observable dynamics for this case. Defining $H_0 = \frac{1}{2}\Omega\sigma_z + f(t)\sigma_x$, and $H_1 = \frac{1}{2}\beta(t)\sigma_z$, and restricting the control to have even number of pulses, then at time $t = T$, we showed that the full evolution unitary is given by

$$U(T) = \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T (\Omega+\beta(\tau))y(\tau)d\tau} & 0 \\ 0 & e^{-i\frac{1}{2}\int_0^T (\Omega+\beta(\tau))y(\tau)d\tau} \end{pmatrix}. \tag{2.224}$$

Similarly, we can calculate the full control unitary (which for the given Hamiltonian would be the exact calculation if we set $\beta(t) = 0$)

$$U_0(T) = \mathcal{T}_+ e^{-i\int_0^t H_0(s)ds} \tag{2.225}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T \Omega y(\tau)d\tau} & 0 \\ 0 & e^{i\frac{1}{2}\int_0^T \Omega y(\tau)d\tau} \end{pmatrix}. \tag{2.226}$$

Thus, the interaction unitary can be calculated as

$$U_I(T) = U_0(T)^\dagger U(T) \tag{2.227}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T \beta(\tau)y(\tau)d\tau} & 0 \\ 0 & e^{i\frac{1}{2}\int_0^T \beta(\tau)y(\tau)d\tau} \end{pmatrix}, \tag{2.228}$$

and

$$\tilde{U}_I(T) = U_0(T)U_I(T)U_0^\dagger(T) \tag{2.229}$$

$$= \begin{pmatrix} e^{-i\frac{1}{2}\int_0^T \beta(\tau)y(\tau)d\tau} & 0 \\ 0 & e^{i\frac{1}{2}\int_0^T \beta(\tau)y(\tau)d\tau} \end{pmatrix}. \tag{2.230}$$

Now, we can evaluate the $V_O = \langle O^{-1}\tilde{U}_I(T)^\dagger O\tilde{U}_I(T)\rangle_c$ operators for $O \in \{X, Y, Z\}$.

$$V_X = \langle X\tilde{U}_I(T)^\dagger X\tilde{U}_I(T)\rangle_c \tag{2.231}$$

$$= \begin{pmatrix} \langle e^{-i\int_0^T \beta(\tau)y(\tau)d\tau}\rangle_c & 0 \\ 0 & \langle e^{i\int_0^T \beta(\tau)y(\tau)d\tau}\rangle_c \end{pmatrix} \tag{2.232}$$

$$= \begin{pmatrix} e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega} & 0 \\ 0 & e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega} \end{pmatrix}. \tag{2.233}$$

Similarly,

$$V_Y = \langle Y\tilde{U}_I(T)^\dagger Y\tilde{U}_I(T)\rangle_c \tag{2.234}$$

$$= \begin{pmatrix} e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega} & 0 \\ 0 & e^{-\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega} \end{pmatrix}. \tag{2.235}$$

Finally, by noticing that $\tilde{U}_I(T)$ commutes with Z, we get

$$V_Z = \langle Z\tilde{U}_I(T)^\dagger Z\tilde{U}_I(T)\rangle_c \tag{2.236}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.237}$$

It is clear that in the absence of noise all three operators are the identity. Moreover, they depend both on noise and control as discussed before. We were able to find a simple closed form as we assumed the control is formed of ideal pulses and that the noise is stationary zero-mean and Gaussian.

2.3.4 Numerical Simulation

The basic idea behind simulating the noisy qubit is to generate different realizations of the noise process, evaluate the Hamiltonian for each realization, simulate the time-ordered evolution to calculate the observables, and finally average over all realizations. As a result of the central limit theorem, the more noise realizations we average over, the more the sample average converges to the population average. This procedure is repeated for each input state and measurement operator.

There are three basic components in the simulator. The first is a function that calculates the time-ordered evolution of a Hamiltonian to generate a unitary. This is based on approximating the calculation using Equation 2.66. The second component is a simulator that generates random realizations of the noise given its power spectral

density (PSD). The algorithm consists of three steps. First, a random phase is added to each sample of the normalized desired PSD. Second, the complex-valued PSD is concatenated with a flipped version that is also complex conjugated. This step is done to ensure that the signal is symmetric around the middle (i.e the sample at $M/2$). Finally, we take the inverse Fourier transform of the signal and this will be real-valued as a result of the symmetry. We assume here that the desired PSD is single-side band, which means that the total power of the signal is obtained by integrating over positive frequencies only. The third component of the simulator is the main loop that simulates the quantum measurement. Inside the loop, we calculate the observables for each realization, and after that we average over all realizations. A pseudocode of the simulator implementation is shown in Algorithm 1.

2.4 Characterization of Quantum Systems

Characterization refers to the process of determining the parameters of some mathematical model of a quantum system given a set of experimentally acquired measurements. This is an important method of verification of engineered quantum systems. The relevant information about the system is encoded either through the state or through evolution dynamics. In the section, we describe few standard characterization techniques for finite-dimensional quantum systems.

2.4.1 Quantum State Tomography

The term "tomography" refers to the process of reconstructing images from projections. In the quantum setting, quantum tomography refers to the process of reconstructing mathematical descriptions of quantum systems using measurements given an infinite number of identical copies of the system. These descriptions can either be the quantum state of the system, the quantum channel representing the evolution of the system, or the quantum measurement representing the detection device. The most important problem is quantum state tomography, because the other two characterizations can be transformed and formulated as a quantum state tomography problem. In the literature, the term quantum state tomography can also refer to the process of estimating the state of a continuous-variable quantum system. If the system is discrete the problem is commonly known as quantum state estimation and can be formulated as follows. Given a set of measurements $\{m_i\}_{i=1}^{K}$ obtained by measuring an unknown state ρ, using the operators $\{X_i\}_{i=1}^{K}$, it is required to find an estimate for that quantum state. The basic assumption is that we have an infinite copies of the unknown state, so that we can perform any required

Algorithm 1 Monte Carlo simulation of a noisy qubit

function EVOLVE(H, δ)
  $U \leftarrow I$
  for $t \leftarrow 0, M-1$ **do**
    $U_t \leftarrow e^{-iH_t\delta}$
    $U \leftarrow U_t U$
  end for
  return U
end function
function GENERATENOISE(S, T, M)
  $N \leftarrow \frac{M}{2}$
  for $j \leftarrow 0, N-1$ **do**
    $\phi \leftarrow \text{Random}(0,1)$
    $P_j \leftarrow \frac{M}{\sqrt{T}}\sqrt{S_j}e^{2\pi i\phi}$
    $Q_{N-j} \leftarrow \bar{P}$
  end for
  $P \leftarrow$ CONCATENATE(P, Q)
  $\beta \leftarrow \text{Re}\{\text{ifft}(P)\}$
  return P
end function
function SIMULATE(ρ, O, T, M, f_x, f_y, f_x, S_X, S_Y, S_Z)
  $\delta \leftarrow \frac{T}{M}$
  $E \leftarrow 0$
  for $k \leftarrow 0, K-1$ **do**
    $\beta_x \leftarrow$ GENERATENOISE(S_X, T, M)
    $\beta_y \leftarrow$ GENERATENOISE(S_Y, T, M)
    $\beta_z \leftarrow$ GENERATENOISE(S_Z, T, M)
    for $j \leftarrow 0, M-1$ **do**
      $t \leftarrow (0.5+j)\delta$
      $H_j \leftarrow \frac{1}{2}\left(\Omega+\beta_z(t)\right)\sigma_z + \frac{1}{2}\left(f_x(t)+\beta_x(t)\right)\sigma_x + \frac{1}{2}\left(f_y(t)+\beta_y(t)\right)\sigma_y$
    end for
    $U \leftarrow$ EVOLVE(H, δ)
    $E \leftarrow E + \text{tr}\left[U\rho U^\dagger O\right])$
  end for
  $E \leftarrow \frac{E}{K}$
  return E
end function

number set of measurements. The straightforward approach to solving this problem is to realize that the problem constitutes a system of linear equations in the form

$$\text{tr}(\hat{\rho}X_i) = m_i, \tag{2.238}$$

where $\hat{\rho}$ is the required estimate, and is the only unknown. Thus, upon inverting the system, we can obtain the estimate. The problem with this approach is that the obtained estimate is not guaranteed to represent a physical state. In other words,

the eigenvalues can be outside the allowed range. Consequently, other approaches are needed to solve this problem. The most common methods are the maximum likelihood and the least-squares methods which will be introduced in the next section.

Another related issue is how to choose the set of measurements in order to estimate the state. This is an interesting information-theoretic question. For example, considering a qubit: if we only measure two of the Pauli operators, we cannot uniquely determine the state till we measure the third Pauli operator. Another way to investigate this question is to consider the space $D(S)$ of density operators defined for the system S. Each point in this space is a valid state of the system. If we choose a basis for this space, we can uniquely describe each point in terms of the basis. For a system of n qubits, the set of Pauli operators on local qubits (i.e. $\{\bigotimes_{i=1}^{n} X_i\}$, where $X_i \in \{I, X, Y, Z\}$) form an orthonormal basis with respect to the Hilbert-Schmidt inner product $\langle A, B\rangle = \mathrm{tr}(A^\dagger B)$. For a general quantum systems of dimension d, we can define a generalization of the Pauli orthogonal basis $\{u_{jk}, v_{jk}, w_l\}$ for $1 \leq j \leq k \leq d$, and $1 \leq l \leq d-1$ where

$$u_{jk} = |j\rangle\langle k| + |k\rangle\langle j| \tag{2.239}$$

$$v_{jk} = -i(|j\rangle\langle k| - |k\rangle\langle j|) \tag{2.240}$$

$$w_l = \sqrt{\frac{2}{l(l+1)}}\left(-l\,|l+1\rangle\langle l+1| + \sum_{j=1}^{l} |j\rangle\langle j|\right). \tag{2.241}$$

This set consists of $d^2 - 1$ traceless Hermitian matrices which reduce to the usual Pauli matrices for $d = 2$, and the Gell-Mann matrices for $d = 3$. Experimentally, we might choose other basis for measurements that might be easier to implement. Also, in many cases due to noise effects, we might choose to perform measurements using an overcomplete basis to boost the robustness of the estimation procedure. Now, this represents a major problem for systems with large number of qubits because in this case $d = 2^n$ where n is the number of qubits. The reason quantum computing is more powerful than classical computing is the very same reason that makes the full characterization of a large system very difficult or even impossible. At the end, the whole idea of tomography is to write a classical description of a quantum state and this definitely will require exponential resources to do. Therefore, tomography is only used for characterizing small quantum systems. For large systems, this is a currently a major challenge and entirely different characterization techniques are needed. This doesn't decrease the importance of tomography as it is typically used as first verification procedure for individual qubits as well as single and two-qubit gates.

2.4.1.1 Overview on the diluted maximum likelihood method

In this section an overview on the diluted maximum likelihood [20] is given. The maximum likelihood method of quantum estimation is based on trying to find the state $\hat{\rho}$ that maximizes the log-likelihood function

$$\log(\mathcal{L}) = \sum_j f_j \log(\mathrm{tr}(\hat{\rho}\Pi_j)), \tag{2.242}$$

where f_j is the relative frequency of outcome j described by the POVM set $\{\Pi_j\}$. This can be achieved by doing iterations in the form

$$\hat{\rho}_{k+1} = R\hat{\rho}_k R, \tag{2.243}$$

where

$$R = \sum_j \frac{f_j}{\mathrm{tr}(\hat{\rho}\Pi_j)}\Pi_j. \tag{2.244}$$

This is the $R\rho R$ algorithm. However, there is no guarantee that this form of update equation will generally converge. So, a modification on the form of the update equation is done to become,

$$\hat{\rho}_{t+1} = \frac{(I+\epsilon R)\hat{\rho}_t(I+\epsilon R)}{\mathrm{tr}\left((I+\epsilon R)\hat{\rho}_t(I+\epsilon R)\right)}. \tag{2.245}$$

This is called the diluted maximum likelihood because it "dilutes" R by mixing it with the identity operator I. The step parameter ϵ can be chosen arbitrarily and can be constant or adaptive. When $\epsilon \to \infty$, the iterations revert back to the $R\rho R$ form. Choosing a value of $\epsilon \ll 1$ ensures that after each iteration the likelihood function is non-decreasing. On the other hand, a higher value of ϵ is better in terms of speed of convergence. A fewer number of iterations is required to achieve the same accuracy compared to a low value of ϵ.

An important thing to notice is that Pauli operators $\{X^{(i)}\}_{i=1}^{d^2-1}$ do not form a set of POVM. But fortunately it is easy to construct a set of POVM out of the Pauli operators, by taking all the "up/down" projectors normalized. So, the POVM set becomes $\{\frac{1}{d^2-1}\Pi_\uparrow^{(i)}, \frac{1}{d^2-1}\Pi_\downarrow^{(i)}\}_{i=1}^{d^2-1}$.

Finally, maximum likelihood is a batch algorithm. So, the iterations are repeatedly run on a set of data. To modify this algorithm to become online, the iterations must be performed on the dataset after each new data point obtained.

2.4.1.2 Overview on the least-squares method

This section gives a short brief on the least-squares method for quantum estimation. More details are given in [21]. The basic idea is to construct a parametric model of the state in the form

$$\rho = \frac{I}{d} + \sum_{i=1}^{d^2-1} \theta_i U_i, \tag{2.246}$$

where the set $\{U_i\}_{i=1}^{d^2-1}$ are some Hermitian basis, and $\theta_i = \mathrm{tr}(\rho U_i)$ are the parameters. The problem of quantum tomography then becomes trying to estimate this parameter vector given the measurement dataset. So given a set of measurement operators represented using their "up/down" projectors in the form $\{\Pi_j\}_{j=1}^{2d^2-2} = \{\Pi_\uparrow^{(i)}, \Pi_\downarrow^{(i)}\}_{i=1}^{d^2-1}$, the associated probabilities are

$$p_j = \mathrm{tr}(\Pi_j \rho) = \frac{1}{d} + \sum_{i=1}^{d^2-1} \theta_i \,\mathrm{tr}(\Pi_j U_i). \tag{2.247}$$

These probabilities can be obtained experimentally but with some errors due to performing finite number of shots. The noisy data is denoted by $\hat{p}_j$. By defining the matrix expansion of the projectors in terms of the chosen basis $X_{i,j} = \mathrm{tr}(\Pi_i U_j)$, the dependent vector with components $Y_j = \hat{p}_j - \frac{1}{d}$, the model can be rewritten as

$$Y = X\theta + e, \tag{2.248}$$

where e is the error vector which converges to a normal distributed random variable at the limit of very large number of measurements. The optimal parameter is defined to minimize the sum of squared errors as

$$\hat{\theta}_{LS} = \mathrm{argmin}_\theta (Y - X\theta)^T (Y - X\theta), \tag{2.249}$$

and the solution of this optimization problem is

$$\hat{\theta}_{LS} = (X^T X)^{-1} X^T Y. \tag{2.250}$$

After the estimation of the unknown parameter, the quantum state is reconstructed as

$$\hat{\rho} = \frac{I}{d} + \sum_{i=1}^{d^2-1} \hat{\theta}_i U_i. \tag{2.251}$$

The reconstructed state might be generally unphysical due to non-positivity of the estimate. So, in this case the state must be projected back to the physical space. One way to do is to redistribute the negative eigenvalues over all other eigenvalues, until there are no more negative eigenvalues. It can be shown [22] that this is an optimal projection method, in the sense that the projected state is nearest to the unphysical state in terms of the Frobenius norm. This method is batch, but can be adapted to become online by doing the whole procedure of estimation and projection after each new data point obtained.

2.4.2 Quantum Process Tomography

Quantum Process Tomography (QPT) refers to the procedure by which an estimate of the quantum channel affecting the quantum system is obtained from a set of measured data. The procedure is to apply a set of input states $\{\rho_i\}_{i=1}^{K}$, and estimate the output states $\{\mathcal{E}(\rho_i)\}_{i=1}^{K}$. Then utilizing the Choi-Jamiolkowski isomorphism for a CPTMP map discussed earlier, the problem can be formulated as estimating the $d^2 \times d^2$ positivize matrix J such that

$$\mathcal{E}(\rho_i) = \mathrm{tr}_{S'} \left(J(I \otimes \rho_i^{\intercal}) \right). \tag{2.252}$$

Similar estimation methods could be used, with the extra trace-preserving constraint must be satisfied for the estimate besides positivity. Now, it is impossible to directly measure $\mathcal{E}(\rho_i)$, and so quantum state estimation procedure has to be performed for each input state. So even for few qubits, the overall procedure becomes too complex and might be infeasible. The main application of QPT is characterizing quantum gates to verify their operation, and this becomes a major challenge for large systems.

2.4.3 Quantum Noise Spectroscopy

Quantum Noise Spectroscopy (QNS) is a procedure that targets to obtain an estimate of the PSD of the noise affecting a quantum system. The idea is to the quantum system itself as a sensor (probe) to study the environment. Going back to Equation 2.213, we see that there are three quantities there. The coherence $W(T)$, which as discussed before can be easily measured experimentally, the PSD of the noise which in this application is assumed to be unknown, and finally the filter function which is known given a particular control sequence. So, this can be used to actually estimate the PSD of the noise. One way is to use the Alvarez-Suter (AS) method for QNS [23], which is based on discretizing the overlap integral, approximating the filter function as a series of impulses, and neglecting harmonics beyond certain point n_p. In this case can express the logarithm of the measured coherence

as

$$\log W(t) = -\frac{1}{2}\int_{-\infty}^{\infty} S(\omega)|Y(\omega)|^2 d\omega \tag{2.253}$$

$$\approx \sum_{k=1}^{n_p} S_k A_k, \tag{2.254}$$

where A_k can be expressed in terms of the filter function. Now, if we choose a set of DD sequences and for each sequence we measure the coherence experimentally, then we can form a system of linear equations, which can be inverted to find the vector S_k representing the sampled PSD of the noise. One way to choose this set is to fix the type of DD sequence (for example CPMG), and each element in the set would correspond to one choice of the number of pulses n. This method is useful; however it suffers from multiple drawbacks. First, it depends on many approximations and assumptions on the control pulses that allow us to simplify the overlap integral as a discrete finite sum. This might not be valid for all cases (for example if the pulses are not ideal and have finite width). Second, the overlap integral formula itself is based on assumptions on the noise (stationary, zero-mean, and Gaussian). In a practical situation where we are studying experimentally a new quantum system, in the absence of any prior information about the noise, these assumptions are not justified. Finally, because we are fixing the total evolution time T, there is a limit on the resolution of the estimated PSD. So, probing low frequency components would require long sequences. Additionally, if the control pulses have finite width (which is the practical situation), then there will be an upper limit on how many pulses can be included in one sequence, resulting in an upper limit on the largest frequency component that can be estimated.

2.4.4 Other Techniques

There exist many other techniques in the literature for characterizing quantum systems in a fast growing field known as Quantum Characterization, Verification, and Verification (QCVV). We mention here two examples. The first is Hamiltonian learning [24–27]. In this case, the assumption is that the system evolves under a Hamiltonian (and thus the system is closed). This reduces the complexity of QPT as the goal is now estimating a $d \times d$ matrix rather than a $d^2 \times d^2$ matrix. The second is Gate Set Tomography (GST) [28,29]. This is a characterization technique that takes into consideration State Preparation and Measurement (SPAM) errors. The problem in standard QST and QPT is that we assume that the measurement operators or the input states we perform experimentally are exact with no errors. However, this might not be the case (for example there might be an over-rotations when ap-

plying Pauli operators). GST solves this problem by applying a set of sequences of gates. So, for example if we apply a Pauli X twice on a state, it should have no effect ($X^2 = I$). But, if we find a different state, then there exist over-rotations. So GST aims to find the kind of sequences that are needed to discover such errors and then given the experimental measurements have an estimate for the actual state and operations happening in the quantum system. The main drawback of this method is that it is computationally very intensive, and thus becomes challenging to apply even for a system of few qubits.

2.5 Quantum Control

In this section, we give a brief overview on quantum control. We start with the basic definition of the problem. Next, we discuss the typical control targets as well as the common constraints that occur in actual experiments. After that we discuss controllability of quantum systems using the Lie Algebra criteria, followed by a discussion on the different control architectures. Finally, we end the section with an overview on GRAPE, one of the most famous numerical quantum control algorithms.

The Hamiltonian of any quantum system can be written in the general form

$$H(t) = H_0(t) + \sum_{k=1}^{N} \alpha_k(t) H_k, \tag{2.255}$$

where $H_0(t)$ is called the drifting Hamiltonian, the terms H_k are called the control Hamiltonian, and $\alpha_k(t)$ are the control pulses. The drifting Hamiltonian represents the free evolution of the system, possibly under the influence of static forces. The control Hamiltonian part represents external dynamical forces that can be "controlled" in a way to modify the dynamics of the system. Given a particular quantum system, there will be a set of allowed control Hamiltonians that are experimentally possibly to implement depending on the system. Quantum control then refers to the problem of finding the set of control pulses given the drifting and control Hamiltonians such that a particular target objective is achieved. The most common objectives are explained as follows.

2.5.1 Targets and Constraints

1. Target state:
   It is desired that a system starting at state $\rho(0)$ to evolve to some target state ρ_T, at time $t = T$. This target could be relaxed to require only reaching a state that is close enough to the target state using some distance measure.

This can be expressed mathematically as $F(\rho(T), \rho_T) \leq \epsilon$, where $\rho(T)$ is the evolved state, ϵ is some target distance (level of accuracy), and F is a distance measure such as fidelity.

2. Target gate:
   It is required that a system evolves according to some target unitary U_T. This means that whatever the initial state of the system is, we require that the final state becomes $\rho(T) = U_T \rho(0) U_T^\dagger, \forall \rho(0)$. This problem is generally more difficult than the first problem in the following non-formal sense. To evolve from a particular state to a target state, there is an infinite number of paths connecting both states and any of them is a valid solution. However, in the second problem the requirement is to transform every possible state the same way, so it is more constrained in this case. Similarly, in this problem one could relax the requirement to be reaching a unitary that is close enough to the target unitary using some distance measure.

3. Decoherence suppression:
   The aforementioned targets assume that the system is closed. If the system is open (which is the case for most implementations), it becomes much more complicated. In this case, the effects of decoherence need to be suppressed first. As discussed previously, DD sequences can be designed to do this if we have enough information about the noise. This is a target on its own with the typical application of a quantum memory. The state of the system at time $t = 0$ is required to stay unchanged until time $t = T$. This is also equivalent to implementing the identity gate $U_T = I$. In practical systems, such a "pure" unitary evolution might not be possible, because the impossibility of completely eliminating the noise (by the very definition of an open system being in interaction with the environment). Thus, it is more practical that the target is relaxed to become in the form $F(\rho(T), \rho(0)) \leq \epsilon$.

4. Decoherence suppression and a target gate:
   This is the most general target for an open quantum system. Besides targeting to minimize the decoherence, it is required simultaneously to implement a target quantum gate. This is more challenging because it becomes a multi-object optimization problem. It can be the case that the optimal pulses that minimizes decoherence are not optimal for implementing the gate, and vice versa. There can also be a trade-off on how much each target it realized.

In general, the control problem is a contained optimization problem. The constraints come from the experimental capabilities (hardware specifications) that are available to a particular system. The most common constraints are listed as follows.

1. The pulse shape:
   In many experiments, it is only possible to generate a particular type of waveforms such as Gaussian pulses or square pulses. Therefore, the solution of the quantum control problem must be of the same pulse shape. Moreover, the pulses need to be physical. For example, perfect Dirac delta pulses that usually occur in designing DD sequences, are impossible to implement physically. We can only approximate the impulse by a narrow Gaussian or square pulse.

2. Limited pulse power:
   In many experimental situations, there is a limitation on the maximum amplitude of the control pulse that can be applied. This can be due to hardware specifications making it hard to generate pulses with arbitrary high amplitudes. The other situation is where the system itself might break if a very high amplitude pulse is applied (for example, extremely high voltages can breakdown integrated photonic circuits).

3. Limited bandwidth:
   There can be additionally limitations on the bandwidths of control signals that can be generated in experiments. This might be due to the specifications of the available waveform generator, or the control pulses will be upconverted to a higher frequency band. In that situation, there will be a limitation on the bandwidth due to the specifications of the modulation system.

4. The Evolution time:
   It might be required to perform the quantum control in the shortest possible time. Doing this allows implementing more quantum gates in the same time interval which means increasing the overall execution speed of the quantum algorithm. Also, this might be advantageous for noisy quantum systems, because the longer the evolution time, the more the qubits will decohere. When the evolution time is to be minimized as well, the problem is referred to as time-optimal quantum control. In general, there is a limitation on the shortest possible evolution time known as the quantum speed limit.

2.5.2 Controllability

Before attempting to design the control pulses, we need to make sure that it is actually possible to achieve the target. For instance, consider a qubit undergoing a closed evolution with both the drift and control Hamiltonians in the same directions (say along z-axis). In this case, it is impossible to implement a Pauli-X gate, because there is no term in the Hamiltonian that results in a X gate upon time evolution. But, if we also have one other controllable direction (say y-axis), then in that case,

we can design a control pulse sequence to implement the X gate. This problem is referred to as controllability which aims to find whether a particular quantum state or quantum gate can be implemented given the drift and control Hamiltonians. For the simple case of a qubit undergoing closed evolution, it is easy to answer the question just by thinking geometrically about the Bloch sphere representation. However, for more complicated systems (such as higher-dimensional systems (qudits), multi-qubit systems, spin chains and lattices, etc.), we need a systematic way of answering the question. It turns out the answer is known for the case of closed finite-dimensional quantum systems with arbitrary control. We will give here a brief overview focusing on the case where the control objective is a target unitary. The idea is to look into the reachable set of unitaries starting from the identity (at $t = 0$, $U(0, 0) = I$). This is the set of all unitaries we can reach given the drift and control Hamiltonians. It is easier to study the problem with Hamiltonians (at the end any unitary can be expressed in terms of some Hamiltonian in the form $U = e^{-iH}$). Denoting the set of reachable Hamiltonians by $\mathcal{R}$, it turns out we have the following:

1. $H_0 + \sum_{k=1}^{N} \lambda_k H_k \in \mathcal{R}$
   This is because we can choose each of the control pulses $\alpha_k(t) = \lambda_k$.

2. $\forall \lambda > 0 : H \in \mathcal{R} \implies \lambda H \in \mathcal{R}$
   This can be achieved simply by rescaling the evolution time by λ.

3. $I \in \mathcal{R}$
   This follows from the quantum recurrence theorem which is a generalization of Poincaré recurrence theorem. The theorem states that given a sufficiently large but finite time, we can always reach the initial state of the system arbitrarily close. In other words, for a Hamiltonian H, and some matrix distance $\|\cdot\|$,

$$\forall T_0 > 0, \forall \epsilon > 0, \exists T > T_0 : \left\| e^{-iHT} - I \right\| < \epsilon. \tag{2.256}$$

   This holds for any finite-dimensional quantum system. The rigorous proof of the theorem depends on the fact that the set of unitaries $\mathcal{U}(d)$ of dimension $d < \infty$ is a compact space (closed and bounded), and thus any sequence in this space will have have a convergent subsequence with the limit in the space. So, if we define a sequence $U_n := e^{-iHn}$, then we can find a convergent subsequence $U_{n(k)} \to U$. We can then write the limit

$$\lim_{k\to\infty} U_{n(k+T)} U_{n(k)}^{\dagger} = \lim_{k\to\infty} U_{n(k+T)} \lim_{k\to\infty} U_{n(k)}^{\dagger} \tag{2.257}$$

$$= UU^{\dagger} \tag{2.258}$$

$$= I. \tag{2.259}$$

Equivalently from our definition of the sequence,

$$\lim_{k\to\infty} e^{-iH(n(k+T)-n(k))} = I, \tag{2.260}$$

and so, by finally defining $T_0 = n(k+T) - n(k) \geq T$, the proof is complete.

4. $H \in \mathcal{R} \implies -H \in \mathcal{R}$
   This also follows from the recurrence theorem. By observing that there exist a time τ such that $e^{-iH\tau} \approx I$. If we stop the evolution before that time particularly at $\tau - 1$, we find that $e^{-iH(\tau-1)} = e^{-iH\tau}e^{-i(-H)} \approx e^{-i(-H)}$. This implies that it is possible to obtain the evolution $e^{-i(-H)}$, and thus $-H \in \mathcal{R}$.

5. $H_1, H_2 \in \mathcal{R} \implies (H_1 + H_2) \in \mathcal{R}$
   This follows from the Lie-Trotter formula

$$\lim_{n\to\infty} \left(e^{-iH_1/n}e^{-iH_2/n}\right)^n = e^{-i(H_1+H_2)} \tag{2.261}$$

   This means that if we apply whatever control sequence needed to implement H_2 for a very short time followed by that needed for realizing H_1 also for very short time, and we then we repeat alternating between both controls for a very long time, effectively we implement the average of both Hamiltonians.

6. $H_1, H_2 \in \mathcal{R} \implies [H_1, H_2] \in \mathcal{R}$
   This follows from the commutator formula

$$\lim_{n\to\infty} \left(e^{-iH_1/n}e^{-iH_2/n}e^{iH_1/n}e^{iH_2/n}\right)^n = e^{-i[H_1,H_2]} \tag{2.262}$$

   This means that alternating between the control sequence to realize each of $-H_2$, $-H_1$, H_2, H_1 each for a very short amount of time, and repeating for a very long time can effectively implement the commutator $[H_1, H_2]$.

Now, properties 1 – 5, implies the linear real span of the Hamiltonians $\{H_0, H_k\}$ is in the reachable set. This implies that the reachable set must be closed under real linear combinations and commutation operation. This is exactly the definition of a Lie Algebra. In other words, the reachable set of Hamiltonians is the Lie Algebra generated by the drift and control Hamiltonians. The lie group corresponding to the Lie Algebra, represents the reachable set of unitaries. Therefore, the conclusion is that if the target Hamiltonian is in the Lie Algebra, then theoretically we can reach this target with arbitrary precision. Now, if we want to make sure that every possible quantum gate is achievable, then we need to ensure that any arbitrary set of basis for the space of Hamiltonians is a subset of the Lie Algebra. If this is the

case, then by linearity of quantum mechanics, we can achieve any target because it can be expressed as a linear combination of the basis.

There are two important points to mention in this context. First, this discussion is only theoretical in the sense that it does not provide a practical way of finding the control pulses. In other words, the guarantees mentioned 1 – 6 are asymptotic, and thus it might not be practical to utilize these arguments (such as Trotterization,...etc.) for pulse design. The second point is that under experimental control constraints, it might not be possible to achieve the target even though a controllability study shows it is possible. For example, we might need a pulse with very high power that exceeds the limits of a quantum device. In that case, there will be a limit on the accuracy of the target. Finally, if the system is open or infinite-dimensional, then the problem becomes more challenging and is a current subject of research.

We end this discussion with two examples. The first is a single qubit with Hamiltonian $H = \sigma_z + f(t)\sigma_x$. We find that $[\sigma_z, \sigma_x] = 2i\sigma_y$. This means that the set $\{\sigma_x, \sigma_y, \sigma_z\} \subset \mathcal{R}$. But, we know that the set of Pauli's form a basis for the space of 2×2 Hermitian matrices, and thus any target Hamiltonian (and consequently any quantum gate) can be implemented. The interesting observation is that we only need one control pulse and rely on the fact that the qubit is naturally drifting. In practice, relying on one control pulse, would probably require a longer evolution time to achieve the target, and generally having more controls is better. On the other hand, in many systems, having more controls can be challenging. For example, superconducting qubits need to operate at cryogenic temperatures inside a dilution fridge. However, the waveform generators and other control hardware are located outside the fridge, and they are connected to the chip inside the fridge through cables. The size of the fridge is limited, and thus the presence of a large number of cables inside the fridge can become impossible simply because of the the lack of space.

The second example is for a two-qubit system with Hamiltonian

$$H = \sigma_z \otimes I + I \otimes \sigma_z + f_{x1}(t)\sigma_x \otimes I + f_{1x}(t) I \otimes \sigma_x + f_{xx}(t)\sigma_x \otimes \sigma_x. \tag{2.263}$$

This Hamiltonian represent a drifting along the Z-axis for each of the two qubits, a local control on each qubit along X-axis, and a controllable coupling term between the two qubits. If we calculate all possible commutators of these terms, including commutators with the intermediate results, we can obtain the set $\{\sigma_i \otimes \sigma_j\}$ for all possible Pauli's including the identity, which spans the space of 4×4 Hermitian matrices. This is a known result also in quantum computation, where any two-qubit gate can be decomposed into local Pauli gates and one entangling gate.

2.5.3 Types of Quantum Control

Next, we will discuss the different architectures of quantum control systems. We can categorize them in two different ways as follows.

2.5.3.1 Open-loop versus closed-loop control

Open-loop refers to the situation where the control pulses are designed beforehand and then applied to the quantum system during operation. This requires the knowledge of the dynamical model in order to be able to design the pulses. If the model is not accurate enough, then the designed pulses will not perform well. Once the pulses are designed over the time interval $[0, T]$, they are fixed and directly applied to the system. On the other hand, closed-loop control refers to the case where the control pulses are designed autonomously during operation through a feedback mechanism. In this scenario, continuous quantum measurements are performed and fed through a controller which evaluates the control pulses that are feedbacked to the quantum system. This does not require full knowledge of the model since the feedback mechanism can compensate for it. The task then becomes how to design the controller such that the control target is achieved. As a result, the controller becomes itself part of the system, and affects its evolution. Continuously monitoring the quantum system will also affect the evolution, and thus the dynamics have to be expressed including both the controller and the measurements. This becomes more challenging, and usually requires a different set of mathematical tools. Particularly, we know that quantum measurements are inherently stochastic. This means that observing certain outcome will determine the post-measured state. As a result, the usual description of the evolution using Schrödinger's equation is not suitable, and usually a stochastic differential equation is used [30]. This requires assumptions on the way measurement process such as the statistical distribution of the outcomes, as well as the Markovianity assumption [31]. This means that at any point in time, the measurement outcome is used instantaneously by the controller to find the value of the pulse at that moment, and then the outcome is thrown away without any storage. The problem with this assumption is that practical controllers have a non-flat frequency response due to the fact that any actual hardware (even as simple as an RC circuit) cannot act as an ideal filter. So, the controller will always keep memory of some previous measurement and its output can only be expressed as a convolution integral. This makes it impossible to write a stochastic master equation, and the whole situation become much more challenging in this case. Moreover, because of the way the dynamics of a quantum system work, the whole problem can be considered a non-linear control problem, which is generally hard to study and solve. The reason that most classical engineering feedback systems work perfectly,

is that the practical systems are either linear, or can be linearised at some operating point. In quantum mechanics however, the evolution itself is non-linear in the control pulses, and is not possible to linearise. Besides the difficulty of modelling and designing feedback control in the quantum case, the actual implementation may also be challenging. The controller should operate at the same speed as the quantum system so that it can keep track of the evolution. In many systems, there will be a gap between the fastest possible available electronics and the time scales of the quantum system. As a result, quantum feedback control is much more challenging and usually open-loop is preferred whenever possible.

2.5.3.2 Online versus offline control

This is another way of classifying quantum control architectures. Online control refers to the case where the control pulses are designed based on probing the actual quantum system. In other words, we directly access the system to get essential information to design the pulses, and we do not use a model for the system. Offline quantum control refers to the case where a computational model for the actual quantum system is utilized to design the control pulses in place of the actual quantum system. The parameters of the model are chosen based on an experimental characterization of the system. Offline methods are usually preferred experimentally to minimize the interaction with the quantum system. However, it is not possible for large quantum systems because simulating such systems can quickly exceed the limits of classical computation. Now, this categorization is independent on whether we use open-loop or closed-loop control. For example, the GRAPE algorithm (will be discussed later), is an open-loop technique. We obtain the pulses without having to continuously monitor the system physically and feeding the measurement outcome to a controller that is also physically connected to the system in feedback. We just use numerical optimization to design the pulses. Now, this technique can be implemented offline in which the system is simulated on a classical computer and used with a optimizer. It can also be implemented online; in which case the numerical optimizer will have access to the actual physical system to probe it whenever needed. Another example is an atom in a cavity interacting with an electromagnetic field in feedback mode. In this case, the field is applied to the cavity as an input, and due to interaction with atom inside the cavity, the output field emerging from the cavity will change (for example the number of photons can change due to absorption/emission by the atom). This output field can then be processed by a controller (for example through homodyne detection or photon counting) and then feedbacked again to the cavity. This is a closed-loop control system. There is continuous monitoring of the atom through the measurement on the fields. The controller can also be tuned online or offline. In the offline case, a computer simulation of

the atom–field system can be used to tune the controller. In the online mode, the controller is connected to the actual system, and tuned in place.

2.5.4 Overview on GRAPE

In this section, we give an overview on one of the most famous and widely used numerical algorithms for quantum control, which is Gradient Ascent Pulse Engineering (GRAPE) [32]. This is an open-loop numerical technique that can be used to design control pulses under experimental constraints. It is based on three ideas. The first idea is to discretize the control pulses over the evolution duration into short time steps, such that the evolution during each time interval can be approximated by a simple matrix exponential of the Hamiltonian rather than a time-ordered exponential. This requires understanding the time scales of the system to ensure no variations are happening during one time step. In classical control, this is usually referred to as bang-bang control when the pulses are piecewise constant. This facilitates the design process, because now we have a parametrization of the control pulses, and so they can be expressed as a real high-dimensional vector rather than a continuous function. This makes numerical optimization much easier, and standard tools such as gradient ascent can be used. Moreover, constraints such as limited amplitudes can be taken into consideration very easily because it translates directly bounds on the amplitude of the pulses at each time step. In other words, the search space becomes compact. The second idea is expressing the target as a cost function including global constraints (such as total power or bandwidth of the control signal). For state transfer this would be simply the fidelity between the target state and the actual state, and for gate implementation this would be the fidelity between the target unitary and the actual evolution. The final and the most remarkable idea is calculating the gradient of the cost function with respect to the pulse amplitudes at each time step in a closed-form. The algorithm can then proceed as any gradient-based approach. We start with a set of random amplitudes for each time step. Next, we calculate the gradient of the cost function, and finally we update the amplitudes using gradient ascent. Now, we will show an example of this approach for control to achieve a target state ρ_T. Going back to Equation 2.255, if we discretize the evolution duration $[0, T]$ into M time steps of length $\Delta t = T/M$, denoting the amplitude of k^{th} control pulse at time step j by $\alpha_k(j)$, starting from state $\rho(0)$ we can write the evolved state at time $t = T$ as

$$\rho(T) = U_M \cdots U_1 \rho(0) U_1^\dagger \cdots U_M^\dagger, \tag{2.264}$$

where each propagator is defined as

$$U_j = e^{-i\Delta t\left(H_0+\sum_{k=1}^{N}\alpha_k(j)H_k\right)}. \tag{2.265}$$

The cost function can be written as

$$\begin{aligned}
\Phi_0 &= \langle\rho_T|\rho(T)\rangle && (2.266)\\
&= \langle\rho_T|U_M\cdots U_1\rho(0)U_1^\dagger\cdots U_M^\dagger\rangle && (2.267)\\
&= \langle U_{j+1}^\dagger\cdots U_M^\dagger\rho_T U_M\cdots U_{j+1}|U_j\cdots U_1\rho(0)U_1^\dagger\cdots U_j^\dagger\rangle && (2.268)\\
&:= \langle\lambda_j|\rho_j\rangle\,. && (2.269)
\end{aligned}$$

The third line follows from the invariance of the trace under cyclic permutations, ρ_j is the evolved state at time step j due to forward propagation starting at time $t = 0$, and λ_j is the backward propagated target state from time $t = T$. It can be shown that up to first order in δt, the gradient of the cost function can be written compactly as

$$\frac{\delta\Phi_0}{\delta\alpha_k(j)} = -\langle\lambda_j|i\Delta t[H_k,\rho_j]\rangle\,. \tag{2.270}$$

So finally, the update equation for the control pulses would be

$$\alpha_k(j)\to\alpha_k(j)+\eta\frac{\delta\Phi_0}{\delta\alpha_k(j)}, \tag{2.271}$$

where η is the step size for the gradient ascent. If we repeat the iterations for enough number, making sure Δt is small enough, then we can find the optimal pulse sequence. It is important to notice that due to using gradient ascent, the obtained solution is a local maximum with no guarantee that is global maximum. Also, we need to choose the step size carefully to ensure convergence at a reasonable rate. Finally, including amplitude constraints can be implemented by replacing the updated pulse amplitude by the maximum or the minimum allowed value whenever the update results in an out-of-range value.

There are many possible extensions and modifications to this idea. For instance, the update rule can be modified to include higher-order gradients as well as momentum-terms [33, 34]. The step size can also be chosen adaptively. Another related algorithm is the Chopped Random Basis (CRAB) [35] which works similarly but the control signal is represented in some basis (such as the Fourier basis), truncated to a maximum number of components (harmonics in case of Fourier basis). This makes the optimization procedure more efficient. Randomness is introduced to the basis in order to enhance the convergence. Finally, with the recent development

of auto-differentiation and numerical packages such as Tensorflow, similar control algorithms have been developed [36–38].

2.6 Conclusion

In this chapter, we gave an overview on three main aspects of studying quantum systems. These are also consecutive steps in the design flow of any engineered quantum system. The first aspect is how to mathematically model a quantum system in the presence of noise. We introduced three methods: modelling noise as a quantum channel, Lindblad master equation, and effective observable dynamics. We applied these three methods to model a qubit undergoing dephasing due to interaction with a classical environment. The second aspect is characterization of a quantum system given experimental measurements. This can be used to characterize quantum states, processes, and noise. Finally, we gave an overview on quantum control, discussing the problem setting, controllability of finite-dimensional closed quantum systems and an overview on GRAPE as a numerical optimal control method. In the next chapter, we are going to introduce a novel method to characterize quantum states efficiently inspired by a classical machine learning algorithm.

Chapter 3

Efficient Online Quantum State Estimation

Abstract In this chapter, we explore an efficient online algorithm for quantum state estimation based on a matrix-exponentiated gradient method previously used in the context of machine learning. The state update is governed by a learning rate that determines how much weight is given to the new measurement results obtained in each step. We show convergence of the running state estimate in probability to the true state for both noiseless and noisy measurements. We find that in the latter case the learning rate has to be chosen adaptively and decreasing to guarantee convergence beyond the noise threshold. As a practical alternative we then propose to use running averages of the measurement statistics and a constant learning rate to overcome the noise problem. The proposed algorithm is numerically compared with batch maximum-likelihood and least-squares estimators. The results show a superior performance of the new algorithm in terms of accuracy and runtime complexity.

3.1 Introduction

The field of quantum information processing has grown rapidly over the past decade, largely motivated by the wide range of prospective applications of quantum computing, quantum cryptography, and quantum communications. However, building scalable quantum devices is still an enormous challenge. A core unsolved problem is the efficient characterization of quantum systems of intermediate size—can we check efficiently whether a quantum device comprised of a few qubits performs as intended? Practical considerations and, in particular, efficiency of the estimation procedure are at the forefront as quantum systems move beyond the curiosity of experimental physics to prototype quantum technology devices.

The most fundamental characterization problem concerns state estimation—

determining an unknown state of a quantum system using a series of different measurements. This procedure is referred to as quantum state estimation or quantum state tomography. Quantum state estimation usually refers to estimating the state using incomplete information, whereas quantum state tomography is often used to describe the situation where complete (and sometimes even noise-free) information about the state is assumed. The two terms can be used interchangeably, though we stick to the former throughout the chapter. The literature on quantum state estimation is extensive (see, e.g. the survey text [39]) with methods ranging from simple linear inversion to least-squares (LS) regression [21], maximum likelihood estimation (MLE) [20, 40], methods based on compressed sensing [41, 42], and the Bayesian approach (see, e.g., [43]). The maximum likelihood method is considered optimal in the sense that it yields a valid state that maximizes the probability of the observed data, and converges to the true state in the limit of many measurements. A disadvantage of the method is that it often yields estimates at the boundary of the state set, i.e. states that are rank deficient.

Gradient-based approximation methods [44, 45], promise to be much faster but they can produce non-physical states (with the estimate either having negative eigenvalues, or being unnormalized) and convergence is in many cases not guaranteed. The former problem can be solved in practice by projecting the state back into the physical space [22]. The same problem is also present in linear regression methods. The matrix exponentiated gradient (MEG) method has found use in classical machine learning [5, 6] and offers an appealing alternative as it by construction ensures positive semidefiniteness of the matrix estimate. In [46], MEG was applied to perform quantum tomography on qubits and approximate the maximum likelihood estimate efficiently. In this chapter, we chose MEG among other online estimation methods as we can show strong convergence results. Other efficient methods such as projected-gradients would be also interesting to explore, but this is outside the scope of this chapter.

In this work, we use the MEG technique to devise an efficient online estimator for quantum states. Our algorithm satisfies the following three desiderata: (1) it is online—providing a running estimate of the state as data is collected; (2) it is fast—its runtime scales well with the dimension of the system; and (3) it comes with a convergence proof. Many other techniques satisfy some of these properties, but we are not aware of any that satisfy all. The main results of our work can be summarized as follows.

- We present the MEG algorithm suitable for online quantum state estimation and robust to noise.

- We prove convergence for noiseless and noisy measurements.

- We numerically compare one of the proposed algorithms with online versions of MLE and LS methods and find that it converges equally fast.

- The proposed algorithm is computationally more efficient than other approaches (such as online versions of MLE and LS), scaling as $O(d^3)$ instead of $O(d^4)$, where d is the dimension of the quantum system.

Our algorithm is naturally *online*, which makes it interesting for many applications. For example, when large amounts of measurements need to be taken to verify a state or when the state is likely to change over time, it can be beneficial to have a running estimate that allows for a rapid diagnosis of error. While any batch algorithm (like the maximum likelihood estimator) can be run on a subset of the initial data points to create an online estimate, this creates a significant overhead and can be avoided using an online estimator.

Related work: A different perspective on quantum state learning has been taken in [47, 48] where instead of learning a full description of the state the goal is only to predict future measurement outcomes. Concurrent with our work, this approach has also been generalized to the online setting in [49], also using variations of the MEG method. The main difference is that their work targets obtaining predictions of future measurement outcomes based on previous ones, which can be achieved without full state tomography. The authors show, somewhat surprisingly, that this can be done up to constant error using only a number of measurements linear in the number of qubits. In contrast full characterization requires exponentially many measurements (see, e.g. [50]). Second, the error criterion to be minimized is based on a mistake bound (i.e. the number of time steps where the prediction was far from the true value), whereas we aim to show asymptotic convergence to the true state. A technical consequence of this is that in [49] the learning rate can be chosen to be a constant whereas we find that for convergence a decreasing learning rate is necessary.

3.2 Summary of Main Results

Let us first describe the MEG update rule (see also Section 3.3 for more details). We assume that the true state, ρ, is finite-dimensional. The update algorithm takes four inputs: $\hat{\rho}_t$ is the estimate of ρ calculated in the previous step; X_t and $\hat{y}_t$ are the observable and measurement outcome at time step t; and η_t is the learning rate at time step t. The algorithm then returns the next estimate of the state, $\hat{\rho}_{t+1}$, as follows.

Algorithm 2 Matrix-exponentiated gradient update rule for quantum state estimation

function UPDATE($\hat{\rho}_t$, X_t, $\hat{y}_t$, η_t)
  $G_{t+1} \leftarrow \log(\hat{\rho}_t) - 2\eta_t(\mathrm{tr}(\hat{\rho}_t X_t) - \hat{y}_t)X_t$ ▷ correct by the gradient of the loss function
  return $\hat{\rho}_{t+1} \leftarrow \frac{\exp(G_{t+1})}{\mathrm{tr}\exp(G_{t+1})}$ ▷ our next estimate, properly normalized
end function

First, we introduce the use of MEG for online quantum state estimation in the ideal case where there is no noise in the measurements. This case may approximate the situation where experimentally a very large number of shots of each measurement are taken. The number of shots refers to the number of copies of the state that are needed to estimate the counts of each possible outcome. So, first the initial estimate is chosen arbitrarily to be the completely mixed state, i.e. $\hat{\rho}_1 = \frac{1}{d}I_d$. Next, a measurement operator X_t is selected at random, and the noiseless measurement is done to obtain $\hat{y}_t = \mathrm{tr}(\rho X_t)$. In this setting the learning rate is chosen to be any constant such that $0 < \eta < \frac{1}{2}$. Finally, the estimate is updated according to the MEG rule as in Algorithm 2. The estimate in this case converges in probability to the true state if the random set of measurements form a unitary one-design, e.g. if they are Pauli measurements in the case of one or more qubits. In other words, we show that for all $\delta > 0$,

$$\lim_{t\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F < \delta\right\} = 1, \tag{3.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (or any other matrix norm) and the probability is taken over the choice of measurements. In fact, we can show that convergence in Frobenius norm is essentially as fast as $1/\sqrt{t}$ in the following sense. For any $\alpha \in (0, \frac{1}{2})$, we have

$$\lim_{T\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F < \frac{1}{t^\alpha}\right\} = 1\,. \tag{3.2}$$

Here the probability is taken over the measurement choices as well as over t uniformly chosen from the set $\{1, 2, \ldots, T\}$. Essentially this tells us that the probability of a random t exceeding the bound $1/t^\alpha$ vanishes, even though we cannot guarantee that the bound is satisfied for any fixed t. The proof of this behavior is presented in Section 3.4.2.

Let us next discuss the (more realistic) case of noisy measurements. Here we are taking a finite number of shots per measurement so that $\hat{y}_t$ is a random variable with mean $\mathrm{tr}(\rho X_t)$ and a variance that depends on the number of shots. In this noisy case the previous scheme will not converge. To see this, assume at some iteration

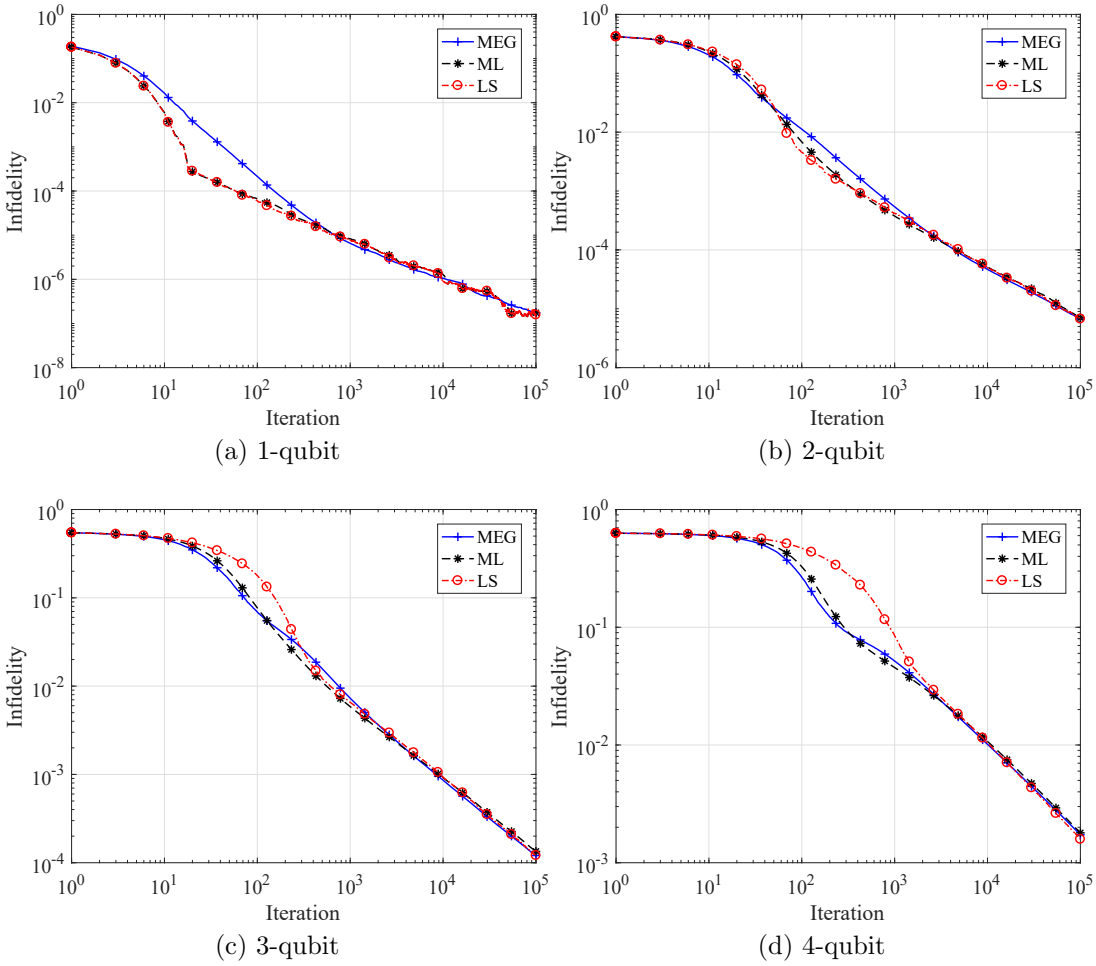


Figure 3.1: Simulation results for different multi-qubit systems: (a) 1-qubit, (b) 2-qubit, (c) 3-qubit, and (d) 4-qubit. The infidelity is averaged over 1000 randomly generated quantum states and plotted versus the iteration number. The three lines correspond to the proposed matrix exponential gradient (MEG) method, maximum likelihood estimator (MLE) and least-squares (LS) estimator. The number of shots per measurement is taken to be 1000 shots.

we hit the true state, $\hat{\rho}_t = \rho$. We then see that even for this state the gradient will be non-zero because in general $\hat{y}_t \neq \mathrm{tr}(\rho X_t)$ and thus the update rule will push the estimate away from the true state. To avoid this behavior, we propose a scheme with an adaptive, decreasing learning rate. We show that a convergence guarantee in the form of (3.2) holds, although the convergence will be slower. To achieve this, for any $\alpha \in (0, \frac{1}{4})$, we set the learning rate to $\eta_t = \frac{1}{4} t^{-\beta}$ with $\beta = \frac{3}{4} - \alpha$ to find that the MEG algorithm satisfies

$$\lim_{T\to\infty} \Pr\left\{ \|\hat{\rho}_t - \rho\|_F < \frac{1}{t^\alpha} \right\} = 1\,, \tag{3.3}$$

where the probability is taken over the measurement choices and outcomes, as well as t uniformly from the set $\{1, 2, \ldots, T\}$. Section 3.4.3 discusses the proof of this

statement.

Finally, for our numerical testing in low dimensions we propose another approach to solve the problem with noisy measurements by using a running average of the measurement outcomes for each measurement. This is effectively equivalent to increasing the number N of shots when certain measurements are repeated. This means that eventually the algorithm approaches the noise-free case and convergence is thus ensured (we leave this as an informal statement). Moreover, numerical simulations show that this method converges faster than using an adaptive learning rate. Figure 3.1 compares the convergence of our algorithm to an MLE and LS estimator for 1-, 2-, 3- and 4-qubit systems, showing that the proposed algorithm converges to the other two methods. We use infidelity between the true state ρ and the estimate $\hat{\rho}_t$ as an accuracy measure, which is defined as $1 - \left(\mathrm{tr}\left|\sqrt{\rho}\sqrt{\hat{\rho}_t}\right|\right)^2$. So, in terms of accuracy measured by infidelity, MEG can perform as well as other methods. Further numerical results can be found in Section 3.5.

In terms of complexity however, MEG outperforms the other methods with complexity of $O(d^3)$ per update compared to $O(d^4)$ for MLE and LS. The bottleneck for MEG is the matrix exponentiation step in the update as seen in Algorithm 2.

3.3 Preliminaries

We give a detailed description of the problem of online quantum state estimation and an overview of the matrix-exponentiated gradient (MEG) update rule.

3.3.1 Problem Statement

Given a quantum system in an unknown state ρ, it is required to find an estimated quantum state $\hat{\rho}$, based on the classical outcomes of some measurements performed on copies of the system. The system has dimensions d, and so for the case of an m-qubit system, we have $d = 2^m$. For the numerical simulations in this chapter we consider such m-qubit systems and perform Pauli measurements on each individual qubit. We shall denote the set of measurements operator by $\{X^{[i]}\}_{i=1}^{d^2-1}$.

The outcome of such a binary measurement is a classical bit. We shall call these outcomes "up" and "down" corresponding to the ± 1 eigenvalues of the Pauli operator. In order to do tomography, we assume that we have an ensemble of identically prepared quantum systems in the same unknown state ρ, so we can perform independent measurements on each of the subsystems, and calculate the average outcome. So, selecting a measurement operator $X_t = X^{[i(t)]}$ at time step t, the expected value of the measurement denoted by y_t as predicted by the Born rule is given by $y_t = \mathrm{tr}(\rho X_t)$, while the actual average we calculate if we repeat the

experiment N times is the random variable

$$\hat{y}_t = \frac{n_\uparrow - n_\downarrow}{N} = \frac{2n_\uparrow - N}{N}. \tag{3.4}$$

Here, $n_\uparrow$ is the number of times the "up" outcome was observed, while $n_\downarrow$ is the number of times the "down" outcome was observed. We know that $n_\uparrow$ follows a binomial distribution. Given a measurement operator represented in terms of its eigenvalue projectors as $X_t = \Pi_\uparrow - \Pi_\downarrow$, we have $n_\uparrow \sim B(N, p)$ with $p = \mathrm{tr}(\rho\Pi_\uparrow)$. It is then easy to verify that

$$\mathbb{E}\{\hat{y}_t\} = 2p - 1 = y_t\,, \quad \text{and} \quad \mathrm{Var}\{\hat{y}_t\} = \frac{4p(1-p)}{N} = \frac{1 - y_t^2}{N}\,. \tag{3.5}$$

We can then repeat the whole procedure and obtain a sequence of data points in the form $\{(X_1, \hat{y}_1), ...(X_t, \hat{y}_t), ...\}$. Notice that the measurement outcomes $\hat{y}_t$ form an independent and identically distributed (i.i.d.) set of random variables. Since we are proposing an online algorithm, we do not have the whole data set in advance. We obtain one point at a time, and use it to update an estimate $\hat{\rho}_t$ of the true state. We would like that $\hat{\rho}_t$ converges to ρ as t increases.

3.3.2 The Matrix-Exponentiated Gradient Method

The MEG method was proposed in [5, 6] for some classical machine learning applications and symmetric matrices. The algorithm trivially generalizes to Hermitian matrices. Given a new data point $(X_t, \hat{y}_t)$, the loss function at time step t, evaluated for a general quantum state σ, is defined as

$$L_t(\sigma) := (\mathrm{tr}(\sigma X_t) - \hat{y}_t)^2. \tag{3.6}$$

The gradient of the loss function at time step t is then

$$\nabla L_t = 2(\mathrm{tr}(\sigma X_t) - \hat{y}_t)X_t. \tag{3.7}$$

Consider now the following online cost function

$$D(\hat{\rho}_{t+1}||\hat{\rho}_t) + \eta_t L_t(\hat{\rho}_{t+1}), \tag{3.8}$$

where D is Umegaki's quantum relative entropy [51] defined as $D(\rho||\sigma) = \mathrm{tr}(\rho\log(\rho) - \rho\log(\sigma))$ for any two states ρ and σ, and η_t is the learning rate. This cost function represents two conflicting goals. The first one is to have an estimate that is near the previous estimate, quantified by the relative entropy. This is important because in the online setting of the problem, we do not want the algorithm to forget what

it has learnt so far. The second goal is to move the new estimate so that the loss function at the new data point is hopefully smaller. The learning rate η_t controls this trade-off. Minimizing the cost function with respect to $\hat{\rho}_{t+1}$ by taking the gradient (see Appendix A in [5] for the details of the calculation) and setting it to zero results in

$$\log(\hat{\rho}_{t+1}) = \log(\hat{\rho}_t) - \eta \nabla L_t(\hat{\rho}_{t+1}) - I, \tag{3.9}$$

where I denotes the identity matrix. Now, since we cannot find an explicit form for $\hat{\rho}_{t+1}$, we may approximate $\hat{\rho}_{t+1}$ by $\hat{\rho}_t$ in the gradient to arrive at $\log(\hat{\rho}_{t+1}) = \log(\hat{\rho}_t) - \eta \nabla L_t(\hat{\rho}_t) - I$, or, equivalently,

$$\hat{\rho}_{t+1} = \exp\left(\log\left(\hat{\rho}_t\right) - \eta \nabla L_t(\hat{\rho}_t) - I\right). \tag{3.10}$$

This form of the update rule ensures that if we start with a positive definite matrix $\hat{\rho}_t$, and a Hermitian operator X_t, then we are sure that the new estimate $\hat{\rho}_{t+1}$ is positive definite. This is because the terms inside the exponential function are Hermitian, and thus the matrix exponential results in a positive definite matrix. Next, we want to make sure that the estimate has unit trace, to be a valid quantum state. So, we normalize to finally obtain the MEG rule:

$$\hat{\rho}_{t+1} = \frac{\exp\left(\log\left(\hat{\rho}_t\right) - \eta \nabla L_t(\hat{\rho}_t)\right)}{\operatorname{tr}\left(\exp\left(\log\left(\hat{\rho}_t\right) - \eta \nabla L_t(\hat{\rho}_t)\right)\right)}. \tag{3.11}$$

The update rule can also be expressed in the following compact alternative form:

$$G_t = G_{t-1} - \eta \nabla L_t(\hat{\rho}_t), \quad G_0 = \log(\hat{\rho}_0), \quad \text{and} \quad \hat{\rho}_t = \frac{\exp(G_t)}{\operatorname{tr}\exp(G_t)}. \tag{3.12}$$

3.3.3 Auxiliary Lemmas

In this Subsection, we present some auxiliary lemmas needed for some proofs. We will start by stating the following lemma [5], which is proved as Lemma 1 in [52].

Lemma 1. *Let $0 \leq q \leq 1$, then for any p,*

$$\log\left(1 - q\left(1 - \exp\left(p\right)\right)\right) \leq pq + \frac{p^2}{8}. \tag{3.13}$$

Next, we state the Golden-Thompson inequality [53, 54].

Lemma 2 (Golden-Thompson Inequality)**.** *Let A and B be two Hermitian matrices, then*

$$\operatorname{tr}(\exp(A + B)) \leq \operatorname{tr}(\exp(A)\exp(B)). \tag{3.14}$$

The following result is presented as Lemma 2.1 in [5].

Lemma 3 (Jensen's Inequality)**.** *Let $0 \leq A \leq I$, and $x, y \geq 0$, then*

$$\exp(xA + y(I - A)) \leq \exp(x)A + \exp(y)(I - A) \tag{3.15}$$

$$= I\exp(y) + (\exp(x) - \exp(y))A. \tag{3.16}$$

Finally, we state the following lemma relating convergence in mean to convergence in probability.

Lemma 4. *Given a sequence of positive random variables Z_t,*

$$\lim_{t\to\infty} \mathbb{E}\{Z_t\} = 0 \implies \forall\delta > 0, \lim_{t\to\infty} \Pr\{Z_t \leq \delta\} = 1. \tag{3.17}$$

Proof. The statement

$$\lim_{t\to\infty} \mathbb{E}\{Z_t\} = 0. \tag{3.18}$$

is equivalent to the statement

$$\forall\epsilon > 0, \delta > 0, \exists T_\delta : \forall t > T_\delta, \mathbb{E}\{Z_t\} \leq \delta\epsilon. \tag{3.19}$$

Now, Markov inequality states that for a non-negative random variable X,

$$\Pr\{X \geq a\} \leq \frac{\mathbb{E}\{X\}}{a}. \tag{3.20}$$

So, the previous definition of the limit becomes

$$\forall\epsilon > 0, \delta > 0, \exists T_\delta : \forall t > T_\delta, \Pr\{Z_t \geq \delta\} \leq \epsilon, \tag{3.21}$$

or,

$$\forall\epsilon > 0, \delta > 0, \exists T_\delta : \forall t > T_\delta, \Pr\{Z_t < \delta\} \geq 1 - \epsilon. \tag{3.22}$$

Writing back as a limit, the expression becomes

$$\forall\delta > 0, \lim_{t\to\infty} \Pr\{Z_t < \delta\} = 1, \tag{3.23}$$

which is the definition of convergence in probability. □

3.4 Convergence Analysis

This section starts with stating some bounds related to the MEG update rule. Next, the proof of convergence for the noise-free case is given, followed by the proof of convergence in the noisy case. Finally, a discussion about the proposed running-average technique is presented.

3.4.1 General Bounds on the Loss Function

We will start by stating the following lemma which bounds the normalization constant that appears in the MEG update rule

$$\log(\hat{\rho}_{t+1}) = \log(\hat{\rho}_t) + \delta_t X_t - \log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))), \tag{3.24}$$

where

$$\delta_t = -2\eta(\mathrm{tr}(\hat{\rho}_t X_t) - \hat{y}_t), \tag{3.25}$$

and measurement operators satisfying $-I \leq X_t \leq I$ to ensure that the updated estimate has unit trace. This bound will be used to prove other important results. The proof generalizes the methods that involved real symmetric matrices in [5] to complex Hermitian matrices.

Lemma 5. *The normalization constant in the MEG rule update is bounded by*

$$\log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))) \leq \frac{\delta_t^2}{2} + \delta_t \,\mathrm{tr}\left(\hat{\rho}_t X_t\right). \tag{3.26}$$

Proof. Recall that

$$\delta_t = -2\eta(\mathrm{tr}(\hat{\rho}_t X_t) - \hat{y}_t). \tag{3.27}$$

Applying Golden-Thompson inequality in Lemma 2, we get

$$\begin{aligned}
\log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))) &\leq \log(\mathrm{tr}(\hat{\rho}_t \exp(\delta_t X_t))) && (3.28)\\
&= \log\left(\mathrm{tr}\left(\hat{\rho}_t \exp(-\delta_t)\exp\left(2\delta_t \frac{X_t + I}{2}\right)\right)\right) && (3.29)\\
&= -\delta_t + \log\left(\mathrm{tr}\left(\hat{\rho}_t \exp\left(2\delta_t \frac{X_t + I}{2}\right)\right)\right). && (3.30)
\end{aligned}$$

Applying Jensen's inequality in Lemma 3 by choosing $A = \frac{X_t+I}{2}$, $x = 2\delta_t$, and $y = 0$;

we get that

$$\log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))) \leq -\delta_t + \log\left(\mathrm{tr}\left(\hat{\rho}_t\left(I - (1-\exp(2\delta_t))\,\frac{X_t + I}{2}\right)\right)\right) \tag{3.31}$$

$$= -\delta_t + \log\left(1 - (1-\exp(2\delta_t))\,\frac{\mathrm{tr}\left(\hat{\rho}_t(X_t + I)\right)}{2}\right) \tag{3.32}$$

$$= -\delta_t + \log\left(1 - (1-\exp(2\delta_t))\,\frac{\mathrm{tr}\left(\hat{\rho}_t X_t\right) + 1}{2}\right). \tag{3.33}$$

Applying now the log identity in Lemma 1, with $p = 2\delta_t$, and $q = \frac{\mathrm{tr}(\hat{\rho}_t X_t)+1}{2}$, we obtain

$$\log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))) \leq -\delta_t + \frac{(2\delta_t)^2}{8} + 2\delta_t \frac{\mathrm{tr}\left(\hat{\rho}_t X_t\right) + 1}{2} \tag{3.34}$$

$$= \frac{\delta_t^2}{2} + \delta_t\,\mathrm{tr}\left(\hat{\rho}_t X_t\right), \tag{3.35}$$

which completes the proof. □

Next, we state the following lemma which puts a bound on the difference between the loss function evaluated at the estimate, and a general state. The lemma relates this difference to the progress of the estimator towards that general state. This is the main lemma that will be used to prove the convergence of MEG. The proof generalizes the results in [5] to the quantum setting.

Lemma 6. *Given the loss function* $L_t(\hat{\rho}_t) = (\mathrm{tr}(\hat{\rho}_t X_t) - \hat{y}_t)^2$ *with measurement operators* $-I \leq X_t \leq I$ *and learning rate* $0 < \eta < \frac{1}{2}$*, then for any state* σ*,*

$$\eta L_t(\hat{\rho}_t) - \frac{\eta}{1-2\eta} L_t(\sigma) \leq D(\sigma||\hat{\rho}_t) - D(\sigma||\hat{\rho}_{t+1}). \tag{3.36}$$

Proof. We start with calculating the right hand side,

$$D(\sigma||\hat{\rho}_t) - D(\sigma||\hat{\rho}_{t+1}) = \mathrm{tr}(\sigma\log(\sigma) - \sigma\log(\hat{\rho}_t)) - \mathrm{tr}(\sigma\log(\sigma) - \sigma\log(\hat{\rho}_{t+1})) \tag{3.37}$$

$$= -\mathrm{tr}(\sigma\log(\hat{\rho}_t)) + \mathrm{tr}(\sigma\log(\hat{\rho}_{t+1})) \tag{3.38}$$

$$= -\mathrm{tr}(\sigma\log(\hat{\rho}_t)) + \mathrm{tr}(\sigma\log(\hat{\rho}_t)) + \mathrm{tr}(\sigma\delta_t X_t)$$
$$-\mathrm{tr}(\sigma\log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t)))) \tag{3.39}$$

$$= \delta_t\,\mathrm{tr}(\sigma X_t) - \log(\mathrm{tr}(\exp(\log(\hat{\rho}_t) + \delta_t X_t))) \tag{3.40}$$

Applying Lemma 5 we get

$$D(\sigma||\hat{\rho}_t) - D(\sigma||\hat{\rho}_{t+1}) \geq \delta_t \operatorname{tr}(\sigma X_t) - \frac{\delta_t^2}{2} - \delta_t \operatorname{tr}(\hat{\rho}_t X_t) \tag{3.41}$$

$$= -2\eta(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t)(\operatorname{tr}(\sigma X_t) - \operatorname{tr}(\hat{\rho}_t X_t)) - 2\eta^2(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t)^2 \tag{3.42}$$

$$= -2\eta(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t)(\operatorname{tr}(\sigma X_t) - \hat{y}_t + \hat{y}_t - \operatorname{tr}(\hat{\rho}_t X_t)) - 2\eta^2(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t)^2 \tag{3.43}$$

$$= \left(2\eta - 2\eta^2\right)(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t)^2 - 2\eta(\operatorname{tr}(\sigma X_t) - \hat{y}_t)(\operatorname{tr}(\hat{\rho}_t X_t) - \hat{y}_t) \tag{3.44}$$

$$\geq \left(2\eta - 2\eta^2\right) L_t(\hat{\rho}_t) - 2\eta\sqrt{L_t(\hat{\rho}_t)L_t(\sigma)} \tag{3.45}$$

$$= \left(\sqrt{\eta - 2\eta^2}\sqrt{L_t(\hat{\rho}_t)} - \sqrt{\frac{\eta^2}{\eta - 2\eta^2}}\sqrt{L_t(\sigma)}\right)^2 + \eta L_t(\hat{\rho}_t) - \frac{\eta^2}{\eta - 2\eta^2} L_t(\sigma). \tag{3.46}$$

If we now choose $\eta - 2\eta^2 > 0$, then the square roots in the last expression are real valued. As a result,

$$D(\sigma||\hat{\rho}_t) - D(\sigma||\hat{\rho}_{t+1}) \geq \eta L_t(\hat{\rho}_t) - \frac{\eta}{1 - 2\eta} L_t(\sigma), \tag{3.47}$$

and the learning factor η must satisfy

$$0 < \eta < \frac{1}{2}, \tag{3.48}$$

which completes the proof of the lemma. To account for noiseless measurements, $\hat{y}_t$ is just replaced by y_t. □

This leads to the following corollary that bounds the loss function of the estimate when the true state is used as the comparison state, in the case of noise-free measurements (i.e. $\hat{y}_t = y_t$).

Corollary 1. *Given the loss function $L_t(\hat{\rho}_t) = (\operatorname{tr}(\hat{\rho}_t X_t) - y_t)^2$ with measurement operators $-I \leq X_t \leq I$ and learning rate $0 < \eta < \frac{1}{2}$. Then, given the true state ρ, the following relation holds:*

$$\eta L_t(\hat{\rho}_t) \leq D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1}). \tag{3.49}$$

Proof. Apply Lemma 6, set $\sigma = \rho$, and use the fact that $L_t(\rho) = 0$. □

3.4.2 Convergence Analysis for Noiseless Measurements

The choice of measurements for doing quantum state estimation is arbitrary. However, in this chapter we consider the case of performing local Pauli measurements on each qubit of a multi-qubit system. This facilitates the experimental realization compared to performing some other, possibly global, measurement. The proofs will start by calculating some expectation values involving Pauli operators and loss functions. These results will be used to prove the main theorem showing the convergence of MEG in the noise-free case. We start with the following lemma about the the set of Pauli operators for multi-qubit systems.

Lemma 7. *The set $U = \{U_i\}_{i=0}^{d^2-1}$ of Pauli operators including the identity operator in a d-dimensional quantum system satisfy*

$$\frac{1}{d}\sum_i U_i \otimes U_i^\dagger = P_{21}, \tag{3.50}$$

where P_{21} is the swap operator defined as

$$P_{21} = \sum_{i,j} |i\rangle\langle j| \otimes |j\rangle\langle i|. \tag{3.51}$$

Proof. The Pauli's form a unitary orthonormal basis of Hermitian $d \times d$ matrices. Therefore, they form a quantum 1-design due to Proposition 6 in [55]. In other words,

$$\int_U U\rho U^\dagger dU = \sum_i \frac{1}{d^2} U_i \rho U_i^\dagger. \tag{3.52}$$

Now, from (3.27) and (3.29) in [55], $\sum_i \frac{1}{d^2} U_i \otimes U_i^\dagger = \frac{P_{21}}{d}$. □

Next, we calculate the expectation value of a Pauli operator that is tensored with itself. This calculation will be needed in the calculation of the expectation of the loss function.

Lemma 8. *The expectation value of the Pauli operators chosen uniformly at random from the set $U - \{I\}$ satisfies the relation:*

$$\mathbb{E}_X \{X \otimes X\} = \frac{d}{d^2-1} P_{21} - \frac{1}{d^2-1} I_d \otimes I_d \tag{3.53}$$

Proof. We have

$$\mathbb{E}_X \{X \otimes X\} = \frac{1}{d^2-1} \sum_{j=1}^{d^2-1} X^{[j]} \otimes X^{[j]} \tag{3.54}$$

$$= \frac{1}{d^2-1}\left(\sum_{i=0}^{d^2-1} U_i \otimes U_i - I_d \otimes I_d\right) \tag{3.55}$$

$$= \frac{d}{d^2-1} P_{21} - \frac{1}{d^2-1} I_d \otimes I_d, \tag{3.56}$$

where the last equality holds from Lemma 7, and I_d is the identity operator of dimension d. □

The following lemma is a commonly-used result in quantum information. The proof is direct—see for example Lemma 1.2.1 in [56].

Lemma 9 (Swap trick)**.** *For any quantum system with arbitrary dimensions, and for two operators M and N, we have $\mathrm{tr}(MN) = \mathrm{tr}\left((M \otimes N)P_{21}\right)$, where P_{21} is the swap operator on the quantum system (interchanges any two copies).*

We are now ready to prove the following lemma in which the expectation of the loss function is calculated.

Lemma 10. *Assuming we select the measurement operator X_t at each time iteration uniformly at random from the set $U - \{I\}$ then for any true state ρ and any state σ independent of X_t,*

$$\mathbb{E}_{X_t}\{L_t(\sigma)\} = \frac{d}{d^2-1}\left\|\sigma - \rho\right\|_F^2. \tag{3.57}$$

Proof. From the definition of the loss function,

$$L_t(\sigma) = (\mathrm{tr}(\sigma X_t) - \mathrm{tr}(\rho X_t))^2. \tag{3.58}$$

Taking the expectation of the loss function with respect to X_t we get:

$$\mathbb{E}_{X_t}\{L_t(\sigma)\} = \mathbb{E}_{X_t}\{(\mathrm{tr}(\sigma X_t) - \mathrm{tr}(\rho X_t))^2\} \tag{3.59}$$

$$= \mathbb{E}_{X_t}\{(\mathrm{tr}(\sigma - \rho)X_t)^2\} \tag{3.60}$$

$$= \mathbb{E}_{X_t}\left\{\mathrm{tr}\left(\left((\sigma-\rho)X_t\right) \otimes \left((\sigma-\rho)X_t\right)\right)\right\} \tag{3.61}$$

$$= \mathbb{E}_{X_t}\left\{\mathrm{tr}\left(\left((\sigma-\rho) \otimes (\sigma-\rho)\right)(X_t \otimes X_t)\right)\right\} \tag{3.62}$$

$$= \mathrm{tr}\left(\left((\sigma-\rho) \otimes (\sigma-\rho)\right)\mathbb{E}_{X_t}\left\{X_t \otimes X_t\right\}\right). \tag{3.63}$$

Then, applying Lemma 8,

$$\mathbb{E}_{X_t}\{L_t(\sigma\} = \mathrm{tr}\left(\left((\sigma-\rho) \otimes (\sigma-\rho)\right)\left(\frac{d}{d^2-1}P_{21} - \frac{1}{d^2-1} I \otimes I\right)\right) \tag{3.64}$$

$$= \frac{d\,\mathrm{tr}\left(\left((\sigma-\rho) \otimes (\sigma-\rho)\right)P_{21}\right)}{d^2-1} - \frac{\mathrm{tr}\left(\left((\sigma-\rho) \otimes (\sigma-\rho)\right)(I \otimes I)\right)}{d^2-1}. \tag{3.65}$$

Now, applying the swap trick in Lemma 9,

$$\mathbb{E}_{X_t}\{L_t(\sigma)\} = \frac{d}{d^2-1}\operatorname{tr}\big((\sigma-\rho)(\sigma-\rho)\big) - \frac{1}{d^2-1}\operatorname{tr}(\sigma-\rho)\operatorname{tr}(\sigma-\rho) \tag{3.66}$$

$$= \frac{d}{d^2-1}\operatorname{tr}\big((\sigma-\rho)^2\big) \tag{3.67}$$

$$= \frac{d}{d^2-1}\|\sigma-\rho\|_F^2 . \tag{3.68}$$

In particular, it is clear that at any time step t, if $\sigma \neq \rho$, then $\mathbb{E}\{L_t(\sigma)\} > 0$. □

Now, we can show the following theorem considering convergence of the noiseless MEG.

Theorem 1. *The state estimate using the MEG update rule converges in probability to the true state, i.e. for any $\delta > 0$,*

$$\lim_{t\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \delta\right\} = 1. \tag{3.69}$$

Proof. We know from Corollary 1 that,

$$\eta L_t(\hat{\rho}_t) \leq D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1}). \tag{3.70}$$

Taking the expectation with respect to X_t,

$$\eta\,\mathbb{E}_{X_t}\{L_t(\hat{\rho}_t)\} \leq \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_t)\} - \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.71}$$

Applying Lemma 10, and using the fact that $\hat{\rho}_t$ is independent of X_t we get

$$\frac{\eta d}{d^2-1}\|\hat{\rho}_t - \rho\|_F^2 \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.72}$$

Taking the expectation of the above inequality over all past time iterations $\mathbb{E} = \mathbb{E}_{X_1} \ldots \mathbb{E}_{X_{t-1}}$ we get

$$\frac{\eta d}{d^2-1}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} \leq \mathbb{E}\{D(\rho||\hat{\rho}_t)\} - \mathbb{E}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.73}$$

Next, we sum the inequality over the time iterations to get

$$\frac{\eta d}{d^2-1}\sum_{t=1}^{T}\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} \leq \mathbb{E}\{D(\rho||\hat{\rho}_1)\} - \mathbb{E}\{D(\rho||\hat{\rho}_{T+1})\} \tag{3.74}$$

$$\leq \mathbb{E}\{D(\rho||\hat{\rho}_1)\}. \tag{3.75}$$

If we now take the limit as $T \to \infty$, we obtain

$$\sum_{t=1}^{\infty} \mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} \leq \frac{d^2-1}{\eta d} \mathbb{E}\{D(\rho||\hat{\rho}_1)\} \tag{3.76}$$

$$= \frac{d^2-1}{\eta d} D(\rho||\hat{\rho}_1). \tag{3.77}$$

Where the last line follows from the fact that the true state ρ and the initial estimate $\hat{\rho}_1$ are independent of X_t and y_t. Now the right hand side of the inequality is constant, so the series on the left hand side of the inequality converges. This implies by the divergence test that

$$\lim_{t\to\infty} \mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} = 0. \tag{3.78}$$

Now we can apply Lemma 4 on the random variable $Z_t = \|\hat{\rho}_t - \rho\|_F^2$ to conclude that

$$\forall \delta > 0 : \quad \lim_{t\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 \leq \delta\right\} = 1. \tag{3.79}$$

Therefore, the estimate $\hat{\rho}_t$ converges in probability to the true state ρ. □

Finally, we prove the main theorem that shows a stronger statement for the convergence of MEG algorithm in the case of noise-free measurements.

Theorem 2. *Let $\delta \in (0,1)$. Then for any $\alpha \in (0,1)$, and learning rate $0 < \eta < \frac{1}{2}$, there exists T_0 given by*

$$T_0 = \left(\frac{\frac{d^2-1}{\eta d}\log d + 2}{\delta}\right)^{\frac{3}{1-\alpha}}, \tag{3.80}$$

such that for any $T > T_0$ we have,

$$\Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\} \geq 1 - \delta, \tag{3.81}$$

where the probability is taken over all measurement choices and t uniformly in $\{1, 2, \ldots, T\}$. Moreover,

$$\lim_{T\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\} = 1\,. \tag{3.82}$$

Proof. Let the initial estimate be $\hat{\rho}_1 = \frac{I_d}{d}$. We know from Corollary 1 that,

$$\eta L_t(\hat{\rho}_t) \leq D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1}). \tag{3.83}$$

Taking the expectation with respect to X_t,

$$\eta \, \mathbb{E}_{X_t}\{L_t(\hat{\rho}_t)\} \leq \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_t)\} - \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.84}$$

Applying Lemma 10, and using the fact that $\hat{\rho}_t$ is independent of X_t we get

$$\frac{\eta d}{d^2-1} \left\|\hat{\rho}_t - \rho\right\|_F^2 \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_{X_t}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.85}$$

Taking the expectation of the above inequality over all past time iterations $\mathbb{E} = \mathbb{E}_{X_0} \mathbb{E}_{X_1} ... \mathbb{E}_{X_{t-1}}$ we get

$$\frac{\eta d}{d^2-1} \mathbb{E}\{\left\|\hat{\rho}_t - \rho\right\|_F^2\} \leq \mathbb{E}\{D(\rho||\hat{\rho}_t)\} - \mathbb{E}\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.86}$$

Next, we sum the inequality over the time iterations to get

$$\begin{aligned} \frac{\eta d}{d^2-1} \sum_{t=1}^{T} \mathbb{E}\{\left\|\hat{\rho}_t - \rho\right\|_F^2\} &\leq \mathbb{E}\{D(\rho||\hat{\rho}_1)\} - \mathbb{E}\{D(\rho||\hat{\rho}_{T+1})\} && (3.87) \\ &\leq \mathbb{E}\{D(\rho||\hat{\rho}_1)\} && (3.88) \\ &\leq \log d. && (3.89) \end{aligned}$$

Now, let $\epsilon_t = \mathbb{E}\{\left\|\hat{\rho}_t - \rho\right\|_F^2\}$, $\delta_t = \frac{1}{t^{\alpha+\gamma}}$, and $\gamma = \frac{2}{3}(1-\alpha)$. Notice that $\alpha + \gamma < 1$. Define the set

$$\mathcal{T} := \left\{ t \in \{1, 2, ..T\} : \epsilon_t \geq \frac{1}{t^{\alpha+\gamma}} \right\}. \tag{3.90}$$

Rearranging the terms in the inequality we get

$$\begin{aligned} \frac{d^2-1}{\eta d} \log d &\geq \sum_{t=1}^{T} \epsilon_t && (3.91) \\ &\geq \sum_{t=1}^{T} \left(\mathbb{1}_{\epsilon_t > \delta_t}\right) \epsilon_t && (3.92) \\ &\geq \sum_{t=1}^{T} \left(\mathbb{1}_{\epsilon_t > \delta_t}\right) \frac{1}{t^{\alpha+\gamma}} && (3.93) \\ &\geq \sum_{t=1}^{T} \left(\mathbb{1}_{\epsilon_t > \delta_t}\right) \frac{1}{T^{\alpha+\gamma}} && (3.94) \\ &\geq \frac{|\mathcal{T}_\delta|}{T^{\alpha+\gamma}}. && (3.95) \end{aligned}$$

In other words, the ratio between the number of iterations in which $\mathbb{E}\{\left\|\hat{\rho}_t - \rho\right\|_F^2\} \geq$

$\frac{1}{t^{\alpha+\gamma}}$ and the total number of iterations T we performed so far is bounded by

$$\frac{|\mathcal{T}_\delta|}{T} \leq KT^{\alpha+\gamma-1}, \tag{3.96}$$

where $K := \frac{d^2-1}{\eta d}\log d$. This implies that

$$\lim_{T\to\infty}\left(\frac{T_\delta}{T}\right) = 0, \tag{3.97}$$

because $\alpha+\gamma<1$. This means that increasing the number of iterations results in decreasing the number of times where the estimate was not accurate enough. Let's state this formally. Assume we do a total number of iterations T. If we select at random a fixed time step $1 \leq \tilde{t} \leq T$, then there will be two possible outcomes. Either $\epsilon_{\tilde{t}} \leq \delta_{\tilde{t}}$ or $\epsilon_{\tilde{t}} > \delta_{\tilde{t}}$. Assume we get the first outcome, then by applying Markov's inequality,

$$\epsilon_{\tilde{t}} \leq \delta_{\tilde{t}} \implies \Pr\left\{\|\hat{\rho}_{\tilde{t}}-\rho\|_F^2\right\} \geq \frac{1}{\tilde{t}^\alpha}\Bigg\} \leq \mathbb{E}\left\{\|\hat{\rho}_{\tilde{t}}-\rho\|_F^2\right\}\tilde{t}^\alpha \tag{3.98}$$

$$\leq \delta_{\tilde{t}}\tilde{t}^\alpha \tag{3.99}$$

$$= \tilde{t}^{-\gamma}. \tag{3.100}$$

Now, we can find the joint probability

$$\begin{aligned}
\Pr_{t,\hat{\rho}_t}\left\{\|\hat{\rho}_t-\rho\|_F^2 \geq \frac{1}{t^\alpha}\right\} &= \Pr\left\{\|\hat{\rho}_t-\rho\|_F^2\} \geq \frac{1}{t^\alpha}\middle| t=\tilde{t}\right\}\Pr\left\{\tilde{t}\in\mathcal{T}_\delta\right\} \\
&\quad + \Pr\left\{\|\hat{\rho}_t-\rho\|_F^2\} \geq \frac{1}{t^\alpha}\middle| t=\tilde{t}\right\}\Pr\left\{\tilde{t}\notin\mathcal{T}_\delta\right\} && (3.101)\\
&\leq \Pr\left\{\tilde{t}\in\mathcal{T}_\delta\right\} + \sum_{\tilde{t}\notin\mathcal{T}_\delta}\Pr\left\{\|\hat{\rho}_t-\rho\|_F^2\} \geq \frac{1}{t^\alpha}\middle| t=\tilde{t}\right\}\frac{1}{T} && (3.102)\\
&\leq \frac{|\mathcal{T}_\delta|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T}\Pr\left\{\|\hat{\rho}_t-\rho\|_F^2\} \geq \frac{1}{t^\alpha}\middle| t=\tilde{t}\right\} && (3.103)\\
&\leq \frac{|\mathcal{T}_\delta|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T}\tilde{t}^{-\gamma} && (3.104)\\
&\leq \frac{|\mathcal{T}_\delta|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T}\left(\frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}}. && (3.105)
\end{aligned}$$

Applying Jensen's inequality on the second term (noting that $f(x)=x^r$ is a

concave function for $0 < r < 1$) yields

$$\Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 \geq \frac{1}{t^\alpha}\right\} = \frac{|\mathcal{T}_\delta|}{T} + \left(\frac{1}{T}\sum_{\tilde{t}=1}^{T}\frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}} \tag{3.106}$$

$$\leq \frac{|\mathcal{T}_\delta|}{T} + \left(\frac{1}{T}\sum_{\tilde{t}=1}^{\infty}\frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}} \tag{3.107}$$

$$= \frac{|\mathcal{T}_\delta|}{T} + \left(\frac{\pi^2}{6}\frac{1}{T}\right)^{\frac{\gamma}{2}} \tag{3.108}$$

$$\leq \frac{|\mathcal{T}_\delta|}{T} + 2T^{-\frac{\gamma}{2}} \tag{3.109}$$

Therefore,

$$\Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\} \geq 1 - \frac{|\mathcal{T}_\delta|}{T} - 2T^{-\frac{\gamma}{2}} \tag{3.110}$$

$$\geq 1 - KT^{\alpha+\gamma-1} - 2T^{-\frac{\gamma}{2}} \tag{3.111}$$

$$= 1 - KT^{-\frac{1-\alpha}{3}} - 2T^{-\frac{1-\alpha}{3}} \tag{3.112}$$

$$= 1 - T^{-\frac{1-\alpha}{3}}(K+2)\,. \tag{3.113}$$

Now, let

$$T_0 = \left(\frac{K+2}{\delta}\right)^{\frac{3}{1-\alpha}}, \tag{3.114}$$

then, if choose $T > T_0$, then

$$\delta \geq T^{-\frac{1-\alpha}{3}}(K+2)\,, \tag{3.115}$$

or,

$$1 - \delta \leq 1 - T^{-\frac{1-\alpha}{3}}(K+2) \tag{3.116}$$

$$\leq \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\}. \tag{3.117}$$

□

Notice, that taking the limit as $T \to \infty$ we obtain that $\delta = 0$, and therefore

$$\lim_{T\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 \geq \frac{1}{t^\alpha}\right\} = 0\,. \tag{3.118}$$

3.4.3 Convergence Analysis for Noisy Measurements

In this part, we show that using an adaptive learning rate with noisy measurements results in the convergence of the MEG estimate to the true state. First, some

expectation values will be calculated based on similar techniques discussed in the noiseless case. After that, the optimal adaptive learning rate is derived in such a way to ensure the convergence of the estimate to the true state in probability. However, the learning rate in this case depends on the true state which is not practical. So, finally we show that we can choose a learning rate independent of the true state and prove even a stronger statement of convergence.

We will start with the following lemma to calculate the expectation value of the noise term that appears in the loss function due to performing finite number of measurements.

Lemma 11. *The expectation value of the Pauli operators chosen uniformly at random from the set $U - \{I\}$ satisfy the relation:*

$$\mathbb{E}_{X_t}\left\{\frac{1-y_t^2}{N}\right\} = \frac{d}{N(d^2-1)}\left(d - \|\rho\|_F^2\right). \tag{3.119}$$

Proof. We have

$$\begin{aligned}
\mathbb{E}_{X_t}\left\{\frac{1-y_t^2}{N}\right\} &= \frac{1-\mathbb{E}_{X_t}\{y_t^2\}}{N} && (3.120)\\
&= \frac{1-\mathbb{E}_{X_t}\{\mathrm{tr}(\rho X_t)^2\}}{N} && (3.121)\\
&= \frac{1-\mathbb{E}_{X_t}\{\mathrm{tr}((\rho\otimes\rho)(X_t\otimes X_t))\}}{N} && (3.122)\\
&= \frac{1-\mathrm{tr}((\rho\otimes\rho)\,\mathbb{E}_{X_t}\{X_t\otimes X_t\})}{N}. && (3.123)
\end{aligned}$$

Applying now Lemma 8, we get

$$\begin{aligned}
\mathbb{E}_{X_t}\left\{\frac{1-y_t^2}{N}\right\} &= \frac{1-\mathrm{tr}\left((\rho\otimes\rho)(\frac{d}{d^2-1}P_{21}-\frac{1}{d^2-1}I\otimes I)\right)}{N} && (3.124)\\
&= \frac{1}{N(d^2-1)}\left(d^2-1-d\,\mathrm{tr}(\rho^2)+\mathrm{tr}(\rho)^2\right) && (3.125)\\
&= \frac{d}{N(d^2-1)}\left(d-\|\rho\|_F^2\right), && (3.126)
\end{aligned}$$

where the swap trick in Lemma 9 is used in the second line. □

Next, we give the following lemma to calculate the expectation of the loss function for the case of noisy measurements.

Lemma 12. *Assuming we select the measurement operator X_t at each time iteration uniformly at random from the set $U - \{I\}$ then for any true state ρ and any state*

σ independent of X_t and $\hat{y}_t$ for any t,

$$\mathbb{E}_t\{L_t(\sigma)\} = \frac{d}{d^2-1}\left(\|\sigma-\rho\|_F^2 + \frac{d-\|\rho\|_F^2}{N}\right). \tag{3.127}$$

Proof. Recall the noisy loss function,

$$L_t(\sigma) = (\mathrm{tr}(\sigma X_t) - \hat{y}_t)^2. \tag{3.128}$$

Note that σ is independent of $\hat{y}_t$, but can depend on the previous history. So, the expectation can be calculated as

$$\begin{aligned}
\mathbb{E}_{\hat{y}_t}\{L_t(\sigma)\} &= \mathbb{E}_{\hat{y}_t}\{(\mathrm{tr}(\sigma X_t) - \hat{y}_t)^2\} && (3.129)\\
&= \mathrm{tr}(\sigma X_t)^2 - 2\,\mathrm{tr}(\sigma X_t)\,\mathbb{E}_t\{\hat{y}_t\} + \mathbb{E}_t\{\hat{y}_t^2\} && (3.130)\\
&= \mathrm{tr}(\sigma X_t)^2 - 2\,\mathrm{tr}(\sigma X_t) y_t + y_t^2 + \frac{1-y_t^2}{N} && (3.131)\\
&= (\mathrm{tr}(\sigma X_t) - y_t)^2 + \frac{1-y_t^2}{N}. && (3.132)
\end{aligned}$$

Now, Let's take the expectation with respect to X_t as

$$\begin{aligned}
\mathbb{E}_t\{L_t(\sigma)\} &= \mathbb{E}_{X_t}\,\mathbb{E}_{\hat{y}_t}\{L_t(\hat{\rho}_t)\} && (3.133)\\
&= \mathbb{E}_{X_t}\left\{(\mathrm{tr}(\sigma X_t) - y_t)^2 + \frac{1-y_t^2}{N}\right\} && (3.134)\\
&= \mathbb{E}_{X_t}\{(\mathrm{tr}(\sigma X_t) - y_t)^2\} + \mathbb{E}_{X_t}\left\{\frac{1-y_t^2}{N}\right\} && (3.135)\\
&= \frac{d}{d^2-1}\left(\|\sigma-\rho\|_F^2 + \frac{d-\|\rho\|_F^2}{N}\right), && (3.136)
\end{aligned}$$

where we used the results of Lemmas 10 and 11 in the last step. Notice that as $N \to \infty$, the result of the noiseless case is recovered. □

Consequently, the following result shows that the true state is the optimal state that minimizes the loss function.

Corollary 2. *The state ρ is the unique state that minimizes the expectation of the noisy loss function, where*

$$\mathbb{E}_t\{L_t(\rho)\} = \frac{d}{d^2-1}\frac{d-\|\rho\|_F^2}{N}. \tag{3.137}$$

The following theorem shows how to select an adaptive learning rate that results in convergence of the MEG estimate in probability for noisy measurements.

Theorem 3. *In the presence of noise, the state estimate using the MEG update rule with learning rate*

$$\eta_t = \frac{1}{2}\frac{\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\}}{\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} + 2\left(\frac{d^2-1}{Nd}\right)}, \tag{3.138}$$

converges in probability to the true state, i.e. for all $\delta > 0$,

$$\lim_{t\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 \leq \delta\right\} = 1. \tag{3.139}$$

Proof. We know from Lemma 6 that,

$$\eta_t L_t(\hat{\rho}_t) - \frac{\eta_t}{1-2\eta_t} L_t(\rho) \leq D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1}). \tag{3.140}$$

Taking the expectation with respect to y_t followed by the the expectation with respect to X_t we get,

$$\eta_t \,\mathbb{E}_t\{L_t(\hat{\rho}_t)\} - \frac{\eta_t}{1-2\eta_t}\,\mathbb{E}_t\{L_t(\rho)\} \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_t\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.141}$$

Applying Lemma 12, we get

$$\begin{aligned}\eta_t \frac{d}{d^2-1}\left(\|\hat{\rho}_t - \rho\|_F^2 + \frac{d - \|\rho\|_F^2}{N}\right) - \frac{\eta_t}{1-2\eta_t}\frac{d}{d^2-1}\left(\frac{d-\|\rho\|_F^2}{N}\right) \\ \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_t\{D(\rho||\hat{\rho}_{t+1})\}.\end{aligned} \tag{3.142}$$

Simplifying this expression and taking the expectation with respect to all previous time instants we get

$$\eta_t\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_t^2}{1-2\eta_t}\left(\frac{d-\|\rho\|_F^2}{N}\right) \leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1})\}. \tag{3.143}$$

The second term on the left hand side depends on the purity of the true state, and it can be bounded to become

$$\eta_t\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_t^2}{1-2\eta_t}\left(\frac{d^2-1}{Nd}\right) \leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1})\}. \tag{3.144}$$

Selecting the learning rate to be

$$\eta_t = \frac{1}{2}\frac{\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\}}{\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} + 2\left(\frac{d^2-1}{Nd}\right)}, \tag{3.145}$$

then summing up the inequality over different time steps yields

$$\sum_{t=1}^{T} \frac{1}{4} \frac{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\}^2}{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} + 2\left(\frac{d^2-1}{Nd}\right)} \leq \frac{d^2-1}{d} \left(\mathbb{E}\{D(\rho||\hat{\rho}_1) - D(\rho||\hat{\rho}_{T+1})\}\right) \tag{3.146}$$

$$\leq \frac{d^2-1}{d} D(\rho||\hat{\rho}_1), \tag{3.147}$$

where $\mathbb{E}\{D(\rho||\hat{\rho}_1)\} = D(\rho||\hat{\rho}_1)$ because $\hat{\rho}_1$ and ρ are independent of X_t and $\hat{y}_t$ for any t. Now taking the limit as $T \to \infty$ we get

$$\sum_{t=1}^{\infty} \frac{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\}^2}{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} + 2\left(\frac{d^2-1}{Nd}\right)} \leq 4\left(\frac{d^2-1}{d}\right) D(\rho||\hat{\rho}_1). \tag{3.148}$$

Since, the left-hand side of the inequality is constant, then the series on the right hand side must converge. Consequently using the divergence test,

$$\lim_{t\to\infty} \frac{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\}^2}{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} + 2\left(\frac{d^2-1}{Nd}\right)} = 0. \tag{3.149}$$

Assume that

$$\lim_{t\to\infty} \mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} = K > 0, \tag{3.150}$$

then

$$\lim_{t\to\infty} \frac{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\}^2}{\mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} + 2\left(\frac{d^2-1}{Nd}\right)} = \frac{K^2}{K + 2\left(\frac{d-\|\rho\|_F^2}{N}\right)} \neq 0, \tag{3.151}$$

which contradicts the condition in (3.149). This means that it must be the case that

$$\lim_{t\to\infty} \mathbb{E}\left\{\|\hat{\rho}_t - \rho\|_F^2\right\} = 0. \tag{3.152}$$

Now we can apply Lemma 4 on the random variable $Z_t = \|\hat{\rho}_t - \rho\|_F^2$ to conclude that

$$\forall \delta > 0, \lim_{t\to\infty} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 \leq \delta\right\} = 1. \tag{3.153}$$

Therefore, the estimate $\hat{\rho}_t$ converges in probability to the true state ρ. □

The problem with this choice of learning rate, is that it depends on the true state. This might be useful in other applications like state tracking, but it will not be practical for tomography applications, where the true state is unknown. So, we show next that in fact we can select another form of the learning rate that is

independent of the true state and show a stronger statement of convergence.

Theorem 4. *Let $\delta \in (0,1)$, $\alpha \in \left(0, \frac{1}{2}\right)$ and $\beta \in \left(\frac{1}{2}, 1-\alpha\right)$. If we choose a learning rate of the form*

$$\eta_t = \frac{\eta_0}{t^\beta} \quad with \quad \eta_0 < \frac{1}{2}, \tag{3.154}$$

then there exists T_0 given by

$$T_0 = \left(\frac{\frac{d^2-1}{\eta_0 d}\left(\log d + \frac{2}{N}\frac{\eta_0^2}{1-2\eta_0}\zeta(2\beta)\right) + 2}{\delta} \right)^{\frac{3}{1-\alpha-\beta}}, \tag{3.155}$$

such that for any $T > T_0$ we have,

$$\Pr\left\{ \|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha} \right\} \geq 1 - \delta, \tag{3.156}$$

where the probability is taken over all measurement choices and t uniformly in $\{1, 2, \ldots, T\}$. Moreover,

$$\lim_{T\to\infty} \Pr\left\{ \|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha} \right\} = 1\,. \tag{3.157}$$

Proof. Let the initial estimate be $\hat{\rho}_1 = \frac{I_d}{d}$. We know from Lemma 6 that,

$$\eta_t L_t(\hat{\rho}_t) - \frac{\eta_t}{1-2\eta_t} L_t(\rho) \leq D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1}). \tag{3.158}$$

Taking the expectation with respect to y_t followed by the the expectation with respect to X_t we get,

$$\eta_t \,\mathbb{E}_t\{L_t(\hat{\rho}_t)\} - \frac{\eta_t}{1-2\eta_t}\,\mathbb{E}_t\{L_t(\rho)\} \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_t\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.159}$$

Applying Lemma 12, we get

$$\eta_t \frac{d}{d^2-1}\left(\|\hat{\rho}_t - \rho\|_F^2 + \frac{d - \|\rho\|_F^2}{N} \right) - \frac{\eta_t}{1-2\eta_t}\frac{d}{d^2-1}\left(\frac{d - \|\rho\|_F^2}{N} \right) \leq D(\rho||\hat{\rho}_t) - \mathbb{E}_t\{D(\rho||\hat{\rho}_{t+1})\}. \tag{3.160}$$

Simplifying this expression and taking the expectation with respect to all previous

time instants we get

$$\eta_t \,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_t^2}{1-2\eta_t}\left(\frac{d - \|\rho\|_F^2}{N}\right) \leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1})\}. \tag{3.161}$$

The second term on the left hand side is a function of the purity of the true state (defined as $\|\rho\|_F^2$). This term comes from the variance of the noise which varies according to the location of the state. It can be bounded to become

$$\eta_t \,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_t^2}{1-2\eta_t}\left(\frac{d^2-1}{Nd}\right) \leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_t) - D(\rho||\hat{\rho}_{t+1})\}. \tag{3.162}$$

Summing up the inequality over different time steps we get

$$\sum_{t=1}^{T} \eta_t \,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_t^2}{1-2\eta_t}\left(\frac{d^2-1}{Nd}\right) \leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_1) - D(\rho||\hat{\rho}_{T+1})\} \tag{3.163}$$

$$\leq \frac{d^2-1}{d}\,\mathbb{E}\{D(\rho||\hat{\rho}_1)\} \tag{3.164}$$

$$\leq \frac{d^2-1}{d} D(\rho||\hat{\rho}_1) \tag{3.165}$$

$$\leq \frac{d^2-1}{d}\log d. \tag{3.166}$$

Now, by choosing learning rate in the form

$$\eta_t = \frac{\eta_0}{t^\beta} : \eta_0 < \frac{1}{2}, \tag{3.167}$$

the inequality becomes

$$\frac{d^2-1}{d}\log d \geq \sum_{t=1}^{T} \frac{\eta_0}{t^\beta}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_0^2}{t^{2\beta} - 2\eta_0 t^\beta}\left(\frac{d^2-1}{Nd}\right) \tag{3.168}$$

$$\geq \sum_{t=1}^{T} \frac{\eta_0}{t^\beta}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_0^2}{t^{2\beta} - 2\eta_0 t^{2\beta}}\left(\frac{d^2-1}{Nd}\right) \tag{3.169}$$

$$\geq \sum_{t=1}^{T} \frac{\eta_0}{t^\beta}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_0^2}{1-2\eta_0}\left(\frac{d^2-1}{Nd}\right)\sum_{t=1}^{T}\frac{1}{t^{2\beta}} \tag{3.170}$$

$$\geq \sum_{t=1}^{T} \frac{\eta_0}{t^\beta}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} - \frac{2\eta_0^2}{1-2\eta_0}\left(\frac{d^2-1}{Nd}\right)\sum_{t=1}^{\infty}\frac{1}{t^{2\beta}} \tag{3.171}$$

$$= -\frac{2\eta_0^2}{1-2\eta_0}\left(\frac{d^2-1}{Nd}\right)\zeta(2\beta) + \sum_{t=1}^{T} \frac{\eta_0}{t^\beta}\,\mathbb{E}\{\|\hat{\rho}_t - \rho\|_F^2\} \tag{3.172}$$

$$\geq -\frac{2\eta_0^2}{1-2\eta_0}\left(\frac{d^2-1}{Nd}\right)\zeta(2\beta)+\sum_{t=1}^{T}\frac{\eta_0}{T^\beta}\,\mathbb{E}\{\|\hat{\rho}_t-\rho\|_F^2\}, \tag{3.173}$$

where $\zeta(\cdot)$ is the Riemann zeta function. Now, let $\epsilon_t = \mathbb{E}\{\|\hat{\rho}_t-\rho\|_F^2\}$, $\delta_t = \frac{1}{t^{\alpha+\gamma}}$, and $\gamma = \frac{2}{3}(1-\alpha-\beta)$. Notice that $\alpha+\beta+\gamma<1$ as long as $\alpha+\beta<1$. Define the set

$$\mathcal{T}_\delta := \{t\in\{1,2,..T\} : \epsilon_t \geq \delta_t\}\,. \tag{3.174}$$

Rearranging the terms in the inequality we get

$$\begin{aligned}
\frac{d^2-1}{\eta_0 d}\left(\log d+\frac{2}{N}\frac{\eta_0^2}{1-2\eta_0}\zeta(2\beta)\right) &\geq \frac{1}{T^\beta}\sum_{t=1}^{T}\epsilon_t && (3.175)\\
&\geq \frac{1}{T^\beta}\sum_{t=1}^{T}\left(\mathbb{1}_{\epsilon_t>\delta_t}\right)\epsilon_t && (3.176)\\
&\geq \frac{1}{T^\beta}\sum_{t=1}^{T}\left(\mathbb{1}_{\epsilon_t>\delta_t}\right)\frac{1}{t^{\alpha+\gamma}} && (3.177)\\
&\geq \frac{1}{T^\beta}\sum_{t=1}^{T}\left(\mathbb{1}_{\epsilon_t>\delta_t}\right)\frac{1}{T^{\alpha+\gamma}} && (3.178)\\
&\geq \frac{|\mathcal{T}_\delta|}{T^{\alpha+\beta+\gamma}}. && (3.179)
\end{aligned}$$

In other words, the ratio between the number of iterations in which $\mathbb{E}\{\|\hat{\rho}_t-\rho\|_F^2\}\geq \frac{1}{t^{\alpha+\gamma}}$ and the total number of iterations T we performed so far is bounded by

$$\frac{|\mathcal{T}_\delta|}{T}\leq KT^{\alpha+\beta+\gamma-1}, \tag{3.180}$$

where $K := \frac{d^2-1}{\eta_0 d}\left(\log d+\frac{2}{N}\frac{\eta_0^2}{1-2\eta_0}\zeta(2\beta)\right)$. This implies that

$$\lim_{T\to\infty}\left(\frac{T_\delta}{T}\right)=0, \tag{3.181}$$

because $\alpha+\beta+\gamma<1$. This means that increasing the number of iterations results in decreasing the number of times where the estimate was not accurate enough. Let's state this formally. Assuming we do a total number of iterations T, then if we select at random a fixed time step $1\leq\tilde{t}\leq T$, then there will be two possible outcomes. Either $\epsilon_{\tilde{t}}\leq\delta_{\tilde{t}}$ or $\epsilon_{\tilde{t}}>\delta_{\tilde{t}}$. Assume we get the first outcome, then by applying Markov's inequality,

$$\epsilon_{\tilde{t}}\leq\delta_{\tilde{t}} \implies \Pr\left\{\|\hat{\rho}_{\tilde{t}}-\rho\|_F^2\}\geq\frac{1}{\tilde{t}^\alpha}\right\}\leq\mathbb{E}\left\{\|\hat{\rho}_{\tilde{t}}-\rho\|_F^2\right\}\tilde{t}^\alpha \tag{3.182}$$

$$\leq \delta_{\tilde{t}} \tilde{t}^{\alpha} \tag{3.183}$$

$$= \tilde{t}^{-\gamma}. \tag{3.184}$$

Now, we can find the joint probability

$$\begin{aligned}
\Pr_{t,\hat{\rho}_t}\left\{\|\hat{\rho}_t - \rho\|_F^2 \geq \frac{1}{t^{\alpha}}\right\} &= \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2\} \geq \frac{1}{t^{\alpha}}\middle| t = \tilde{t}\right\} \Pr\left\{\tilde{t} \in \mathcal{T}_{\delta}\right\} \\
&\quad + \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2\} \geq \frac{1}{t^{\alpha}}\middle| t = \tilde{t}\right\} \Pr\left\{\tilde{t} \notin \mathcal{T}_{\delta}\right\} && (3.185) \\
&\leq \Pr\left\{\tilde{t} \in \mathcal{T}_{\delta}\right\} + \sum_{\tilde{t} \notin \mathcal{T}_{\delta}} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2\} \geq \frac{1}{t^{\alpha}}\middle| t = \tilde{t}\right\} \frac{1}{T} && (3.186) \\
&\leq \frac{|\mathcal{T}_{\delta}|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T} \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2\} \geq \frac{1}{t^{\alpha}}\middle| t = \tilde{t}\right\} && (3.187) \\
&\leq \frac{|\mathcal{T}_{\delta}|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T} \tilde{t}^{-\gamma} && (3.188) \\
&\leq \frac{|\mathcal{T}_{\delta}|}{T} + \frac{1}{T}\sum_{\tilde{t}=1}^{T} \left(\frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}}. && (3.189)
\end{aligned}$$

Applying Jensen's inequality on the second term (noting that $f(x) = x^r$ is a concave function for $0 < r < 1$). Thus,

$$\begin{aligned}
\Pr_{t,\hat{\rho}_t}\left\{\|\hat{\rho}_t - \rho\|_F^2 \geq \frac{1}{t^{\alpha}}\right\} &= \frac{|\mathcal{T}_{\delta}|}{T} + \left(\frac{1}{T}\sum_{\tilde{t}=1}^{T} \frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}} && (3.190) \\
&\leq \frac{|\mathcal{T}_{\delta}|}{T} + \left(\frac{1}{T}\sum_{\tilde{t}=1}^{\infty} \frac{1}{\tilde{t}^2}\right)^{\frac{\gamma}{2}} && (3.191) \\
&= \frac{|\mathcal{T}_{\delta}|}{T} + \left(\frac{\pi^2}{6}\frac{1}{T}\right)^{\frac{\gamma}{2}} && (3.192) \\
&\leq \frac{|\mathcal{T}_{\delta}|}{T} + 2T^{-\frac{\gamma}{2}} && (3.193)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^{\alpha}}\right\} &\geq 1 - \frac{|\mathcal{T}_{\delta}|}{T} - 2T^{-\frac{\gamma}{2}} && (3.194) \\
&\geq 1 - KT^{\alpha+\beta+\gamma-1} - 2T^{-\frac{\gamma}{2}} && (3.195) \\
&= 1 - KT^{-\frac{1-\alpha-\beta}{3}} - 2T^{-\frac{1-\alpha-\beta}{3}} && (3.196) \\
&= 1 - T^{-\frac{1-\alpha-\beta}{3}}\left(K + 2\right). && (3.197)
\end{aligned}$$

Now, let

$$T_0 = \left(\frac{K+2}{\delta}\right)^{\frac{3}{1-\alpha-\beta}}, \tag{3.198}$$

then, if choose $T > T_0$, then

$$\delta \geq T^{-\frac{1-\alpha-\beta}{3}}(K+2), \tag{3.199}$$

or,

$$1-\delta \leq 1 - T^{-\frac{1-\alpha-\beta}{3}}(K+2) \tag{3.200}$$

$$\leq \Pr\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\}. \tag{3.201}$$

Now, taking the limit as $T \to \infty$ we obtain finally that,

$$\lim_{T\to\infty} \Pr_{t,\hat{\rho}_t}\left\{\|\hat{\rho}_t - \rho\|_F^2 \geq \frac{1}{t^\alpha}\right\} = 0, \tag{3.202}$$

or, equivalently,

$$\lim_{T\to\infty} \Pr_{t,\hat{\rho}_t}\left\{\|\hat{\rho}_t - \rho\|_F^2 < \frac{1}{t^\alpha}\right\} = 1. \tag{3.203}$$

□

3.4.4 Convergence of the Noisy Measurements case with Averaging

As discussed previously, doing the running-average over the measurements with a small number of shots is equivalent to increasing the number of shots without having to do this experimentally per each measurement. This method also does not require the use of an adaptive learning rate. So, given the data point $(X_t, \hat{y}_t)$, we calculate the running average $\bar{y}_t$:

$$\bar{y}_t = \frac{\hat{y}_{r_1} + \hat{y}_{r_2} + ...\hat{y}_{r_{n-1}} + \hat{y}_t}{n_{X_t}} = \frac{(n_{X_t}-1)\bar{y}_{t-1} + \hat{y}_t}{n_{X_t}}, \tag{3.204}$$

such that $\{r_i\}_{i=1}^{n_{X_t}} = \{t' : X_{t'} = X_t\}$ are the time indices in which the measurement operator X_t appeared before (which means that $r_n = t$), and n_{X_t} is the number of times it appeared until time t. If we are choosing the measurement operators randomly then after enough number of iterations we may assume that we visited all operators the same number of iterations. So as $t \to \infty, n_{X_t} \to \infty$. Now from the

strong law of large numbers:

$$\frac{1}{n_{X_t}} \sum_{i=1}^{n_{X_t}} \hat{y}_{r_j} \to \mathbb{E}\{\hat{y}_t\} = y_t \quad a.s. \tag{3.205}$$

So the gradient of the loss function satisfies that:

$$\nabla L_t(\hat{\rho}_t) = 2(\mathrm{tr}(\hat{\rho}_t X_t) - \bar{y}_t)X_t \quad \to \quad 2(\mathrm{tr}(\hat{\rho}_t X_t) - \mathrm{tr}(\rho X_t))X_t \quad a.s. \tag{3.206}$$

In other words, after enough number of iterations, the situation becomes similar to the noise-free measurements case which allows the possibility of convergence to the true state with a constant learning rate.

3.5 Simulation Results

This section discusses the methods and results of the numerical simulations. An overview of the simulations settings is given first, followed by discussion on the significance of the results.

3.5.1 Methods

In order to assess the performance of the proposed method, we created a dataset consisting of 1000 randomly generated quantum states for 1-, 2-, 3-, 4-, and 5-qubit systems, as well as simulating 100000 random measurement outcomes for 10, 100, 1000, 10000 shots for each of these states. The estimate after each measurement is calculated, and compared to the true state using the infidelity measure defined as

$$1 - F(\rho, \hat{\rho}_t) = 1 - \left(\mathrm{tr}\left|\sqrt{\rho}\sqrt{\hat{\rho}_t}\right|\right)^2. \tag{3.207}$$

Figure 3.2 shows the behavior of MEG under different learning rates in the form $\eta_t = 0.5t^{-\beta}$, compared to using the running average (RA) method with a constant learning rate. The plot shows that using the running average leads to the fastest convergence compared to the case of adaptive learning rate. So, we choose the RA method for further discussion.

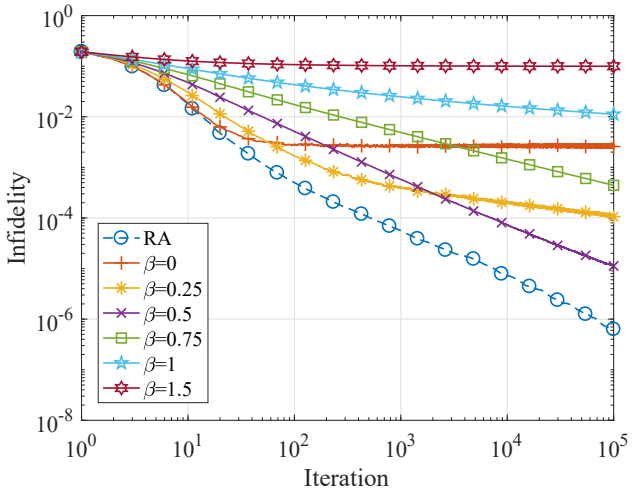


Figure 3.2: Simulation results for MEG estimation for a single-qubit system and 100 shots per measurement. The infidelity of the proposed matrix exponential gradient (MEG) method is averaged over 1000 randomly generated quantum states and plotted versus the iteration number. The plot is for the running average case, as well as the variable learning rate $\eta_t = 0.5t^{-\beta}$ for different values of β

In addition to the matrix exponential gradient (MEG) estimator, the least squares (LS) method in [21] and the diluted maximum likelihood (MLE) in [20] are also implemented and used for comparison in the setting of 1-, 2-, 3-, and 4-qubit systems. In these simulations, the learning rate of the MEG rule is 0.5. For the maximum likelihood method, the iteration step parameter ϵ (controlling the dilution) is taken to be 0.1. Since this value is much smaller than 1, it is guaranteed that after each internal iteration, the likelihood is increased as proved in [20]. The number of internal MLE iterations is chosen to be 10, which is a small number to reduce the total runtime of this method. In other words, for every new data point, we recalculate the MLE estimate starting from the previous estimate using 10 iterations, and then evaluate the infidelity. An optimal setting would be a variable number of internal iterations that starts out large and decreases afterwards. However, it should be noted that in this work the objective is not optimizing the implementation of the MLE, but to have the simplest implementation for comparison purpose. Additionally, we are interested more in the asymptotic behavior of the estimators. So, after a large number of data points, the estimate will be very near the true state. Consequently, there will be no need to have a large number of MLE internal iterations at that stage. The source code is publicly available[1]. Figure 3.1 shows the infidelity versus the number of iterations for 1-, 2-, 3- and 4-qubits when the number of shots per measurement is taken to be 1000. Figure **??** shows the performance for a 4-qubit system at different number of measurement shots. For 5-qubit systems, only the performance of MEG is assessed as shown in Figure 3.4.

[1] `https://github.com/akramyoussry/MEG_online_QST`

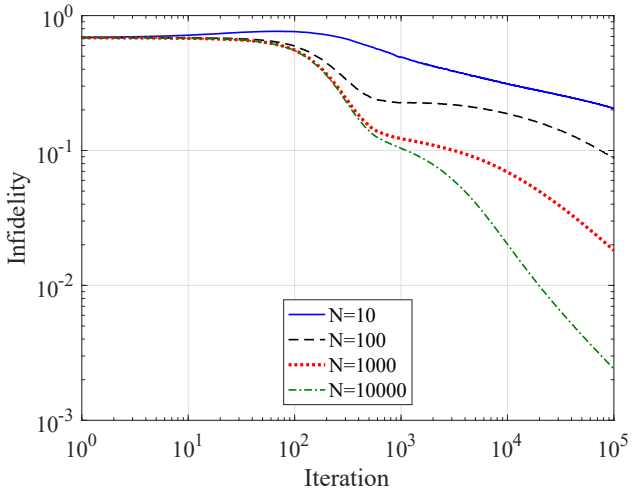


Figure 3.4: Simulation results for a five-qubit system. The infidelity of the proposed matrix exponential gradient (MEG) method is averaged over 1000 randomly generated quantum states and plotted versus the iteration number. The number of shots per measurement is taken to be 10, 100, 1000, and 10000 shots.

Additonally, we compared with the projected-gradient method (PGD) proposed in [44] in the batch setting. We implemented this method in the online setting, and investigated its performance numerically in this case. The estimate at time iteration $t+1$ is given by

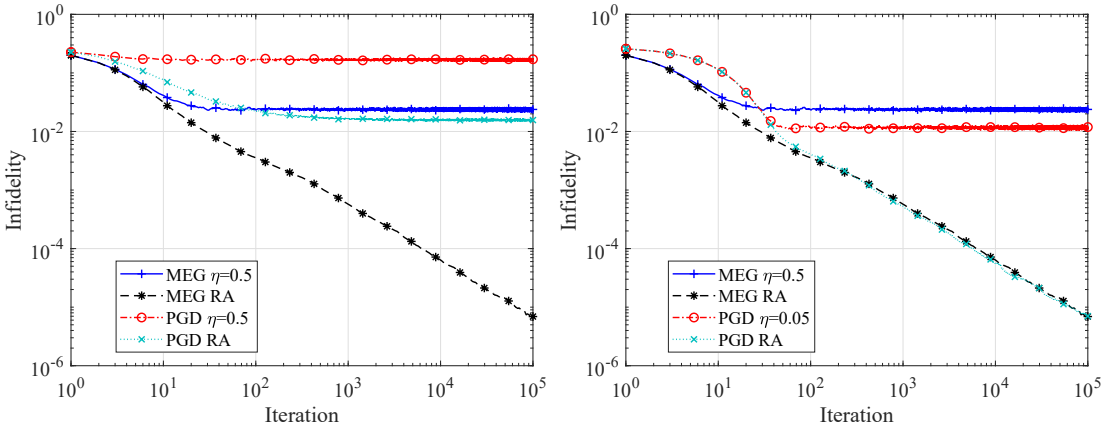
$$\hat{\rho}_{t+1} = \mathcal{P}\left(\hat{\rho}_t - \eta \nabla L_t(\hat{\rho}_t)\right), \tag{3.208}$$

where η is the learning rate, L_t is the loss function as defined previously, and $\mathcal{P}$ denotes projection into the physical space as discussed in Section 2.4.1.2.

3.5.2 Discussion

The maximum likelihood method is a batch method that requires that the whole dataset is available for post-processing. So, if a new measurement is done, the entire algorithm must be repeated again from the beginning. Additionally, the storage requirement of the data operators may be large, especially for multi-qubit systems. Our proposed method does not need to store all the data set, just the last averaged outcome for each measurement operator in the most sophisticated case. The same comparison applies to least-squares, which also acts on the whole batch of data and is not an online algorithm.

An additional advantage of our algorithm is that it guarantees positivity of the estimated operator at all times. Least-squares and similar approaches are not guaranteed to produce a physical state unless a further step of projection back to the physical space is done. This forms an additional choice and overhead on the algorithm. Moreover, the use of running average allows using a constant learning rate.



(a) PGD with learning rate of 0.5 (b) PGD with learning rate of 0.05

Figure 3.5: Simulation results for comparing MEG with constant learning rate of 0.5 and MEG with running averages to the projected-gradient descent method (PGD) with constant learning rate as well as running averages for learning rate of (a) 0.5 and (b) 0.05. The infidelity is averaged over 1000 randomly generated quantum states and plotted versus the iteration number. The number of shots per measurement is taken to be 10 shots.

This solves the problem of having to evaluate the optimum learning rate at each time step.

For the projected-gradient method, it turns out the performance is highly dependent on choosing the learning rate, and generally seems very similar to MEG when the learning rate is chosen low. In Figure 3.5a, we compared MEG with constant learning rate, MEG with running average (RA), PGD with constant learning rate and PGD with running average. The same value of $\eta = 0.5$ was used in the four of them. This plot shows that the MEG method has better convergence. On the other hand, by changing the step size of the PGD methods to 0.05, we see that both methods seem to have similar performance for the running average case after significant number of iterations as shown in Figure 3.5b. This makes it very difficult to give a fair comparison with MEG. Note that even for MEG we were not particularly concerned with finding the optimal learning rate — we simply tested a few learning rates that are compatible with the limitations given by the convergence proof. However, faster (but not provable) convergence might be possible, as it often is, if we go outside that range. For PGD we simply do not know what the restrictions on the learning rate are so that the algorithm still provably converges with similar parameters as MEG. Thus, it will be interesting as a future work to look into the convergence of the PGD method.

Considering the accuracy of the estimate, the simulation results show that after a sufficient number of iterations, the MEG estimates converge to both the maximum-likelihood and least squares estimates which are considered the optimal estimators in

batch processing systems. MLE produces a point estimate for the model that maximizes the probability of the observed data, while LS minimizes the sum of squared errors due to observation noise. As the number of shots increase, the accuracy of all estimators gets better (i.e lower average infidelity for a given number of iterations) because the noise becomes less effective. On the other hand, as the number of qubits gets higher, more iterations are needed to achieve a low average infidelity. This is because at each iteration one basis is selected randomly for measurement. However, for high-dimensional systems there are many more bases that need to be covered to form a complete set ($d^2 - 1$ bases).

As for complexity, maximum-likelihood scales as $O(d^4)$. This is because the bottleneck operation is calculating the gradient of the log-likelihood function $R = \sum_j \frac{f_j}{NPr_j}\Pi_j$. For a complete set of measurement, at least $d^2 - 1$ measurement operators are needed, each of dimension $d \times d$. So, this implies that calculating R requires $O(d^4)$ complex multiplication operations. For the least-squares method, the complexity is $O(d^4)$ as discussed in [21]. In this case the bottleneck operation is the matrix multiplication part X^TY of the estimation equation $\hat{\theta} = (X^TX)^{-1}X^TY$. That is because again for a complete set of measurements we need at least $d^2 - 1$ operators, and thus Y is of dimensions $(d^2 - 1) \times 1$, and X is of dimensions $(d^2 - 1) \times (d^2 - 1)$. Finally, for the proposed method, the bottleneck is in calculating the matrix exponential. The complexity will depend on the particular way of implementation. The most common way is by performing eigendecomposition, followed by exponentiating the diagonal matrix of eigenvalues. In this case, the complexity is usually assumed to be $O(d^3)$ [57, 58]. It should be noted that the complexities discussed here are obtained per iteration, i.e for each update given a new data point. Table 3.1 summarizes these results.

Table 3.1: Summary of runtime complexities per iteration for the MLE, LS, and MEG algorithms

Algorithm	**Runtime**
MLE	$O(d^4)$
LS	$O(d^4)$
MEG	$O(d^3)$

In order to verify the claim that MEG should have the fastest performance, the execution times per 1 iteration were recorded in the simulation for the three methods. Figure 3.6 shows the average of these execution times. It is clear that as the number of qubits increases, MEG has the least runtime compared to the other two methods.

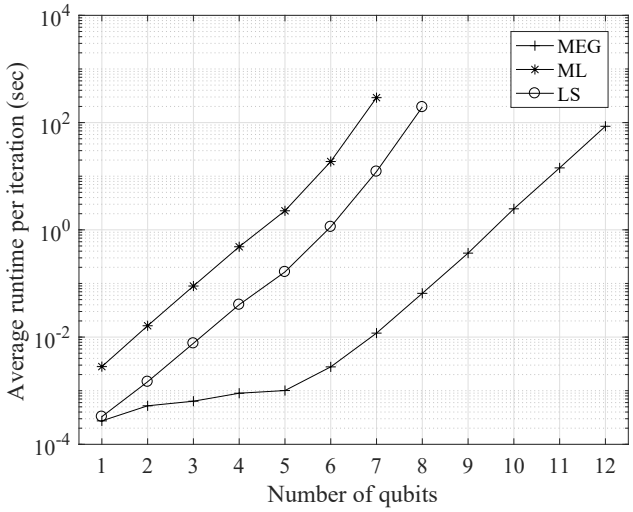


Figure 3.6: The average runtime of the update step for the maximum-likelihood (MLE), least-squares (LS), and matrix-exponentiated gradient (MEG) methods, measured for increasing number of qubits.

3.6 Conclusion

In this chapter, we introduced the idea of using the running average on the noisy measurements together with the MEG update rule to construct a fast and simple online quantum state estimator. We proved convergence of the proposed algorithm using information-theoretic tools for both ideal and noisy measurements. We also showed numerically the accuracy and efficiency of the estimator. The focus of this chapter was on characterization of quantum states. In the next two chapters, we will focus on characterization as well as control of quantum processes. We will start first by presenting a novel deep learning based framework suitable for modelling quantum systems, and apply it to closed evolution dynamics.

Chapter 4

Modelling and Control of Closed Quantum Systems

Abstract The complexity of experimental quantum information processing devices is increasing rapidly, requiring new approaches to control them. In this chapter, we address the problems of practically modeling and controlling an integrated optical waveguide array chip—a technology expected to have many applications in telecommunications and optical quantum information processing. This photonic circuit can be electrically reconfigured, but only the output optical signal can be monitored. As a result, the conventional control methods cannot be naively applied. Characterizing such a chip is challenging for three reasons. First, there are uncertainties associated with the Hamiltonian model describing the chip. Second, we expect distortions of the control voltages caused by the chip's electrical response, which cannot be directly observed. And third, there are imperfections in the measurements caused by losses from coupling the chip externally to optical fibers. We have developed a deep neural network approach to solve these problems. The architecture is designed specifically to overcome the aforementioned challenges using a Gated Recurrent Unit (GRU)-based network as the central component. The Hamiltonian is estimated as a blackbox, while the rules of quantum mechanics such as state evolution is embedded in the structure as a whitebox. The resulting overall graybox model of the chip shows good performance both quantitatively in terms of the mean square error and qualitatively in terms of the shape of the predicted waveforms. We use this neural network to solve a classical and a quantum control problem. In the classical application we find a control sequence to approximately realize a time-dependent output power distribution. For the quantum application we obtain the control voltages to realize a target set of quantum gates. The method we propose is generic and can be applied to other systems that can only be probed indirectly.

4.1 Introduction

The complexity of experimental quantum information processing devices is increasing rapidly, requiring new approaches to control them. Noisy Intermediate-Scale Quantum (NISQ) devices are emerging nowadays, with lots of experimental challenges [1,2,59]. In this present work, we deal with the problem of modelling a device that can process some input signals to generate output signals, and the operation of the device can be manipulated using control signals. There are three possible methods to model such a device presented as follows.

The first approach is through direct physical modelling. We look for a mathematical description of the output signals expressed in terms of the input and control signals. The equations will involve some unknown parameters which should be chosen to match the performance of an actual realization of the device. And thus, we perform measurements on the device and use methods of parameter estimation in order to find the unknown parameters of the model. We call this approach a whitebox approach. This would be the first approach one would try to use. The problem however is that if there are uncertainties in the relations between some variables, or some assumptions are made to derive some formulas (which might not be true for an actual device), then the resulting model might not be accurate enough to fit and predict actual measurements. Imperfections in the measurement process will also decrease the accuracy of the obtained model. Additionally, relations between some variables may be completely unknown and thus the problem becomes not just estimating parameters but also estimating functional forms (maps between variables). Moreover, there are situations where estimating the unknown parameters requires measurements that are not experimentally possible. For instance, if we want to estimate the parameters of a transfer function of an electrical circuit, then we will need to measure voltages at some nodes of the circuit. However, if we cannot physically access those nodes then the problem becomes more difficult. Finally, the complexity of the problem increases if the physical models involve non-linear relations. Thus, the whitebox approach might face lots of challenges in practical situations.

The second way to solve the problem is through the blackbox approach. We do not obtain a set of physical equations describing the device, but rather we construct a generic function that approximates the relationship between the output and the input and control signals. This is usually a highly non-linear function with a large number of parameters that can be estimated using the measurements. If the function is complex enough, then it can model and predict any unknown relations between variables. For that type of modelling, machine learning structures, such as artificial neural networks, are very suitable. This approach has an advantage of being capable of predicting the output signals given the input and control signals. However, there

are few drawbacks. First, the resulting model provides the least amount of information about the physics of how the device works. And so it would be difficult to use the model for re-engineering the device if required. Second, the resulting accuracy may not be as high as expected. This is because the structure does not have any prior information about the map between inputs and outputs, and so it might need to "discover" some complicated laws of physics (such as the evolution of quantum systems) beside other unknown relations. This makes the training process harder. Consequently, a larger dataset and higher number of iterations would be needed to reach a good level of accuracy, which might turn out to be impractical.

The last approach is a combination of the other two approaches, we would refer to as a graybox model. In this case, we use direct physical modelling for parts of the description that we have complete certainty about (whitebox part), while we use a blackbox for the other parts that we are uncertain about. The model should be built such that the measurements required for the learning process are physically available; there is enough physical modelling through the whiteboxes to allow extracting useful information about the behaviour of the device; and any measurement imperfections should be accounted for. Machine learning structures are also suitable for this type of modelling. Standard machine learning layers would be used for the blackboxes. However, we also need to define non-standard layers to account for the whiteboxes, and these are application specific. The overall structure should be consistent to allow standard learning algorithms to work. In this chapter, we explore the use of hybrid deep learning architecture to solve problems related to experimental modelling and control of quantum systems. Although the focus is on a photonic device that will be introduced shortly, the graybox approach can be considered very general, applying to many situations where there is a system that cannot be probed arbitrarily as discussed.

An example of this approach is when we have a quantum device described by a quantum system. The input signal is modelled by the initial quantum state. The output signal is modelled by a measurement performed on the system after evolving according to a given Hamiltonian. The control signals would then be some external forces applied to the system, and they are modelled by some terms in the Hamiltonian. Using the laws of quantum mechanics, we could write down the relation between the input, output, and control. This would correspond to the whitebox part of the model. Some of the terms inside the Hamiltonian might be unknown, so we would use a blackbox to evaluate these terms. The resulting overall model is then a graybox model. This is a useful approach because it still gives an insight on the physics of the device and one can evaluate physically significant quantities. Additionally, the terms that reduce the accuracy of the models, due to inaccurate physical modelling, are now replaced by blackboxes, resulting in a more

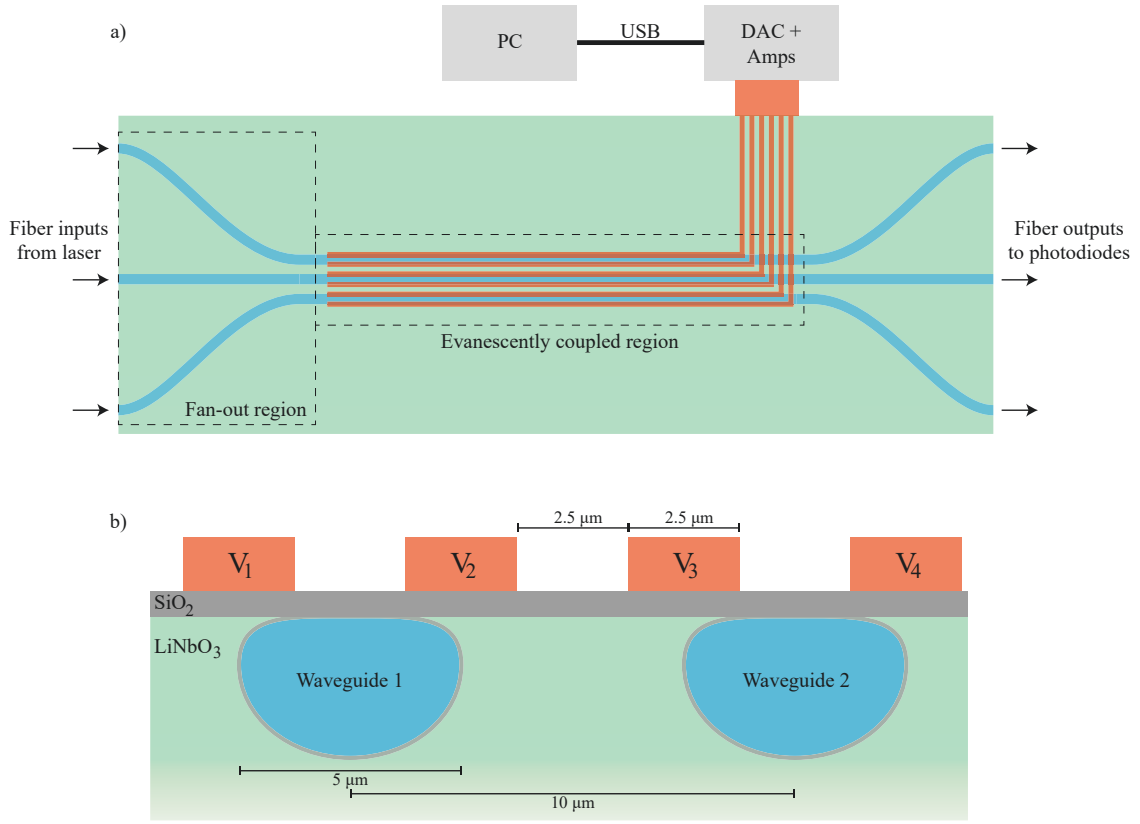


Figure 4.1: a) Top view and b) cross section schematics of a three waveguide reconfigurable array. The waveguides initially fan-in from 127μm spacing to enable coupling to optical fibres. The waveguides in the array are separated by 10μm which enables nearest-neighbour evanescent coupling. The electric field between the electrodes causes a local change in refractive index to the waveguide or the cladding.

accurate overall model.

In this chapter, we focus on a particular system, currently being developed by some of the authors, which is an array of nearest neighbour coupled waveguides with a reconfigurable Hamiltonian. Characterizing such a chip is a significant challenge as will be discussed later. The device we consider is an array of nearest neighbour coupled waveguides that implements a continuous time quantum walk on photons propagating along the array [60,61]. In all previous work, static quantum walks were studied with fixed coupling parameters. Here, we demonstrate a reconfigurable waveguide array by exploiting the electro-optic control of Lithium Niobate. The waveguides are fabricated by reverse proton exchange and we apply local electric fields to change the properties of the coupled array. Figure 4.1 shows the schematic of the chip. We inject laser light into one input waveguide of the array and measure the output optical power distribution across all the waveguides. The electrodes can be controlled to alter the output distribution.

Numerical simulations of such a device shows a host of potential applications. The chip can operate as a classical device with possible applications in telecom-

munications such implementing a Mach-Zehnder interferometer or an electro-optic modulator. Being able to characterize and control such a device is important and has a strong economic impact, but at the same time is very challenging as will be discussed later. Additionally, the chip can work as a quantum device. This includes operating as a quantum router, where single photons can be directed to propagate and be detected at one of the output ports by dynamically changing the control voltages. It can also be used to generate complicated quantum states (such as W state) and realize different quantum gates on single or multiple photons.

We focus on two different scenarios of using this device. The first is when experimentally we are only measuring powers at the output of the chip. This is also equivalent to a single-photon experiment where we only detect photons at the outputs. In this situation, we can use classical modelling only. However, we use a quantum model for two reasons. First, to show the applicability of the graybox approach when the whitebox parts are quantum. The second reason is that although we only measure powers at the output, we allow for arbitrary states in the input, including entangled states which cannot be described by a pure classical model. Also, if there are multiple input photons at different waveguides, then a quantum mechanical description of the chip is required to describe the correlations of the photons at the output [61]. The second application is when we can also measure phases through Mach-Zehnder type of interferometry. In this case, we show the possibility of implementing single-qubit quantum gates with high fidelity, where the qubits live in the subspace formed of the two far-end waveguides. The proposed framework allows finding the set of control voltages required to obtain a target sequence of quantum gates, given the different challenges faced during the characterization process.

Machine learning has been a very active area of research recently, with focus on both the algorithms as well as the wide range of applications touching every field of science and beyond. Deep learning has particularly gained attention as it becomes more and more feasible. This is due to today's enormous computational power, as well as the availability of big datasets for training. The survey [62] covers the common architectures used in deep learning and the range of possible applications.

The physics community is also currently exploring the use machine learning to solve some practical problems faced in designing, controlling, and automating experiments. Some examples of recent work include the automated design of quantum optical setups [63] using reinforcement learning [64], and using deep learning and genetic algorithms [65]. Machine learning was also used in [66] to configure an optical signal processor, which itself can work as an artificial neural network with linear activation functions. Deep learning was also used in Ref. [67] to discover and characterize topological phases of matter and phase transitions. Techniques of both deep learning and reinforcement learning have been applied in quantum control [68–70].

These works differ from ours by treating the entire learned model, including quantum dynamics, as a blackbox, with no detailed modeling of an experimental realization.

Methods of machine learning have also been used in other areas of quantum information. For example, the work presented in [49, 71] is about developing online quantum state estimation algorithms inspired by the matrix exponentiated gradient method, a technique used in classical machine learning. Other applications include the use of neural networks in quantum cryptography [72] and in quantum error correction [73]. Another related problem to what we present here is Hamiltonian learning [24–27]. This is a Bayesian framework that allows updating the priors on Hamiltonian estimates given observed measurements. This approach although very useful on its own, is not suitable for the problem under consideration. The reason is that we are interested in estimating the map between control voltages and the Hamiltonian through indirect measurements. The Bayesian approach is suitable if the Hamiltonian is fixed (the control voltages are fixed). Another Bayesian approach is presented in [74] where the focus is on the real-time prediction of the set of optimal measurements to perform on a quantum dot, using partial information available so far. This allows efficient characterization of the device.

The structure of the remainder of the chapter is as follows. The discussion starts with an overview on the quantum-mechanical description of the chip in Section 4.2.1, followed by the experimental constraints and challenges in Section 4.2.2. Next, in Section 4.3 we present the proposed deep learning architecture in detail. After that, we present the numerical results of the simulations and discuss their significance in Section 4.4. Finally, we end with the conclusion in Section 4.5. Section 4.6 contains figures related to Section 4.4 placed there for maintaining the readability and continuity of the text.

4.2 Problem Setup

This section starts with describing quantum mechanically the photonic circuit we are trying to model and control, followed by the challenges we face in characterizing it experimentally.

4.2.1 Chip Model

The chip with n-waveguides can be described quantum mechanically in $\mathbb{C}^n$ Hilbert space, with the computational basis encoding the presence of photons in each waveguide. For example for $n = 3$ the state $|0\rangle = [1, 0, 0]^T$ encodes a photon present at the first waveguide, the state $|1\rangle = [0, 1, 0]^T$ encodes a photon in the second waveguide and so on. The evolution of the system represents the behaviour of the chip when

light propagates along the waveguides. So, the initial state of the system represents the mode distribution at the inputs of the waveguides, while the final state represents the distribution at the output of the waveguides. For example, if the system evolves from the state $|0\rangle$ to the state $|1\rangle$, then this means that we started with injecting a photon at the first waveguide (at one end of the chip), and the photon got perfectly transferred to the second waveguide after propagating along the chip until the output. This evolution can be described by the unitary

$$U = e^{-iHl}, \tag{4.1}$$

where l is the length of the chip, and H is the Hamiltonian of the chip. In general, we can write the Hamiltonian in the form

$$H = H_0 + H_I(\mathbf{v}), \tag{4.2}$$

where H_0 is the zero-voltage Hamiltonian, and H_I is the interaction Hamiltonian which is a function of the voltages $\mathbf{v}$ applied on the electrodes. Note that the control voltages are time-dependent, however, the time scale of the change is much slower than the time scale of the photon travel across the chip. That is, each photon can see only one time-independent Hamiltonian from the moment it enters the chip until the moment it reaches the output. But the next photon to arrive can experience a different Hamiltonian. This assumption is plausible since it is impossible to change the voltage faster than the flight time of the photon in the chip. This is what allows us to write the evolution as the matrix exponential of the Hamiltonian as in Equation 4.1, without the time-ordering operator.

In the basic experimental setup we can only measure output power distribution. For example, for an $n = 3$ chip, if the input state is $|0\rangle$, and the output state after evolution is $U|0\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle$, then the output distribution we measure is $(|\alpha|^2, |\beta|^2, |\gamma|^2)$. However, to characterize a fully quantum model, we need to measure phases at the output. One of the convenient ways experimentally to measure relative phase shifts between two optical paths is through Mach-Zehnder interferometry as shown in Figure 4.2. Recall the basic idea is to construct a quantum circuit whose output probability amplitude depends on the phase shift required to be measured. With an initial state $|0\rangle$, a standard calculation shows that the final state after the beamsplitter at the bottom-right of the diagram is

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(\frac{\alpha + e^{i\theta}}{\sqrt{2}}|0\rangle + \beta|1\rangle + \gamma|2\rangle + \frac{\alpha - e^{i\theta}}{\sqrt{2}}|3\rangle\right) \tag{4.3}$$

Now, if we measure the power at the detector, we get $P(\theta) = \frac{1}{4}|\alpha + e^{i\theta}|^2$. Now if we

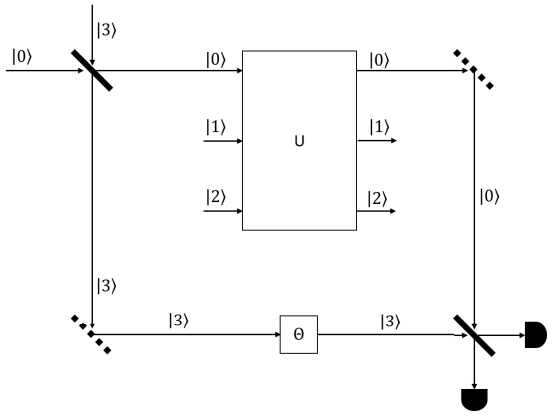


Figure 4.2: A schematic for a typical Mach-Zehnder interferometer to measure the phase shift between the inputs and outputs of the photonic chip. The kets on the optical paths represent the encoding along this direction. The dotted lines are mirrors, while the solid thick lines are beamsplitters.

do two of such measurements corresponding to values of $\theta = 0$ and $\theta = \frac{\pi}{2}$, we can exactly calculate both the amplitude and phase of this output. Particularly,

$$|\alpha|^2 + 1 + 2|\alpha| \cos \angle\alpha = 4P(0) \tag{4.4}$$

$$|\alpha|^2 + 1 + 2|\alpha| \sin \angle\alpha = 4P\left(\frac{\pi}{2}\right), \tag{4.5}$$

where $\angle\alpha$ denotes the phase of α. These two equations can be solved simultaneously to find the amplitude and phase of α. Now, the procedure can be repeated by placing the mirror at the top-right of the diagram at all other outputs of the chip and obtain the amplitude and phase of this part of the state. Since we have an n-dimensional pure state, it is completely defined by $2n$ degrees of freedom corresponding to real and imaginary part of each coefficient. (In fact, only $2n-2$ are needed since we have the normalization constraint, and a non-significant global phase shift). The same procedure can be executed to characterize the output state when other inputs are activated. Finally, it is worth mentioning that this setup for measuring phase is not the only possible way, there might be a more efficient way to measure the phases at the output without requiring to move the optical components spatially. This is however out of the scope of this chapter.

4.2.2 Experimental Challenges

There are many experimental challenges faced when characterizing a fabricated chip, as well as designing the control voltages to implement some desired behaviour. Any model for the device should account for these constraints. These challenges are listed as follows.

1. The drifting in the measured output optical power.

   This problem is caused by charges getting trapped at the interface between the Silicon Dioxide and the Lithium Niobate [75–77]. These charges have very low mobility and therefore take a long time to accumulate and a long time to diffuse when the voltage is removed. These trapped charges are the central reason we have difficultly controlling and characterizing this device. The long diffusion time results in the voltage never 'resetting' to zero. It then becomes extremely difficult to infer what electric field is being applied to the waveguide. In any case, the chip has some equivalent electrical circuit model. But this is difficult to model and characterize experimentally, as we cannot measure physically the voltages the chip actually senses when we apply externally some control voltages. The only available measurements are the output waveguide power distribution, which depends non-linearly on the control voltages. This makes the problem a non-linear control and estimation problem and that is classically difficult to solve. These effects cannot be neglected as well because the distortions in the control voltages will be reflected on the measured power distribution. It will also have a memory effect in the sense that when we apply some control pulse, the output power will be affected by that pulse in addition to the previous pulses that were applied. This means that if at some point in time we set all the control voltages to ground, we will still observe variation of the power distribution in time. The classic way of overcoming this problem is during fabrication by etching the buffer layer between the electrodes [75]. However, for the particular chip we are working with, the dimensions are very small and technologically it is difficult to do this process. Thus, this problem has to be addressed differently. Therefore, the model should account for these unknown distortions, and it should be trainable using only available power measurements.

2. The uncertainties regarding the structure of the Hamiltonian.

   Usually, the Hamiltonian in these chips is assumed to have a tridiagonal form reflecting the fact that only adjacent waveguides are coupled [61,78]. But there is a possibility that there are non-negligible higher order couplings between the waveguides leading to more off-diagonal terms. Also, one could assume the linear dependence of the Hamiltonian on the control voltages. But this assumption might not be true as there might be higher order terms. Thus, the model should not assume any particular form of the Hamiltonian except that it is Hermitian as required by quantum mechanics.

3. The power losses at the output.

Losses in the measured power occur due to the coupling of the chip to the external optical fibres connected to the photodetector. These will cause inaccuracies in the measurements affecting any parameter estimations. These losses also have to be characterized so that we can make corrections for the detected power signals. We will model the losses by

$$\hat{P}_k = \frac{\epsilon_k P_k}{\sum_{i=1}^{n} \epsilon_i P_i}, \tag{4.6}$$

where $\hat{P}_k$ is the k^{th} normalized measured power at waveguide k, and P_i is the actual power at the output of the chip for waveguide i. The normalization is for making the measurements constitute a probability distribution. The model should account for these losses.

4. The limitation on the control voltages.

   Generally, in order to obtain some target output for the device, we need control voltages that can be arbitrarily large. However, if the potential difference across any pair of electrodes exceeds some maximum value, the device will break down. It might be the case that within this limitation one cannot obtain the target with infinite precision. This controllability issue is a different problem and is a subject of the future work of this thesis. And so, any control algorithm should try to maximize the accuracy of the target output without exceeding the allowed range for the control voltages.

As a result of all the previous challenges, estimating and controlling the Hamiltonian directly from measured data is very difficult using the whitebox approach. The complete blackbox might perform well but as discussed it will not give physical insights on the device. Thus, we will seek the graybox approach for modelling the chip under the aforementioned constraints. The blackbox part will represent the map between the Hamiltonian and the control voltages. This allows getting rid of any assumptions on the Hamiltonian as well as accounting for the pulse distortions. The whitebox part will represent the other certain relations derived from quantum mechanics. The next section will give more details about how to construct such a model using deep learning.

4.3 Methods

In the previous section we described the challenges we face in experimentally characterizing the chip if we use conventional methods of model and parameter estimation. In order to address all these challenges, we propose to use a completely data-driven

approach rather than a parametric approach. We are going to use graybox model where the Hamiltonian will be treated as a blackbox, while the quantum evolution and quantum measurement will be treated as whitebox. This is because all the uncertainties are in the Hamiltonian, while the all the laws of quantum mechanics are known. We will design a deep learning structure to implement this idea. The problem will be divided into two stages. The first stage, a set of known control voltages and corresponding power distribution will be used by a supervised deep learning algorithm to find a complete graybox model for the chip. The second stage will be creating another deep learning structure to find the control voltages that results in some desired behaviour of the chip, using the estimated model from the first stage.

This section starts with a detailed description of the architecture used to model the chip. Next, the training and testing procedures are presented. After that, the detailed description of the control voltages predictor for the chip is presented. Finally, the section ends with extending the proposed structure to account for a fully-quantum setting where phases can be measured at the output.

4.3.1 Chip Model Architecture

The deep learning architecture is shown in Figure 4.3. The first layer in the model is a Gated Recurrent Unit (GRU) [9]. This is a variant of the Long-Short Term Memory (LSTM) structure often used in sequence prediction and classification [10]. GRU is more efficient than LSTM as it has fewer parameters to be learned during the training stage. However, in terms of accuracy, it is not very clear which is better generally, and this remains an open topic under investigation within the machine learning community [79]. The number of inputs is equal to the number of electrodes which is $2n$. For our implementation, the number of hidden units of the GRU is chosen to be 60. In general, more hidden units allow modelling more complex waveforms, but on the expense of more parameters to learn and thus more computational resources required. The objective of this layer is to learn the interaction Hamiltonian, i.e. learn how the Hamiltonian depends on the external voltages. This should also include the parasitic effects in the chip causing distortions of the applied voltage waveforms. The number of free parameters of any real-valued symmetric Hamiltonian of size $n \times n$ is $\frac{n}{2}(n+1)$. However, the output of the GRU is the output of each hidden node. So, to extract the required number of outputs, we add a neural network (NN) formed of a single layer that is fully-connected to all of the outputs of the GRU. The number of neurons is exactly equal to $\frac{n}{2}(n+1)$, as each neuron generates one output. Linear activation is used for all neurons, to allow the output to take any value and not be restricted in some range if we use

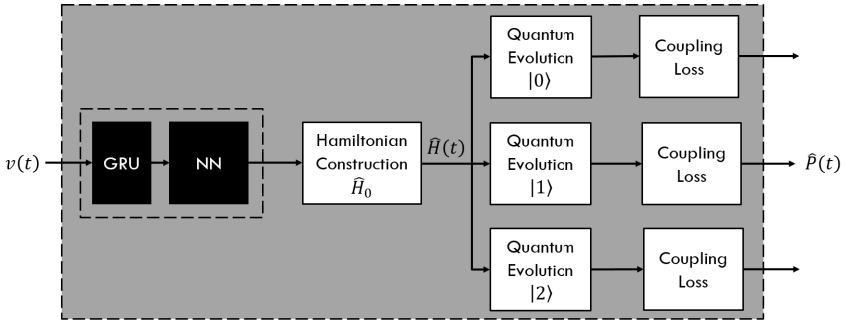


Figure 4.3: The different layers for the proposed graybox architecture to model a 3-waveguide chip ($n = 3$), with inputs representing the control voltages and outputs representing the measured optical power. The first two layers together represent the blackbox part of the model, while all other layers represent the whitebox part. The Gated Recurrent Unit "GRU" layer together with the Neural Network "NN" layer model the map between the interaction Hamiltonian and the control voltages, as well as the distortions of the control voltages. The "Hamiltonian construction" layer converts the output from the previous layer into a Hermitian matrix and adds the zero-voltage Hamiltonian. The "Quantum Evolution" layer calculates the probability amplitudes of the evolved quantum state given the Hamiltonian from the previous layer. The "Coupling Loss" layer models the power loss during measurement. The branching ensures that we get the same Hamiltonian for different initial states to be consistent with linearity of quantum mechanics. The graybox model is trained over two stages. The first stage is training the whitebox parts using zero-voltage measurements. The second stage is training the blackbox parts using random pulse measurements.

other activations such as sigmoid. Notice, that the GRU is a sequential layer, so the output has an extra dimension of time. However, the NN layer is static acting equivalently on each time slice of the output of the GRU. This means that weights applied to the GRU output at every time instant are the same. These two layers together act as a device to learn the free parameters of the Hamiltonian as a function of the input voltages.

The third layer in the structure is a custom-defined layer that has two functionalities. The first one is to reconstruct a symmetric matrix from the output of the previous layer. This is done by reshaping the outputs as an upper triangular matrix, and then sum it with its transpose. The second functionality is to add to the drifting Hamiltonian, that is the zero-voltage Hamiltonian that models the inherent coupling between the waveguides. The parameters of this drifting Hamiltonian are learned during the training process as will be illustrated later. The final output of this layer is therefore the full Hamiltonian of the system.

The next layer of the model is the quantum evolution layer. This is a custom defined layer, that takes some Hamiltonian as input, an initial quantum state as a defining parameter, and generates the probability amplitudes of the evolved state

as output. These probability amplitudes correspond to the waveguide power distribution. So, the layer first calculates the evolution matrix $U = e^{-iHl}$. Next, it calculates the evolved state $|\psi_F\rangle = U|\psi_0\rangle$. Finally, it calculates the probability amplitudes of the evolved state $|\langle m|\psi_F\rangle|^2, m = 0, 1, \ldots, n-1$.

Now, a problem arises if we train the model with the structure so far. Since, only one initial state is used in the quantum layer, then the learned Hamiltonian will be valid only for evolutions of this state. But, if we use the same Hamiltonian to evolve other initial states, we might not obtain a correct evolution. So, the algorithm will need to learn a different Hamiltonian for each initial state. This is a major problem, since quantum mechanics is a linear theory, so the Hamiltonian should not depend on the quantum state being evolved. Thus, we have to constrain the Hamiltonian in some sense so that it works for all states. The way we propose to solve this problem is to have different copies of the quantum layer each parametrized by a different initial state. Then, we connect the input of all these layers to the same output of the previous Hamiltonian layer. In this case, during the training, the model will be enforced to generate a Hamiltonian that correctly evolves each of the initial states. Since a unitary can be completely characterized by knowing the outputs corresponding to each of the computational bases as input states, we only need n of 'parallel' quantum structures each generating n outputs. So, the total number of outputs for this whole layer is n^2.

The final layer in the model is also a custom-defined layer that models losses during power measurements. This physically occurs due to coupling between between the chip and optical fibres connected to the photodetectors. The layer simply implements the calculation $\hat{P}_k = \frac{\epsilon_k P_k}{\sum_{i=1}^{n} \epsilon_i P_i}$, where $\hat{P}_k$ is the k^{th} measured power at waveguide k, and P_i is the actual power at the output of the chip for waveguide i. The denominator in the expression is to ensure that the measured powers are normalized, (i.e. form a distribution). The coupling coefficients are learned during the training stage as will be discussed later. For each quantum block in the quantum evolution layer, we cascade one of these coupling layers. However, all of these copies of the coupling layers are identical (i.e. have the same parameters). This reflects the fact that the losses are independent of which waveguide was used as input, and just related to the hardware of the experiment.

4.3.2 Training and Testing

There are two stages to do the training of the model, where all the unknown parameters of the model are leaned by providing examples. The first stage is to learn all the zero-voltage parameters, i.e. the drifting Hamiltonian and the coupling losses coefficients. All these parameters are static and do not depend on the input volt-

ages. For this training step, we detach the GRU and NN layers from the model. The input of the model is directly connected to the Hamiltonian construction layer, and is fixed to be all zeros. The output is the lossy power distribution. This is obtained experimentally by fixing the physical voltage on the chip to zero, using one of the waveguides as input and measure the power across each waveguide. The procedure is repeated for all input waveguides. Since, the distribution in this case is static, we get a total of n^2 readings. With this pair of training data (zero voltage as input, and n^2 readings as output), the model is trained by backpropagation using RMSprop [80], and all the unknown parameters are learned. We use the mean square error (MSE) as the loss function and also as the performance metric. This is because the problem is predicting a waveform, and MSE is one of the most commonly-used metrics for quantifying similarity between two waveforms. The lack of phase information at the output prevents us from constructing a full quantum state and thus evaluating quantum measures such as fidelity is not possible.

The second stage of training is to obtain the dynamic behaviour of the chip, (i.e. how the waveguide power distribution changes in time being a function of the input time-varying voltage. In this stage, the full model is used, and the input is connected to the GRU layer. All parameters learned from the first stage are fixed and do not change during this stage. Backpropagation is used to train the remaining unknown parameters using the pair of some voltage waveforms as input, and the corresponding measured power distribution waveforms as output, with MSE acting as loss function. After this stage, all the learned parameters are fixed and the model can be used in the testing phase.

In the testing phase, the model is given a new input that was not in the training set, and the predicted output is compared with the actual output. A good model is a one that generalizes well over new inputs. The end goal of using this architecture is a graybox model of the chip, capable of predicting the output distribution for any control voltage. However, practically this is a hard requirement due to the behaviour of machine learning algorithms. Usually, these structures have the ability to generalize for inputs that share some similarity with the training examples. In our case, the voltage waveform shape should be the same for the training and testing datasets (i.e. fix the pulse shapes to be square, Gaussian, raised cosine, etc.). After fixing the shape, the waveform parameters (such as amplitude, phase shift, etc.) for each example can be arbitrary. If we want the model to predict the output for other waveform shapes, then the training set has to include the other shapes as well. In this chapter, we restrict all the voltage waveforms to have the form of arbitrary synchronized square pulses. This means that for each example, the pulses across all electrodes start at the same time instant, have the same width, but can have different amplitudes. These parameters will differ though across different examples

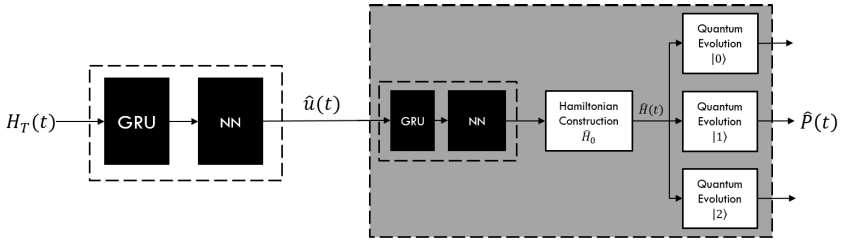


Figure 4.4: The proposed architecture of the controller for the $n = 3$ chip, with inputs representing the target sequence of Hamiltonians and outputs representing the target controlled optical power distribution. The trained graybox model of the chip is fixed preventing any further changes. An extra blackbox is added at the beginning. The Gated Recurrent Unit "GRU" and the Neural Network "NN" layers model the map between the sequence of target Hamiltonians and control voltages. The activation function in the NN layer is chosen to ensure that the resulting control voltages are limited to the allowed range the chip supports. After training, the control voltages can be estimated by probing the output of the "NN" layer.

in the datasets.

The architecture of this model has a major advantage which is the possibility of monitoring the output of each layer during testing, each corresponds to a physically significant quantity. So, the output of the NN layer is a prediction of the interaction Hamiltonian as a function of the input voltages and time. The output of the "Hamiltonian Construction" layer is a prediction of the total Hamiltonian matrix. The "Quantum Evolution" layers predict the ideal power distribution for each initial state, while the output of the last layer is prediction of the measured power distribution. This shows that relevance of this deep learning structure. For instance, had we used one LSTM-based blackbox instead of the proposed graybox, we would have been able to predict the measured power distribution only, but not the other quantities.

4.3.3 Controller Architecture

The second major task required after characterizing the chip is finding the control voltages needed to obtain a desired power distribution, resulting from the evolution of a target Hamiltonian. The architecture of prosed controller is shown in Figure 4.4. The first layer is a GRU layer followed by a fully-connected neuron layer similar to that used in the model architecture. However, the input is some desired target Hamiltonian, and output shall represent the control voltages which is a $2n$ vector. Since we need at least one of the electrodes to be connected to ground, we actual enforce the very first electrode to zero. Also, we enforce the last electrode arbitrarily to zero. This leaves out $2n - 2$ control voltages to predict. For efficiency purposes, we actually input only the upper triangular part of the Hamiltonian flattened into

an $\frac{n}{2}(n+1)$ vector.

One major issue to consider is that the voltage across any two adjacent electrodes should not exceed in absolute value $V_{\max}$. So, all the neurons at the output have a scaled hyperbolic tangent sigmoid activation in the form $f(x) = \frac{1}{2}V_{\max}\tanh(x)$. This ensures the output at each electrode is in $[-\frac{1}{2}V_{\max}, \frac{1}{2}V_{\max}]$, and thus the potential difference across any two adjacent electrodes is limited to $[-V_{\max}, V_{\max}]$.

Next, we cascade a copy of the previously trained model without the coupling loss layers. The reason behind dropping that layer is that the power loss is due to the measurement process, and not the operation of the chip. For instance, if two chips were connected in cascade with perfect coupling, then we would be interested to predict the control voltages for the first chip to produce some desired state at its output, and there will be no effects of the losses for the first chip. All the trained parameters of the model are fixed and do not change during the training of the controller. Connecting the pre-trained model enforces the whole controller structure to generate the ideal target power distribution. Thus, all the distortions that appear in the power distribution are dealt with automatically by the controller. The algorithm is enforced to produce voltage waveforms that undo the distortion effects in order to minimize the MSE. This means the algorithm is effectively learning an inverse model of the equivalent circuit of the chip, and simultaneously ensuring the final quantum state is correct. In some sense, this structure does both classical control (undoing the distortions) and quantum control (obtaining the target quantum state). By probing the output of the NN layer, the desired control voltage can be estimated.

It is worth mentioning that there is no requirement on the controller to generalize to every possible target Hamiltonian/target-distribution pair. Whenever we are interested to realize some sequence of operation on the chip, we redo the training of the controller, and probe the output of the NN layer. So, in some sense we are using backpropagation as a direct optimization procedure rather than a learning procedure. Additionally, the controller input is a sequence representing the Hamiltonian at each time step. This means we can obtain control voltages that allows changing the behaviour of the chip dynamically whilst operating.

The last point to note is that not every possible Hamiltonian can be realized with the chip model. Some Hamiltonians may require voltages that exceed the maximum allowed range. An open question is what kind of quantum gates can be actually implemented using this chip given the constraints. This is however outside the scope of this chapter.

4.3.4 Fully-quantum model

The architectures described so far are not fully quantum in the sense that the Hamiltonian is assumed to be real, and that we can only measure powers at the output (corresponding to probability amplitudes). However, it is possible to extend the proposed method to the fully quantum case, if we perform the Mach-Zehnder type of measurements as discussed previously. The overall architecture is quite similar, with the following modifications:

- The neural layer after the GRU is set to produce n^2 outputs instead of the $n(n+1)/2$, to account for the imaginary part of the Hamiltonian matrix elements.

- The Hamiltonian layer reshapes the output of the neural layer to an $n \times n$ matrix, where the lower triangular part represents the imaginary part of the Hamiltonian while the upper triangular part represents the real part. So, by multiplying the lower triangular part by i and adding the whole matrix to its Hermitian conjugate, we end up with an $n \times n$ Hermitian matrix. Also, the zero-voltage Hamiltonian H_0 is manipulated similarly to account for the possibility of complex-valued entries.

- The quantum layer outputs the Mach-Zehnder interferometer power measurements instead of the probability amplitudes. So if the final state is $\sum_k \alpha_k |k\rangle$, then the layer's outputs are $P_k(0) = \frac{1}{4}|\alpha_k + 1|^2$, and $P_k\left(\frac{\pi}{2}\right) = \frac{1}{4}|\alpha_k + i|^2$, for all $k = 1, ...n$. So, the total number of outputs for this layer is $2n$, and for the whole model is $2n^2$. We do not need to explicitly calculate the amplitude and phases from the interferometer measurements for the training. We will just use the interferometer measurements directly. The training follows the same procedure as discussed previously.

- For simplicity, we removed the coupling layer as the focus in this application is on exploring the possibility of learning a fully quantum system. However, in general we can include it.

- We still use MSE as a loss function and performance metric because the output is still a waveform (although representing interference measurements now). However, since there is complete information to reconstruct the state and the evolution matrix, we can use other metrics for performance evaluation such as fidelity.

- The controller architecture is the same, the only difference is the input of the first layer is the real and imaginary parts of the target unitaries, rather than the Hamiltonians. This seemed to perform better than having the Hamiltonians as input. This might be due to the fact that there exist infinitely

many Hamiltonians (all related with a factor of integer multiple of 2π in the eigenvalues) giving rise to the same unitary. And thus, the GRU might have trouble finding some of these equivalent Hamiltonians. However, if the input is directly the unitary then there is no redundancy. For the classical application, this did not seem to cause any problems because there was more freedom as the optimization is over the power distribution only. In the quantum application, it is more restrictive since the optimization is over the phase information as well.

4.4 Simulation Results

This section discusses the implementation details of our method and the results of the numerical simulations. A discussion on the significance of the results is given afterwards.

4.4.1 Implementation

For implementing the proposed architecture we used the "Tensorflow" Python package [7], and its high-level API package "Keras" [8]. The Python implementation of our algorithm is publicly available[1].

In order to do training and testing, we created a dataset consisting of control voltages in the form of random pulses, and the corresponding waveguide output power distribution for different input waveguides. We generated a total of 4000 examples, 3500 of which were used for training and 500 for testing. The amplitudes of the pulses are from -5 to +5 volts and the time domain is limited to the interval $0 \leq t \leq 200(ms)$ with sampling time of $0.2(ms)$. In each example, the voltage on the first and last electrodes are fixed at zero, while the pulses are applied on the remaining electrodes. The restriction on these pulses is that they have to be synchronized across the different electrodes, starting and ending at the same time. However, the durations and amplitudes are chosen randomly from one example to another. The experimental setting would be generating these pulses, applying them physically to the chip, measuring the output power distribution, and finally training the model. However, in this chapter, we restrict the study to computer simulations. So, we created a simulator for the chip that generates the waveguide power distribution given a set of control voltages, using the Hamiltonian model

[1] https://github.com/akramyoussry/GRUBI

described by the tridiagonal real-valued matrix

$$H = \begin{pmatrix} \beta_1 & C_{1,2} & 0 & 0 & \cdots & 0 \\ C_{1,2} & \beta_2 & C_{2,3} & 0 & \cdots & 0 \\ 0 & C_{2,3} & \beta_3 & C_{3,4} & \cdots & 0 \\ 0 & 0 & C_{3,4} & \beta_4 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & C_{n-1,n} \\ 0 & 0 & 0 & 0 & C_{n-1,n} & \beta_n \end{pmatrix}, \tag{4.7}$$

where β_i is the propagation constant along the i^{th} waveguide, and $C_{i,j}$ is the coupling coefficient between waveguides i and j. The propagation constant is given by

$$\beta_i = \frac{2\pi}{\lambda}\left(n_0 + \Delta n \Delta V_i\right), \tag{4.8}$$

where λ is the wavelength, n_0 is the intrinsic refractive index of the waveguide, Δn is a dynamical proportionality constant that determines how much the the propagation constant changes by changing the voltage across the waveguide ΔV_i. The coupling coefficient is given by

$$C_{i,j} = C_0 + \Delta C_1 \Delta V_{i,j} + \Delta C_2 \left(\Delta V_i + \Delta V_j\right), \tag{4.9}$$

where C_0 is the intrinsic coupling between two adjacent waveguides, $\Delta V_{i,j}$ is the potential difference across the substrate between the two waveguides i and j, ΔV_i and ΔV_j are the voltages across waveguides i and j, and ΔC_1 and ΔC_2 are dynamical proportionality constants that determine the amount of change of the coupling between two waveguides by changing the voltages across them. These relations assume that Hamiltonian depends on the voltages linearly, and that the coupling is always between neighbouring waveguides. The simulator takes into account the non-ideal effects due to the equivalent circuit behaviour of the chip, by simulating distortions on voltage pulses. It also simulates coupling losses. For the results presented in this chapter, the simulation parameters were as follows. $n = 3$, $\lambda = 808 \times 10^{-9}$, $l = 3.6 \times 10^{-2}$, $n_0 = 2.1753$, $\Delta n = 5 \times 10^{-6}$, $C_0 = 100$, $\Delta C_1 = 1.5$, $\Delta C_2 = -1.3$, and $\epsilon_k = \{0.9, 0.8, 0.5\}$.

4.4.2 Results

For the task of modelling the chip, the MSE obtained after 10^4 iterations was about 2.1×10^{-4} for the training dataset. Figure 4.5a shows the MSE versus the number of iterations. For the testing dataset, the MSE evaluated is 3.4×10^{-4}. Supplementary Figures 4.8,4.9, and 4.10 show examples selected randomly of the testing dataset

Table 4.1: Target Hamiltonians used for testing the proposed architecture for the controller as per Equation 4.10 for the classical model, and Equation 4.12 for the fully-quantum model.

Symbol	Expression	Description
I	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Identity (100% decoupling between waveguides)
X_{13}	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	Perfect Transfer between waveguide 1 and waveguide 3
H_{13}	$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \end{pmatrix}$	50-50 Power split between waveguide 1 and waveguide 3 (Hadamard gate)
X_{12}	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Perfect transfer between waveguide 1 and waveguide 2
Z_{13}	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	Phase shift of π between waveguide 1 and waveguide 3
$R_{X_{13}}(\theta)$	$\exp(-i\theta X_{13})$	Rotation about X-axis by angle θ between waveguide 1 and waveguide 3
$R_{Z_{13}}(\theta)$	$\exp(-i\theta Z_{13})$	Rotation about Z-axis by angle θ between waveguide 1 and waveguide 3

including the control voltages, simulated measured waveguide power distribution and the predicted power distribution.

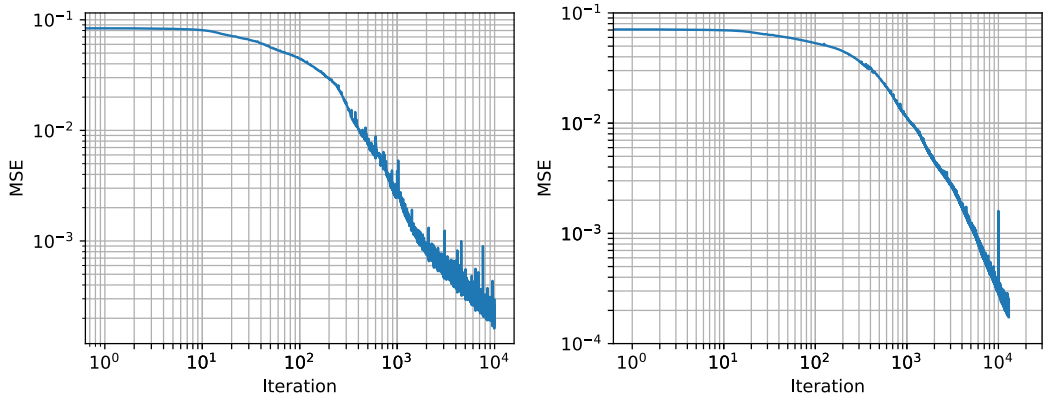
To test the control part, we defined as an example a sequence of target unitaries in the time interval $0 \leq t \leq 300(ms)$, given by

$$U(t) = \begin{cases} X_{13} & 50 \leq t < 80 \\ H_{13} & (110 \leq t < 140) \vee (250 \leq t < 280) \\ X_{12} & 170 \leq t < 210 \\ I & \text{otherwise} \end{cases} \tag{4.10}$$

where the unitaries are defined in Table 4.1. The Hamiltonian is then evaluated for each time interval by taking the matrix logarithm $H = \frac{i}{l} \log U$.

After training the controller model for 500 iterations, the MSE was 2×10^{-2}. The MSE versus the number of iterations is plotted in Figure 4.6a. The resulting control voltages are shown in Supplementary Figure 4.11, and the resulting predicted ideal power distribution in Supplementary Figure 4.12.

For the second application which is the fully-quantum setting, we use the same dataset of pulses, but now we have the interferometer power measurements as the



(a) Classical (b) Quantum

Figure 4.5: The proposed machine learning structure is trained over a set consisting of 3500 examples. The set consists of random control pulses and the corresponding simulated output optical power for the classical application, and interferometer power measurements for the fully-quantum application. The plot shows the MSE, which is used as the loss function for the training, versus the number of iterations for the (a) classical application and (b) the quantum application.

model output. The number of iterations is 1.3×10^4, which is more than the other model to account for doubling the size of the outputs. Figure 4.5b shows the performance of the training in this case. The MSE evaluated for the testing dataset it 2.88×10^{-4}, while it was 1.74×10^{-4} for the training set. This is an indication for the the ability of the model to fit the training dataset as well as generalize to the testing dataset. Supplementary Figures 4.13 and 4.14 show the result of the predicted waveforms using the same control pulses as in Supplementary Figures 4.8 and 4.9. Now, since the phase is also measured, then we can have a complete quantum description of the output state, and thus we can construct the evolution unitary. A commonly used measure for the closeness of two quantum gates U and V of dimension d, is the gate infidelity defined as

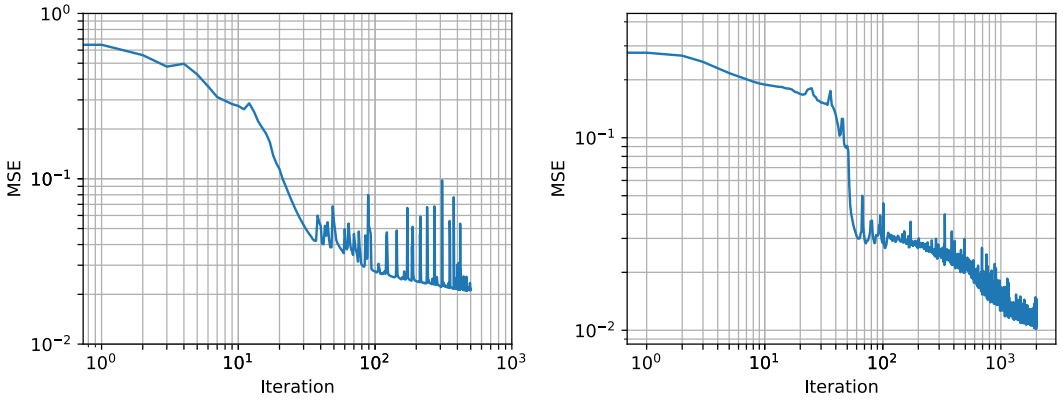
$$1 - F(U, V) = 1 - \frac{|\operatorname{tr}\left(U^\dagger V\right)|^2}{d^2}. \tag{4.11}$$

Infidelity is thus a number between 0 and 1, with 0 representing complete overlap (i.e. same matrices). Supplementary Figure 4.15 shows the infidelity between the predicted unitary and actual unitary as a function of time for these two examples. Finally, for evaluating the control algorithm in this setting, we used as an example

the following sequence for $0 < t < 280(ms)$

$$U(t) = \begin{cases} X_{13} & 50 \leq t < 90 \\ R_{X_{13}}\left(\frac{\pi}{4}\right) & 130 \leq t < 170 \\ R_{Z_{13}}\left(0.1\right) & 210 \leq t < 250 \\ I & \text{otherwise} \end{cases} \tag{4.12}$$

The history of the MSE of the controller during the training is shown in Figure 4.6b. The resulting infidelity between the desired sequence of quantum gates and the controlled quantum gates are shown in Figure 4.7, while the control voltages are shown in Supplementary Figure 4.16.



(a) Classical (b) Quantum

Figure 4.6: The proposed controller architecture is trained to obtain the control voltages needed to achieve the target sequence of (a) optical output power distribution for the classical application defined in Equation 4.10, and (b) unitary gates for the quantum model defined in Equation 4.12. The plot shows the MSE used as the loss function versus the number of iterations.

4.4.3 Discussion

The presented results show the accuracy of the proposed architecture in modeling the chip with all the constraints mentioned earlier. Quantitatively, the loss represented by the MSE decreases on average by increasing the number of iterations during the training phase, reaching a small value that is in order of 10^{-4}. However, this is not sufficient to completely asses the behaviour of the proposed algorithm. The plots of the waveforms in Supplementary Figures 4.8,4.9, and 4.10 show qualitatively the accuracy of the model. The difference between the predicted and simulated power distribution is almost negligible. More importantly, since the model has not been trained on the testing set, it proves that the proposed structure can generalize to

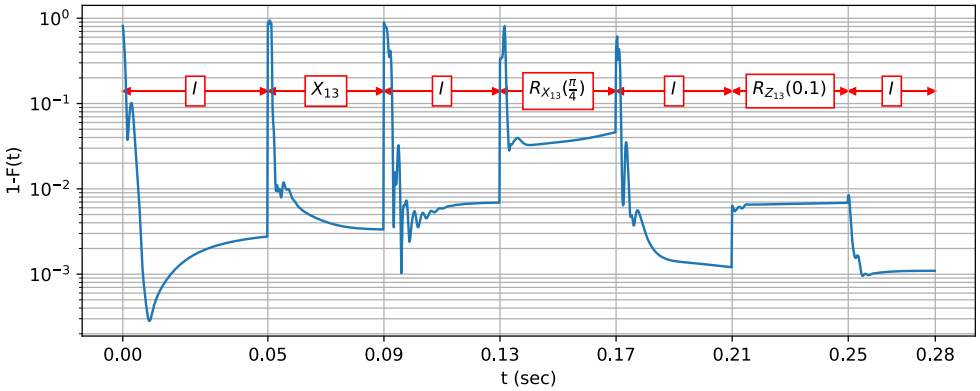


Figure 4.7: The resulting infidelity for the sequence of target quantum gates defined in Equation 4.12, after training the proposed controller architecture. Photons arriving at different time instants would see different quantum gates. The plot shows the infidelity of between the ideal quantum gate and the actual gate obtained by the controller at each time step.

examples that are drawn from the same distribution as the training set. Absolute generalization to every possible input is not possible when using ML technqiues. However, this might not be an issue in a hardware experiment, since the nature of the inputs (control pulses) are usually fixed. The architecture does not allow to give explicit mathematical expression for the Hamiltonian. But nonethelss, we can use it directly to estimate the Hamiltonian given the control voltages. Also, quantitatively the MSE evaluated for the testing set is also in the order of 10^{-4}, without much degradation than the value for the training set.

The qualitative results also show that the architecture is able to handle all the challenges described in Section 4.2.2. We were able to model the distortions caused by the equivalent circuit without the need to explicitly define a particular circuit model or how the Hamiltonian depends on the circuit response. This also saves us from having to characterize these parasitic effects experimentally, which is difficult as discussed previously.

For the control task, the proposed method was also very successful in obtaining the required control voltages as reflected in Supplementary Figure 4.12. We see that the distortions that were present in the power distributions are not there anymore, and at the same time we were able to achieve the required functionality. The control voltages were also limited to the desired operating range. However, we see that for the X_{12} gate, the algorithm could not do full transfer between waveguides 1 and 2. We believe that this is related to the fact that not all gates are possible to implement, which is a subject of the future work. A final thing to notice is that all the examples in the training set were limited to the time range $0 \leq t \leq 200(ms)$. However, the target control sequence has a wider range $0 \leq t \leq 300(ms)$, and still we are successful in our task. This is a result of using the GRU layers, and shows

how the whole model generalizes quite well.

The proposed modifications in the architecture to account for fully-quantum models was also very successful. This is evident from the low MSE value for both training and testing datasets with small difference between both. This is supported qualitatively through the plots of the power waveforms and infidelity versus time. Also, the controller architecture seems to perform quite well. The example we tested shows the possibility of implementing some basic quantum gates which are identity, Pauli X, rotation about X-axis with angle $\pi/4$ which is equivalent to a Hadamard gate with phase shifts, and rotation about Z-axis. At each time instant, the photon travelling through the chip will sense a different quantum gate. The gate infidelities at all time instants, apart from the transition moments, are low (worst case was 4×10^{-2}). The gates act on a qubit spatially encoded between the first and last waveguide. However, there is a major advantage for our proposed controller architecture, which is the input is the target sequence of quantum gates rather than a single gate. In general, the control voltages required for realizing a particular gate can depend on the previous history of gates realized so far due to the drifting problem described earlier. In other words, the same gate could need different control pulses at different points in time during the operation of the chip. Our proposed method deals automatically with this issue compared to standard quantum control literature that deals with one target quantum gate only [32, 35, 81].

4.5 Conclusion

In this chapter, we proposed a deep learning structure that is suitable to model a reconfigurable integrated waveguide array chip. The architecture addresses three major problems faced when characterizing the chip experimentally. The uncertainty in the Hamiltonian model, the presence of undesired macroscopic dynamics causing distortions, and losses due to imperfect measurements. The proposed architecture followed a graybox model approach, where the Hamiltonian as a function of control voltages is treated as a blackbox utilizing a GRU network as a main component. The waveguide power distribution as function of the Hamiltonian is treated as a whitebox since the laws of quantum mechanics are known. We also proposed another complementary deep learning structure to obtain the control voltages required to achieve some target sequence of gates. The qualitative as well as quantitative results showed a very promising performance for both tasks.

Comparing the proposed method to standard techniques such as GRAPE is not fair. There are two approaches by which we can apply standard quantum control techniques. The first approach is to apply the algorithm in an online fashion, where control pulses are optimized by directly accessing the device. The main drawback of

this approach is how to experimentally measure the fidelity in an efficient way. The second approach is to apply GRAPE offline, where a model for the device is used to estimate the fidelity and optimize the control pulses. The challenge here is the existence of uncertainties in the model, and thus it is difficult to write down accurate equations to describe the system. This is actually the main motivation behind the proposed framework. This leaves us with a final possible option which is using GRAPE with the proposed ML model after training. In that case the performance of GRAPE will be dependent on the performance of the trained model, and thus a fair comparison with the full ML approach is not possible.

Finally, there are limitations of the presented ML-based approach that one has to consider. The first limitation is related to the scalability of the method. This method is suitable for small-scale systems, in which constructing a dataset from an experiment is a feasible task. For large-scale systems, the method could be applied for a partial characterization (For example a limited set of controls or target outputs). The growth of computational power makes it possible to train huge datasets. The bottleneck is actually constructing those datasets. The second limitation is related to the generalization capabilities of the model as discussed earlier. This problem while persistent in most ML applications, it might not be challenge for the application under consideration. This is because usually the experimental settings (For example the control pulse shapes and constraints) in which we characterize and operate the device is usually fixed. The final limitation is that this method assumes a closed-system evolution dynamics. In the next chapter, we extend this framework to open quantum systems in the absence of prior information about the environment.

4.6 Supplementary Figures

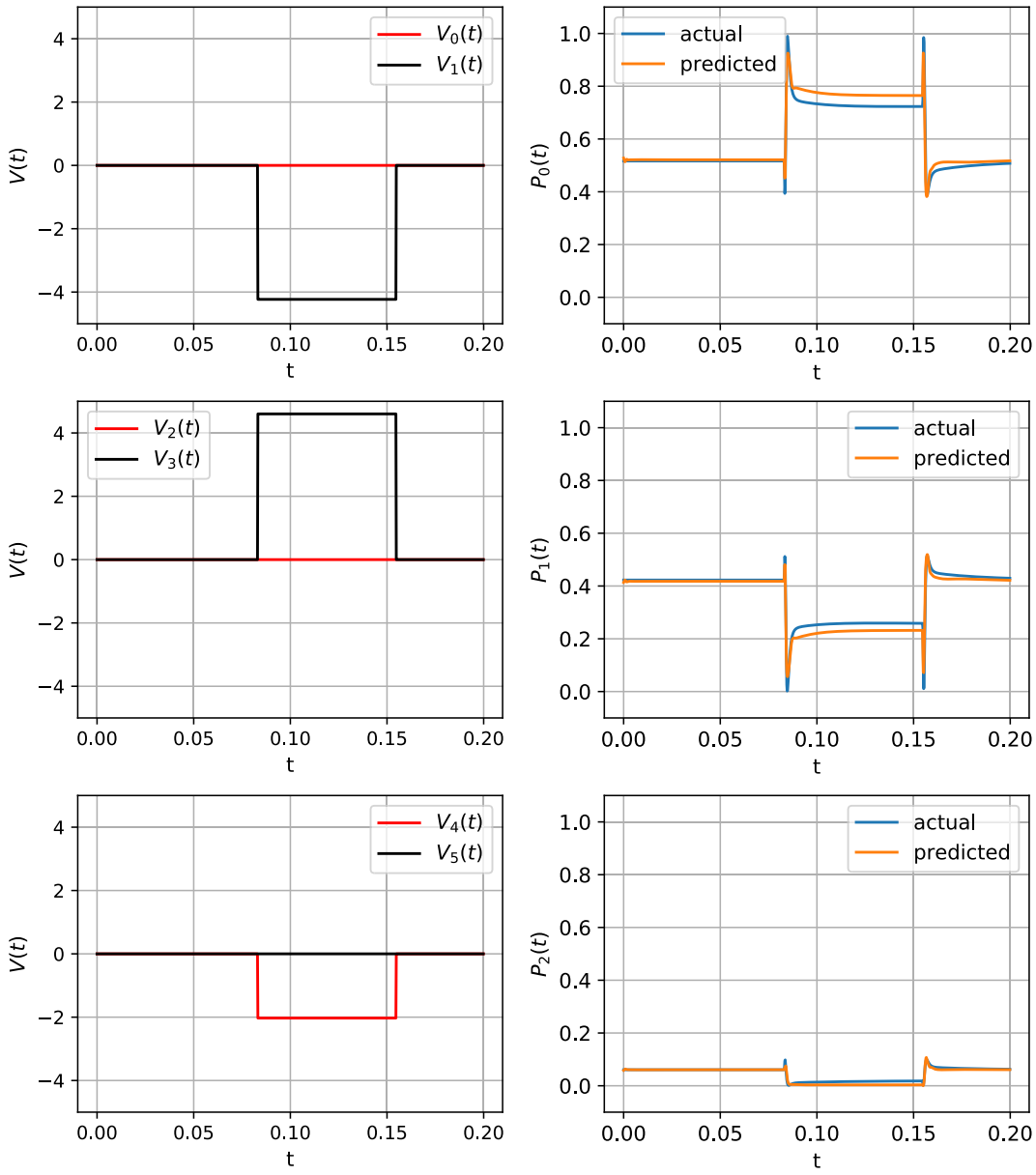


Figure 4.8: Random example from the testing dataset. The left column is the control voltages applied to the electrodes across the waveguides. The right column is the waveguide output power distribution, both simulated as in the dataset and predicted by the proposed algorithm. The initial state is $|0\rangle$, i.e. full power at the first waveguide.

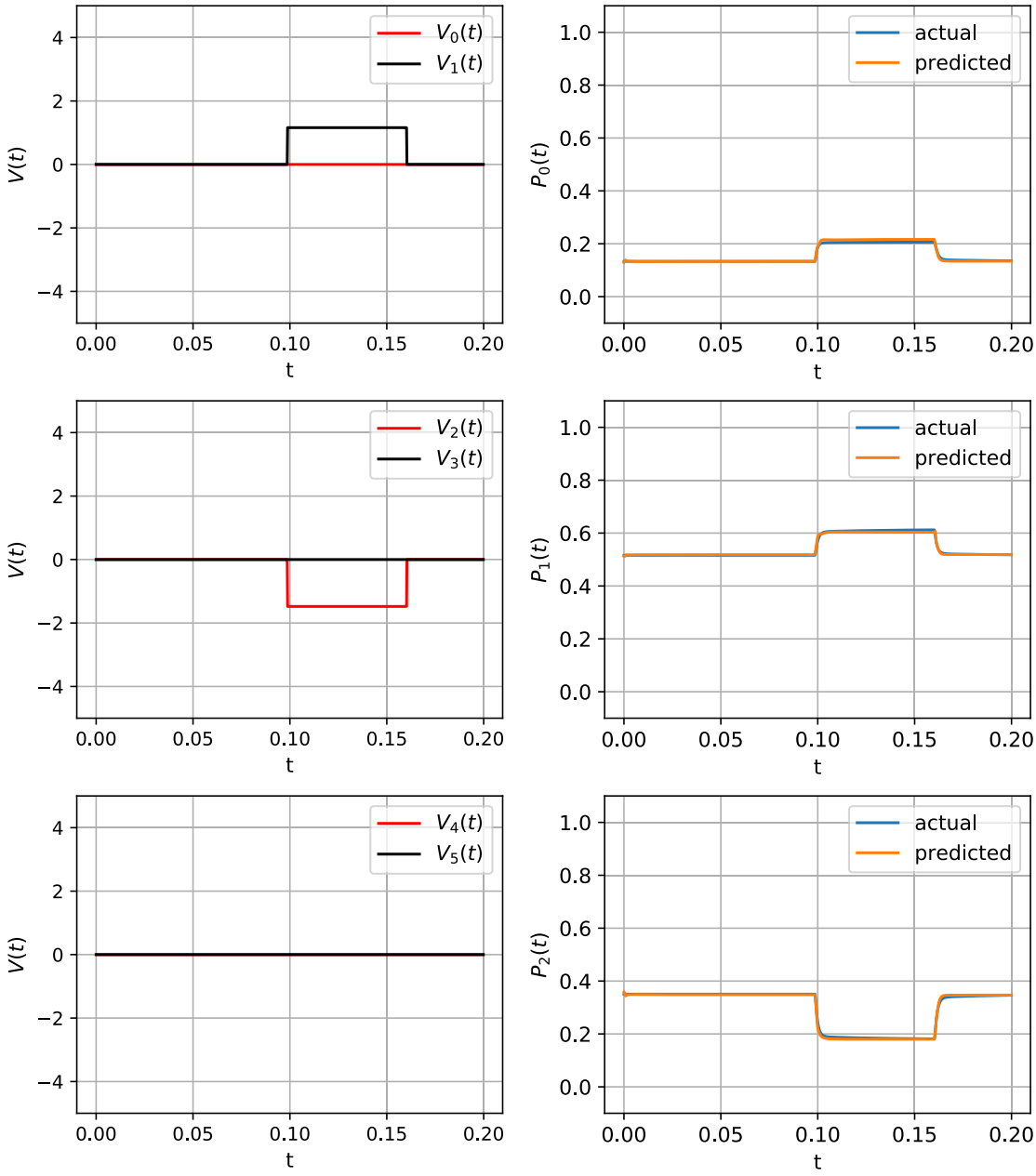


Figure 4.9: Random example from the testing dataset. The left column is the control voltages applied to the electrodes across the waveguides. The right column is the waveguide output power distribution, both simulated as in the dataset and predicted by the proposed algorithm. The initial state is $|2\rangle$, i.e. full power at the third waveguide.

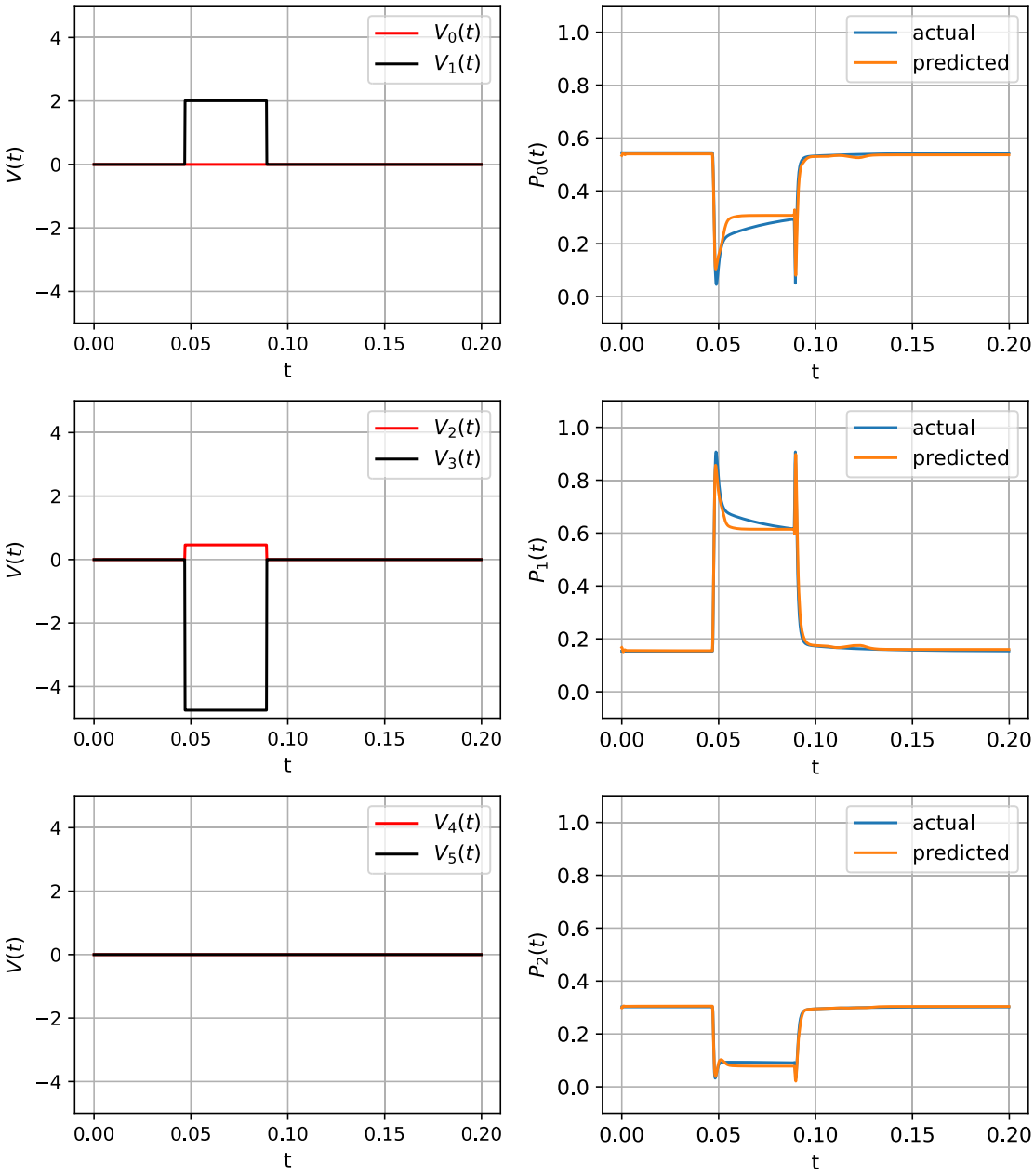


Figure 4.10: Random example from the testing dataset. The left column is the control voltages applied to the electrodes across the waveguides. The right column is the waveguide output power distribution, both simulated as in the dataset and predicted by the proposed algorithm. The initial state is $|1\rangle$, i.e. full power at the second waveguide.

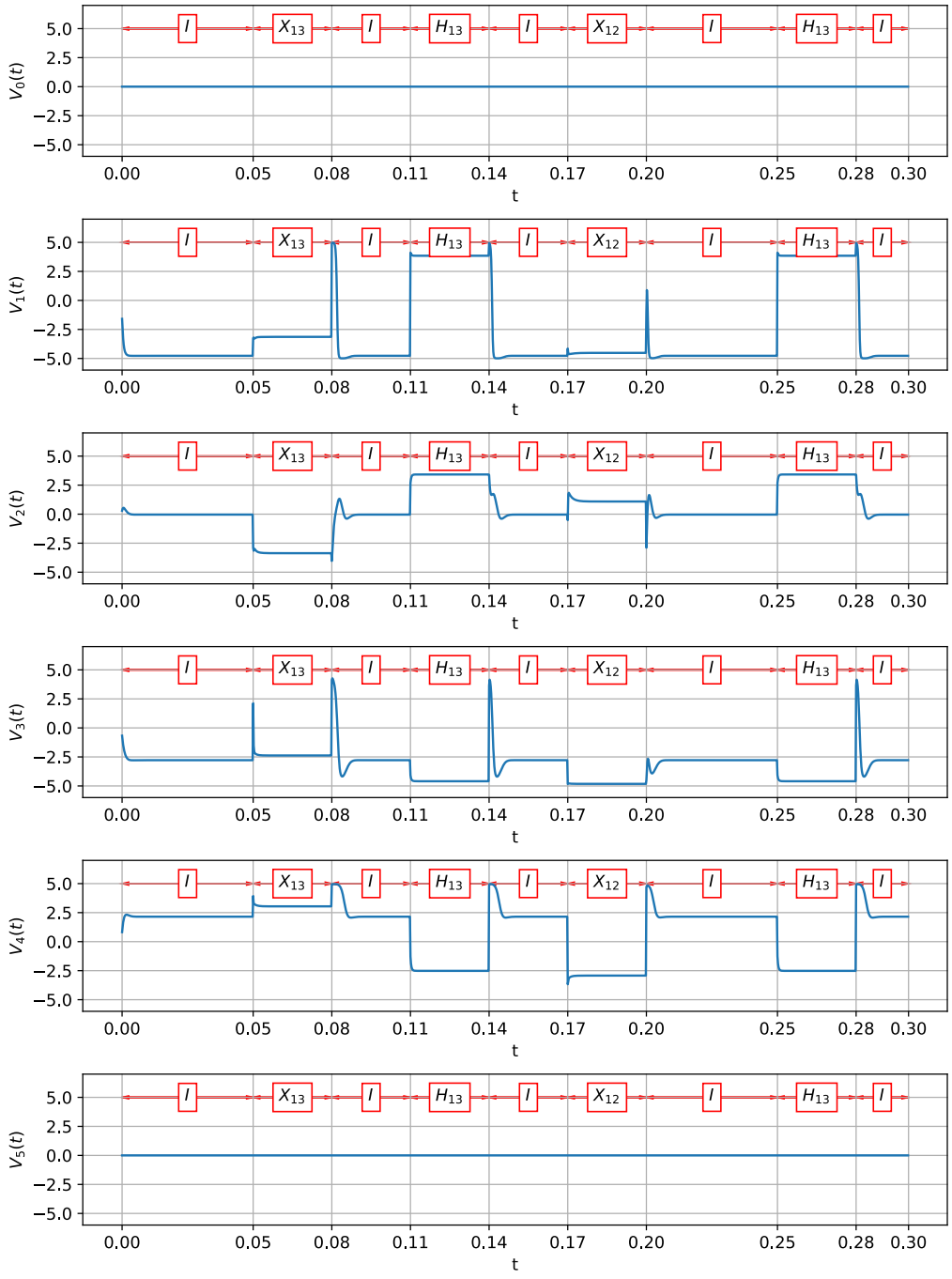
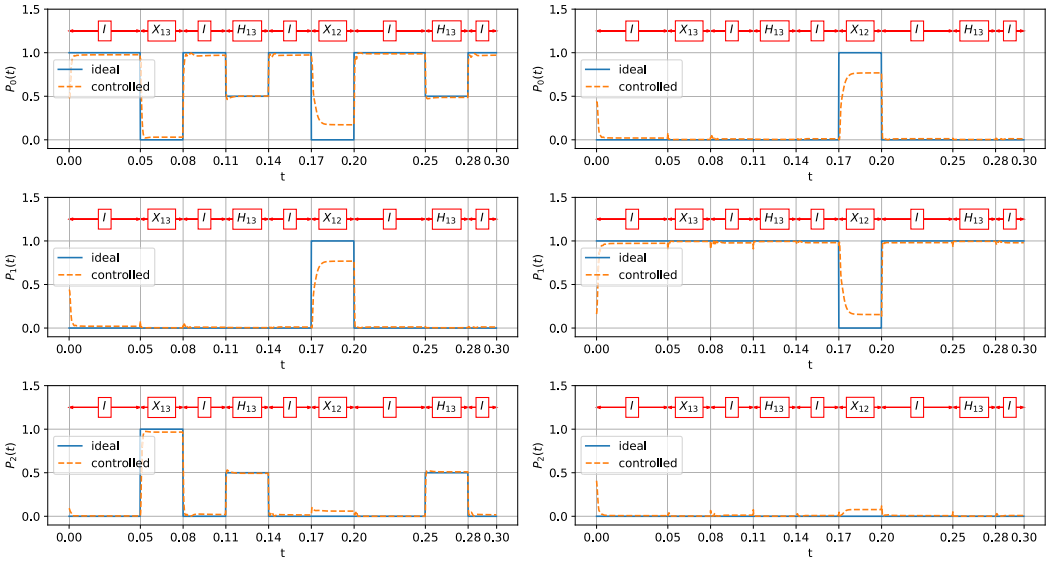
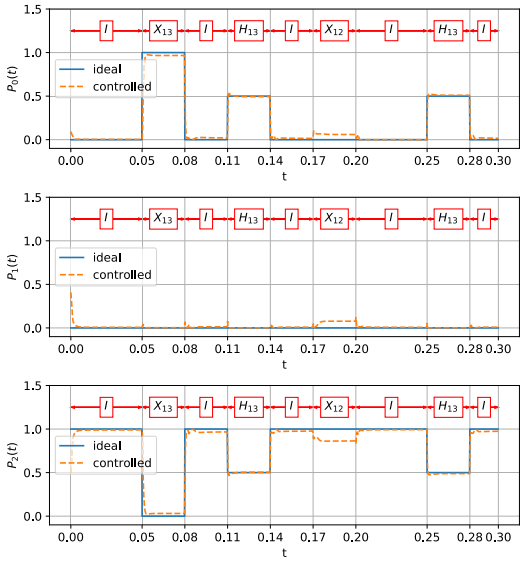


Figure 4.11: The resulting control voltages to obtain the sequence of targets defined in Equation 4.10.



(a) initial state = $|0\rangle$ (b) initial state = $|1\rangle$



(c) initial state = $|2\rangle$

Figure 4.12: The resulting waveguide distribution realizing the sequence of targets defined in Equation 4.10, for the initial states a) $|0\rangle$, b) $|1\rangle$, and c) $|2\rangle$.

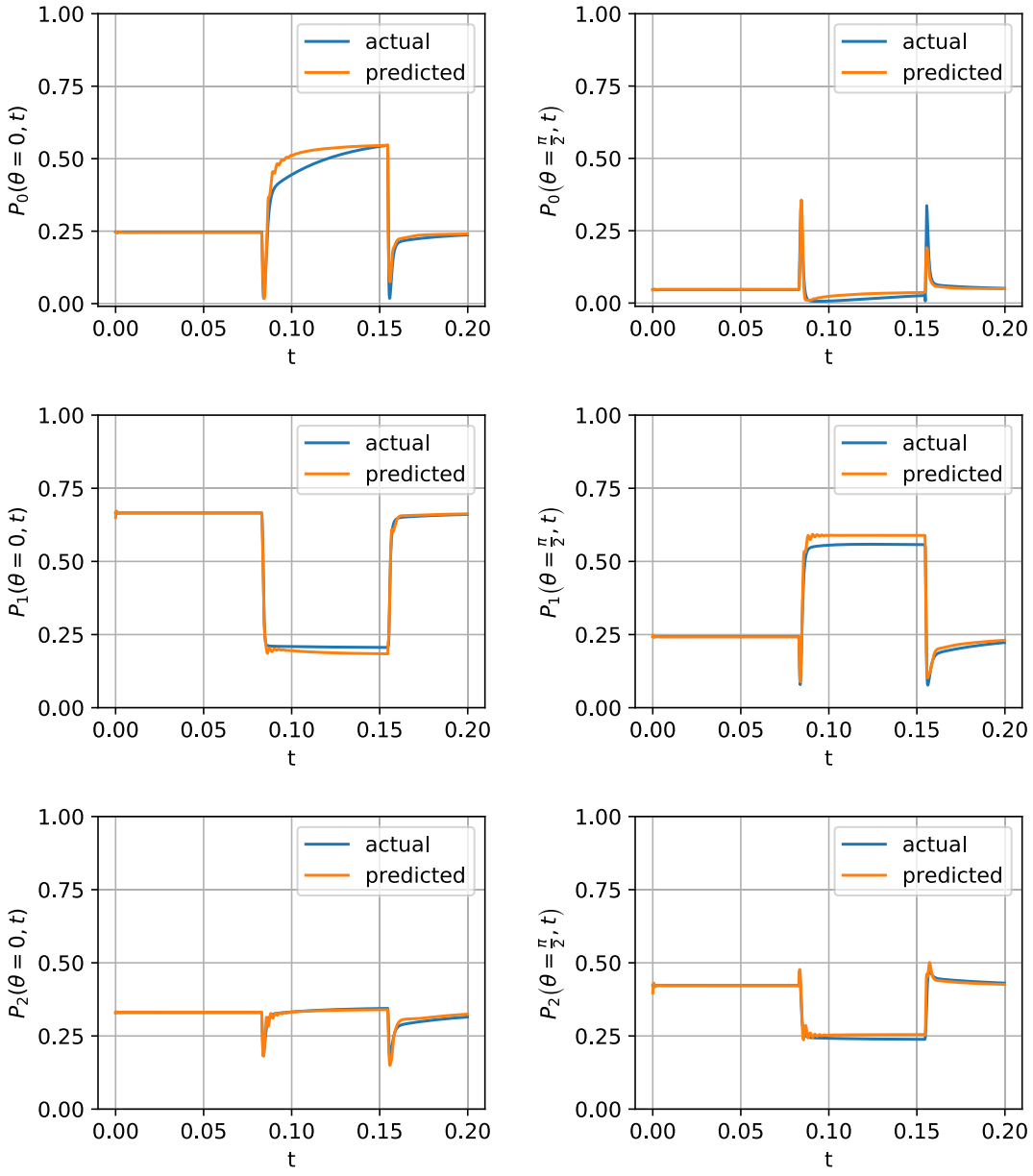


Figure 4.13: Random example from the testing dataset showing the interferometer power measurements across each waveguide, both simulated as in the dataset and predicted by the proposed algorithm. The same control voltages are applied as in Supplementary Figure 4.8. The initial state is $|0\rangle$.

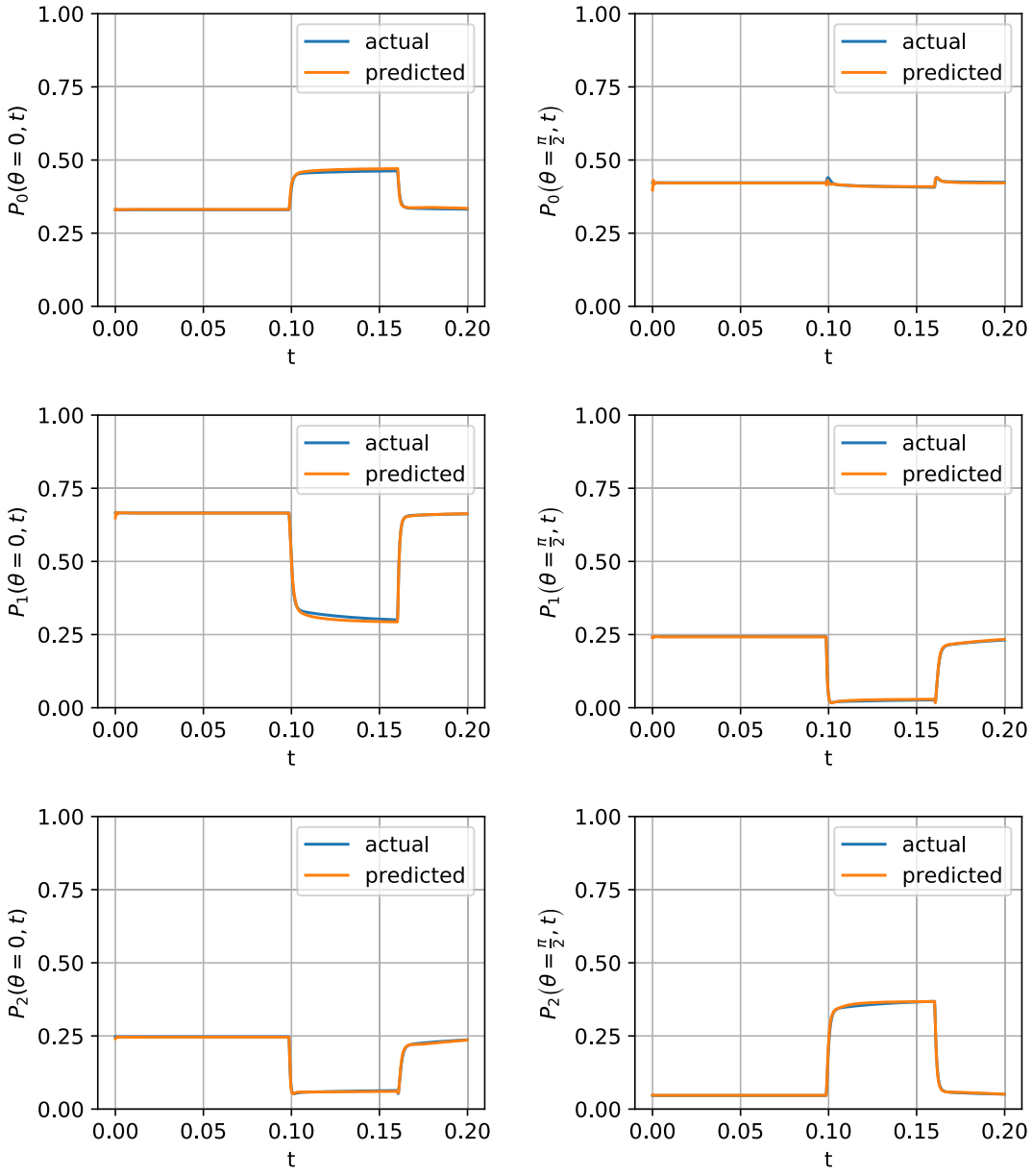
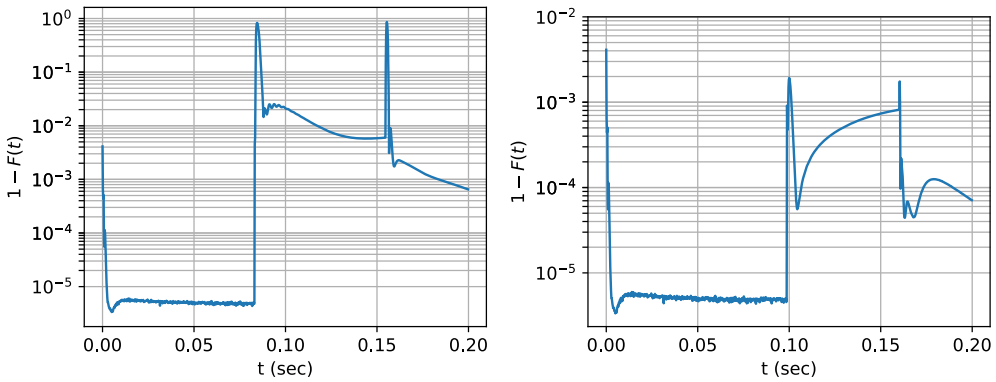


Figure 4.14: Random example from the testing dataset showing the interferometer power measurements across each waveguide, both simulated as in the dataset and predicted by the proposed algorithm. The same control voltages are applied as in Supplementary Figure 4.9. The initial state is $|2\rangle$.



(a) Infidelity for the example in Supplementary Figure 4.13

(b) Infidelity for the example in Supplementary Figure 4.14

Figure 4.15: The gate infidelity evaluated between the predicted evolution unitary and the actual evolution unitary for two random examples from the testing set.

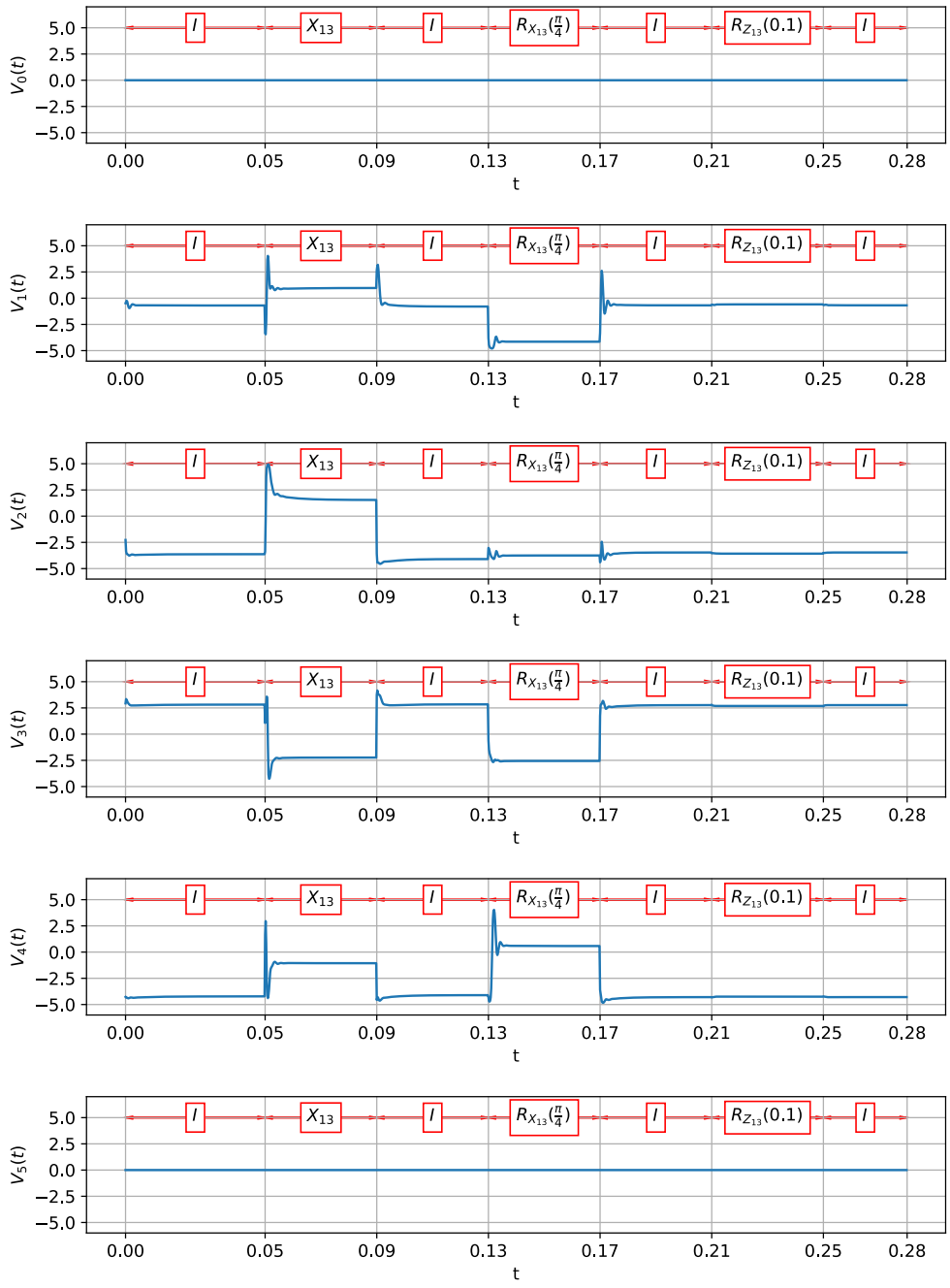


Figure 4.16: The resulting control voltages to obtain the sequence of targets defined in Equation 4.12.

Chapter 5

Modelling and Control of Open Quantum Systems

Abstract The ability to use quantum technology to achieve useful tasks, be they scientific or industry related, boils down to precise quantum control. In general, it is difficult to assess a proposed solution due to the difficulties in characterising the quantum system or device. These arise because of the impossibility to characterise certain components *in situ*, and are exacerbated by noise induced by the environment and active controls. Here we present a general purpose characterisation and control solution making use of a deep learning framework composed of quantum features. We provide the framework, sample datasets, trained models, and their performance metrics. In addition, we demonstrate how the trained model can be used to extract conventional indicators, such as noise power spectra.

5.1 Introduction

Accurately controlling the dynamics of open quantum systems is a central task in the successful implementation of quantum-enhanced technologies. Doing so to the highest possible level of accuracy involves a two-stage approach: first, quantum noise spectroscopy (QNS) [23, 38, 82–100] protocols are used to infer characteristics of the open quantum system that affect the open quantum system dynamics, and then optimal control routines (OC) exploit this information to minimize the effect of noise and produce high quality gates [101, 102].

In this work, we go beyond the aforementioned approach and pose the problem in a machine learning (ML) context. By doing so, we provide a common language in which the "learning" (equivalent to QNS) and "validation" (the precursor to OC) cycles are directly related to the objective of controlling an open quantum system. Notably, we show that doing so considerably extends the real-world applicability of

the aforementioned two-stage strategy, as one can forgo some of the non-trivial control and model assumptions necessary for the implementation of sufficiently general QNS protocols. The success of the approach relies on the fact that the ML algorithm learns about the dynamics *relative to a given set of control capabilities*, which effectively reduces the complexity of the problem in a way meaningful to experimental constraints (see also [103] for a formal analysis of this problem).

The guiding principles of this work were to develop a framework that enables us to have models that are independent on any assumptions. It has to be suitable for estimating physically relevant quantities. And finally, it should have the capacity to do standard tasks such as decoherence suppression and quantum control.

The proposed method is based on a graybox approach, where all the known relations from quantum mechanics are implemented as custom whitebox layers—quantum features—while the parts that depend on assumptions on noise and control are modelled by standard blackbox machine learning layers. In this chapter, we show how to construct such a model. The proposed model can be trained from experimental data consisting of sets of control pulses, and corresponding quantum observables. However, in this chapter, we do not have access to an actual experiment, and thus we train the algorithm and asses its performance using synthesized datasets, obtained by computer simulations of a noisy qubit system. The results show high accuracy in terms of prediction error. We also show the possibility of utilizing the trained model to do basic quantum control operations. This chapter opens the door for a number of possible novel machine-learning methods in the fields of quantum dynamics and control.

This work complements the existing literature applying classical machine learning to the quantum domain. Recently, machine learning and its deep learning framework [62] have been applied to many areas of quantum information, and physics more generally. Application areas include quantum control [68–70], characterization of quantum systems [49,67,71,104], experiment design [63–65], quantum cryptography [72], and quantum error correction [73,105,106]. A related approach is Bayesian learning which was applied for Hamiltonian learning [26, 27], quantum noise spectroscopy [107], and characterization of devices [74].

The structure of the remainder of the chapter is as follows. The chapter starts with overview on the formulation of the problem under consideration in Section 5.2. Section 5.3 describes in detail the proposed solution using a graybox ML approach. Next, Section 5.4 discusses the implementation of the proposed method followed by the presenting the numerical results and its significance. Section 5.5 concludes the chapter. Section 5.6 contains supplementary figures related to the results discussed in Section 5.4.

5.2 Problem Statement

In broad terms, our objective is to effectively "characterize" and accurately predict the dynamics of a two-level open quantum system, i.e., a qubit interacting with its environment, undergoing user-defined control picked from a fixed set, e.g., consistent with the control capabilities available to a given experimental platform. In what follows we make this statement precise.

For concreteness we start by choosing a model for our dynamics. We will consider a qubit evolving in the presence of both quantum and classical noise [90] via a time-dependent Hamiltonian of the form,

$$H(t) = \sum_{\alpha=x,y,z} \sigma_\alpha \otimes B_\alpha(t) + H_{\text{ctrl}}(t), \tag{5.1}$$

where $B_\alpha(t) = \tilde{B}_\alpha(t) + \beta_a(t) I_B$ is an operator capturing the effect of a quantum bath, via the operator $\tilde{B}_\alpha(t)$ typically resulting by working in the interaction picture with respect to the bath internal Hamiltonian, and classical noise, via the stochastic process $\beta_a(t)$. Control is implemented via the Hamiltonian

$$H_{\text{ctrl}}(t) = \Omega \frac{\sigma_z}{2} + \sum_{\alpha=\{x,y,z\}} f_\alpha(t) \frac{\sigma_\alpha}{2}, \tag{5.2}$$

where Ω denotes the energy gap of the qubit, $f_\alpha(t)$ implements the user-defined control pulses along the $\alpha-$direction, and σ_α is the α Pauli matrix. Notice that this generic model can accommodate classical noise process simply by making $B_\alpha(t) = \beta_\alpha(t) I_B$, with $\beta_\alpha(t)$ an stochastic process. In all equations and simulations in this chapter, we work with units where $\hbar = 1$. Since we are interested in predicting the dynamics of the qubit in a time interval $[0, T]$, we will be interested in the expectation value $\mathbb{E}\{O(T)\}_\rho$ of observables O at time T given an arbitrary initial state of the system ρ and a choice of $\{f_\alpha(t)\}$.

While these expectation values contain the necessary information, it will be convenient to further isolate the effect of the noise. To this end we proceed as follows. Our starting point is the usual expression,

$$\mathbb{E}\{O(T)\}_\rho = \langle \text{tr}[U(T)(\rho \otimes \rho_B)U(T)^\dagger O] \rangle_c, \tag{5.3}$$

where $U(T) = \mathcal{T} e^{-i \int_0^T ds H(s)}$ and $\langle \cdot \rangle_c$ denotes classical averaging over the noise realizations of the random process $\beta_\alpha(t)$, and ρ_B is the initial state of the bath. One can then move to a toggling-frame with respect to the control Hamiltonian $H_{\text{ctrl}}(t)$,

inducing a control unitary $U_{\text{ctrl}}(T)$ via,

$$U_{\text{ctrl}}(T) = \mathcal{T}_+ e^{-i\int_0^T H_{\text{ctrl}}(s)ds}, \tag{5.4}$$

which enables the decomposition,

$$U(T) = \tilde{U}_I(T)U_{\text{ctrl}}(T),$$

with,

$$\tilde{U}_I(T) = \mathcal{T}_+ e^{-i\int_0^T U_{\text{ctrl}}(T)H_I(s)U_{\text{ctrl}}^\dagger(T)ds},$$

the (modified) interaction picture evolution (see Section 2.2.3 for details), and $H_I(t)$ is defined as

$$H_I(t) = U_{\text{ctrl}}^\dagger(t)H_1(t)U_{\text{ctrl}}(t), \quad H_1(t) = \sum_{\alpha=x,y,z} \sigma_\alpha \otimes B_\alpha(t). \tag{5.5}$$

In turn, this allows us to rewrite,

$$\mathbb{E}\{O(T)\}_\rho = \text{tr}[V_O(T)U_{\text{ctrl}}(T)\rho U_{\text{ctrl}}(T)^\dagger O], \tag{5.6}$$

where the operator

$$V_O(T) = \langle O^{-1}\tilde{U}_I^\dagger(T)O\tilde{U}_I(T)\rangle\,, \tag{5.7}$$

where $\langle\cdot\rangle = \langle \text{tr}_B[\cdot\rho_B]\rangle_c$ conveniently encodes the influence of the noise. As such, this operator is central to understanding the dynamics of the open quantum system. If our objective is, as is common in optimal control protocols and imperative when quantum-technology applications are considered, to minimize the effect of the noise, e.g., via a dynamical decoupling [108–110] or composite pulses [111, 112], then one needs to determine a set of controls for which $V_O \to I$. Notice that $\text{tr}[O\,\langle\tilde{U}_I^\dagger O\tilde{U}_I\rangle]$ can be interpreted as the "overlap' between the observable O and its time evolved version $\langle\tilde{U}_I^\dagger O\tilde{U}_I\rangle$, which is maximum when the evolution is noiseless. If additionally one wants to implement a quantum gate G, then we further require that $U_{\text{ctrl}}\rho U_{\text{ctrl}}^\dagger \to G\rho G^\dagger$. Regardless of our objective, it is clear that one needs to be able to predict $V_O(T)$ given (i) the actual noise affecting the qubit and (ii) a choice of control. However, realistically the information available about the noise is limited, and by the very definition of an open quantum system is something that cannot typically be measured directly.

Fortunately, this limitation can in principle be overcome by quantum noise spectroscopy (QNS) protocols [23, 82–93]. These protocols exploit the measurable response of the qubit to a known and variable control and the noise affecting it, in

order to infer information about the noise. The type of accessible information is statistical in nature. That is, without any other information, e.g., about the type of stochastic noise process, the best one can hope to learn are the bath correlation functions $\langle \beta_{\alpha_1}(t_1) \cdots \beta_{\alpha_k}(t_k) \rangle$. If the QNS protocol is sufficiently powerful to characterize the leading correlation functions and matches the model, in principle the inferred information can be plugged into a cumulant expansion or a Dyson series expansion of $V_O(t)$ to successfully obtain an estimate of the operator for any choice of $f_\alpha(t)$, as desired. This has led to a proliferation of increasingly more powerful QNS protocols, including those capable of characterizing the noise model described here [90], some of which have even been experimentally verified [23, 82–84, 87, 88, 93, 113, 114]. More generally, the idea of optimizing control procedures to a known noise spectrum [101] is behind some of the most remarkable coherence times available in the literature [115].

QNS protocols, however, are not free of complications. The demonstrated success of these protocols relies on the assumptions which support them being satisfied. Different protocols have different assumptions, but they can be roughly grouped into two main flavours:

- *Assumptions on the noise.*— Existing protocols assume that the only a certain subset of the correlation functions effectively influence the dynamics or, equivalently that a perturbative expansion ov $V_O(t)$ can be effectively truncated to a fixed order. In practice, this is enforced in various ways. For example, demanding that the noise is Gaussian and dephasing or, more generally, that one is working in an appropriately defined "weak coupling" regime [85, 89, 91].

- *Assumptions on the control.*— Many QNS protocols, especially those based around the so-called "frequency-comb" [23, 85, 89, 90], rely on specific control assumptions, such as that pulses are instantaneous. This assumption facilitates the necessary calculations, which ultimately allow the inferring of the noise information. However, it enforces constraints on the control that translate into limitations on the QNS protocol, e.g., a maximum frequency sampling range [23, 85]. Moreover, experimentally one cannot realize instantaneous pulses, so comb-based QNS protocols are necessarily an approximation with an error that depends on how far the experiment is from satisfying the instantaneous pulse assumption.

This work overcomes these limitations by bypassing the step of inferring the bath correlation functions. We maintain the philosophy of QNS regarding characterizing the open quantum system dynamics, but pose it in a machine learning context. Thus, we address the question:

Can an appropriately designed machine learning algorithm "learn" enough about the open quantum system dynamics (relative to a given set of control capabilities), so as to be able to accurately predict its dynamics under an arbitrary element of the aforementioned set of available controls?

We answer positively to this question by implementing such ML-based approach. Concretely our ML algorithm (i) learns about the open quantum system dynamics and (ii) is capable of accurately estimating – without assuming a perturbative expansion – the operator $V_O(T)$, and consequently measurement outcomes, resulting from a control sequence picked from the family control pulses $\{f_\alpha(t)\}$ specified by an assumed (but in principle arbitrary) set of control capabilities.

5.3 Methods

In this section we present in detail the proposed method to solve the problem under consideration. We start by giving an overall summary of our proposed solution. Next, in Section 5.3.2 we discuss some of the mathematical properties of the V_O operator. This will allow us to find a suitable parametrization that will be useful to build the architecture of the ML model. Next, we present exactly the architecture of the ML model in Section 5.3.3. After that, we give an overview on how to construct datasets in order to train and test the model in Section 5.3.4. Finally, in Section 5.3.5 we conclude with the training and testing procedures, including the metrics used to assess the performance of the proposed model.

5.3.1 Overview

The ML approach naturally matches our control problem, which becomes clear from the following observation. For most optimal control applications, e.g. achieving a target fidelity for a gate acting on an open quantum system, one does not need to have full knowledge of the noise. To see this consider a hypothetical scenario where the available control is band limited [88,91,116], i.e., whose frequency domain representation $F(\omega)$ is compactly supported in a fixed frequency range $|\omega| \leq \Omega_0$. If the response of the open quantum system to the noise [117] is captured by a convolution of the form $I = \int_{-\infty}^{\infty} d\omega F(\omega)S(\omega)$, where $S(\omega)$ represents the noise power spectrum, then it is clear that one only needs to know $S(\omega)$ for $|\omega| \leq \Omega_0$. While this statement can be formalized and made more general, we do so Ref. [103], the above example captures a key point: only the "components" of the noise that are relevant to the available control need to be characterized. Conversely, this means that a fixed set of resources, e.g., a set of control capabilities, can only provide

information about the "components" of the noise relevant to them. The above observations make the ML approach particularly well suited for the problem: it is natural to draw the connection between the control problem of "characterizing a system with respect to a restricted set of control capabilities in order to predict the dynamics under any control such capabilities can generate" and the fact that the training and testing datasets typical in ML make sense when the datasets are generated in the same way, i.e., by the same "control capabilities". Of course, the details of the ML approach which can seamlessly integrate with the quantum control equations are important, and we now provide them.

In order to address the question presented at the end of section 5.2, we are going to use an ML graybox based approach similar to the one presented in [104]. The basic idea of a graybox is to divide the ML model into two parts, a blackbox part and a whitebox part. The blackbox part is a collection of standard ML layers, such as neural networks (see Section 2.1.1 for an overview), that allows us to learn maps between variables without any assumptions on the actual relation. The whitebox part is a collection of customized layers that essentially implement mathematical relations that we are certain of. This approach is better than a full blackbox, because it allows us to estimate physically relevant quantities, and thus enables us to understand more about the physics of the system. In other words, the blackboxes are enforced to learn some abstract representations, but when combined with the whiteboxes we get physically significant quantities. In the parlance of machine learning, these whitebox layers are "quantum features", which extract the expect patterns in the data fed to the network.

In the case of the problem under consideration, we are going to use the blackbox part to estimate some parameters for reconstructing the V_O operators. The reason behind the use of a blackbox for this task is because the calculation of the V_O operators depends on assumptions on the noise and control signals. So, by using a blackbox we get rid of such assumptions. Whereas the whitebox parts would be used for the other standard quantum calculations that we are certain of, such as the time-ordered evolution, and quantum expectations. Thus, we end up with an overall graybox that essentially implements Equation 5.6, with input representing the control pulses, output corresponding to the classical expectation of quantum observable over the noise, and internal parameters modelling abstractly the noise and its interaction with the control. With this construction, we would be able to estimate important quantities such as V_O, and U_{ctrl}. Now, since we are using machine learning, then we will need to perform a training step to learn the parameters of the blackboxes. Thus, the general protocol would be as follows:

1. Prepare a dataset consisting of pairs of random input pulse sequences applied to the qubit (chosen from a fixed and potentially infinite set of allowed se-

quences), and the measured outcomes after evolution.

2. Initialize the internal parameters of the graybox model.

3. Train the model for some number of iterations until convergence.

4. Fix the trained model and use it to predict measurement outcomes for new pulse sequences as well as the V_O operators.

It should be noted the nature of quantum information enforces characterization of quantum systems of large dimension to blow-up. This is a common problem in any characterization protocol including quantum state tomography, quantum process tomography, quantum noise spectroscopy, and even quantum control unless some simplifying assumptions are made. However, in most practical situations, a full characterization may not be required. A full characterization of a set of small subsystems can be sufficient. This is the case for quantum computing where arbitrary multi-qubit quantum gates are compiled into 1- and 2- qubit gates. Alternatively, a partial characterization of the whole system can be sufficient. This is the case for quantum error correction where only a small subset of observables (error syndromes) is of interest. In either case, using the proposed machine learning framework will remain feasible.

5.3.2 Mathematical Properties of the V_O Operator

If we look back into the definition of the V_O operator, we find that it is convenient to express it as

$$V_O = O^{-1} \langle \tilde{U}_I^\dagger O \tilde{U}_I \rangle \equiv O^{-1} \langle W_O \rangle_c, \tag{5.8}$$

because the observable is independent of the noise. As a result we can see that W_O is a system-only operator with the following properties. Namely, for a given realization of the classical noise process and recalling that O is a traceless qubit-only (system-only) observable, one has that

1. The trace of W_O is bounded. This follows by noting that

$$\begin{aligned} \mathrm{tr}[W_O] &= \mathrm{tr}_S[O \,\mathrm{tr}_B[\tilde{U}_I(I_S \otimes \rho_B)\tilde{U}_I^\dagger]] &(5.9)\\ &= \mathrm{tr}_S[O\mathcal{E}(I_S)] \in [-1, 1], &(5.10)\end{aligned}$$

where $\mathcal{E}(\cdot) = \mathrm{tr}_B[\tilde{U}_I(\cdot \otimes \rho_B)\tilde{U}_I^\dagger]$ is a CPTP map. In the special case where the noise generates a unital channel, i.e., when $\mathcal{E}(I_S) = I_S$, as is the case for classical-only noise, then $\mathrm{tr}[W_O] = 0$, since O is a traceless observable.

2. W_O is Hermitian. One can see this by noting that W_O can always be written as $W_O = \sum_{\alpha=0,x,y,z} \mathrm{tr}_S[\sigma_\alpha \,\mathrm{tr}_B[\tilde{U}_I^\dagger O \tilde{U}_I (I_S \otimes \rho_B)]]\, \sigma_\alpha/2$. Since

$$\begin{aligned}
\mathrm{tr}_S[\sigma_\alpha \,\mathrm{tr}_B[\tilde{U}_I^\dagger O \tilde{U}_I (I_S \otimes \rho_B)]] &= \mathrm{tr}_{SB}[O \tilde{U}_I (\sigma_\alpha \otimes \rho_B) \tilde{U}_I^\dagger] && (5.11)\\
&= \mathrm{tr}_S[O\mathcal{E}(\sigma_\alpha)] && (5.12)\\
&= \mathrm{tr}_S\left[O\left(\mathcal{E}\left(\frac{I_S+\sigma_\alpha}{2}\right) - \mathcal{E}\left(\frac{I_S-\sigma_\alpha}{2}\right)\right)\right] \in \mathbb{R}, && (5.13)
\end{aligned}$$

   it follows that $W_O = W_O^\dagger$.

3. Given the above and observing that $\mathrm{tr}_S[\sigma_\alpha W_O] = \mathrm{tr}_S[O\mathcal{E}(\sigma_\alpha)] \in [-1,1]$, it follows that W_O has real eigenvalues $\lambda_1, \lambda_2 \in [-1,1]$.

If one also considers the effect of the average over realizations of the stochastic process, one further finds that the full average (including classical and quantum components) satisfies the following properties

1. Its trace is bounded, i.e., $\mathrm{tr}[\langle W_O \rangle_c] \in [-1,1]$.

2. It is Hermitian, i.e., $\langle W_O \rangle_c^\dagger = \langle W_O^\dagger \rangle_c = \langle W_O \rangle_c$.

3. Since for any realization W_O has real eigenvalues in $[-1,1]$, the average over realizations of that process will also have that property, i.e., its eigenvalues are such that $-1 \leq \lambda\left(\langle W_O \rangle_c\right) \leq 1$. This can be proved as follows. Suppose for the sake of convenience that the probability distribution of the W_O with respect to the noise is finite and discrete. That is it can only take values $\tilde{W}_i$ with probability P_i. Then

$$\begin{aligned}
\lambda_{\max}\left(\langle W_O \rangle_c\right) &= \lambda_{\max}\left(\sum_{i=1}^{i_{\max}} P_i \tilde{W}_i\right) && (5.14)\\
&= \lambda_{\max}\left(P_1 \tilde{W}_1 + \sum_{i=2}^{i_{\max}} P_i \tilde{W}_i\right) && (5.15)\\
&\leq \lambda_{\max}\left(P_1 \tilde{W}_1\right) + \lambda_{\max}\left(\sum_{i=2}^{i_{\max}} P_i \tilde{W}_i\right) && (5.16)\\
&= P_1 \lambda_{\max}\left(\tilde{W}_1\right) + \lambda_{\max}\left(\sum_{i=2}^{i_{\max}} P_i \tilde{W}_i\right) && (5.17)\\
&\leq P_1 + \lambda_{\max}\left(\sum_{i=2}^{i_{\max}} P_i \tilde{W}_i\right). && (5.18)
\end{aligned}$$

   The third line follows from Weyl's inequality since all the terms of the form $P_i \tilde{W}_i$ are Hermitian. Now, if we repeat recursively the same steps on the

second remaining term, we get

$$\lambda_{\max}\left(\langle W_O\rangle_c\right) \leq P_1 + P_2 + \cdots P_{i_{\max}} \tag{5.19}$$

$$= 1, \tag{5.20}$$

as the P_i's form a probability distribution. Similarly, we can show that

$$\lambda_{\min}\left(\langle W_O\rangle_c\right) \geq \lambda_{\min}\left(\sum_{i=1}^{i_{\max}} P_i \tilde{W}_i\right) \tag{5.21}$$

$$\geq -1. \tag{5.22}$$

and so by combining the two results we get

$$-1 \leq \lambda\left(\langle W_O\rangle_c\right) \leq 1 \tag{5.23}$$

This proof can be extended to the more realistic situation when the noise distribution is continuous. Thus by specifying a diagonal matrix D whose entries are real numbers in the interval $[-1, 1]$ adding up to a number that lies in the [-1,1], and by choosing a general unitary matrix Q, we can reconstruct any V_O operator in such a way that satisfies its mathematical properties, using the eigendecomposition

$$V_O = \langle O^{-1} U_I^\dagger O U_I \rangle = O^{-1} Q D Q^\dagger. \tag{5.24}$$

In particular for the case of qubit presented in this chapter (i.e. $d = 2$), we can completely specify the V_O operator using 6 parameters which we would refer to as ψ, θ, Δ, λ_1, and λ_2 such that

$$Q = \begin{pmatrix} e^{i\psi} & 0 \\ 0 & e^{-i\psi} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} e^{i\Delta} & 0 \\ 0 & e^{-i\Delta} \end{pmatrix}, \tag{5.25}$$

where we neglected a degree of freedom that represents an overall global phase shift, and

$$D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \tag{5.26}$$

satisfying $|\Lambda| = |\lambda_1 + \lambda_2| \leq 1$. The parameters ψ, θ, and Δ can take any real values as they are arguments of periodic functions.

In this chapter we will focus on the unital case, i.e., $\lambda_1 + \lambda_2 = 0$, since we will validate our ML approach via simulated experiments using classical-only noise. In

this case, we we can write the eigenvalue matrix as

$$D = \begin{pmatrix} \mu & 0 \\ 0 & -\mu \end{pmatrix}, \tag{5.27}$$

and $\mu \in [0, 1]$. In what follows, when building our ML machinery, we will specify how this constraint is implemented and how it can be obviated in each layer. Note that in an actual experiment, where no assumption on the classicality of the noise is valid a priori, the general, non-unital, case should be considered.

5.3.3 Model Architecture

The proposed graybox ML model is shown in Figure 5.1. We shall explain in detail the structure as follows.

5.3.3.1 Model inputs and outputs

The purpose of the proposed architecture is to have a model that relates the control pulses applied on the qubit (which we have control over in an experiment) to the classical average of the quantum observables (which we can physically measure). The model internal parameters will act as an abstract representation for the noise, as well as how it affects the measurement outcomes. The model inputs represent different "features" extracted from the control pulse sequence. In this chapter, we make use of two independently-processed features, and thus the model has two inputs. The first feature is a global feature of the control pulse: the parametrization of the pulse. The other is a local feature: the time-domain representation of the pulse. These two features are explained as follows.

The first feature represents the set of parameters defining the control pulse sequence. We assume in this chapter that the control signal can be parametrized by a finite set of parameters. This still allows having infinitely large number of possible control signals since each of the parameters can take infinitely many values. For example, a train of N Gaussian pulses can be completely defined by $3N$ parameters: the amplitude, mean, and variance of each of the N pulses. Similarly, a train of square pulses can be defined by each of the pulse positions, pulse widths, and pulse amplitudes. Now, these parameters have to be represented in a way that is suitable for the subsequent blocks (standard ML blackboxes) to process. So, the first step is to normalize each of signal parameters to be in the range of $[0, 1]$ across all the examples. For instance, the pulse locations can be normalized with respect to the total evolution time T since there will be no pulses beyond this point in time. The pulse amplitudes could also be normalized such that the maximum amplitude for

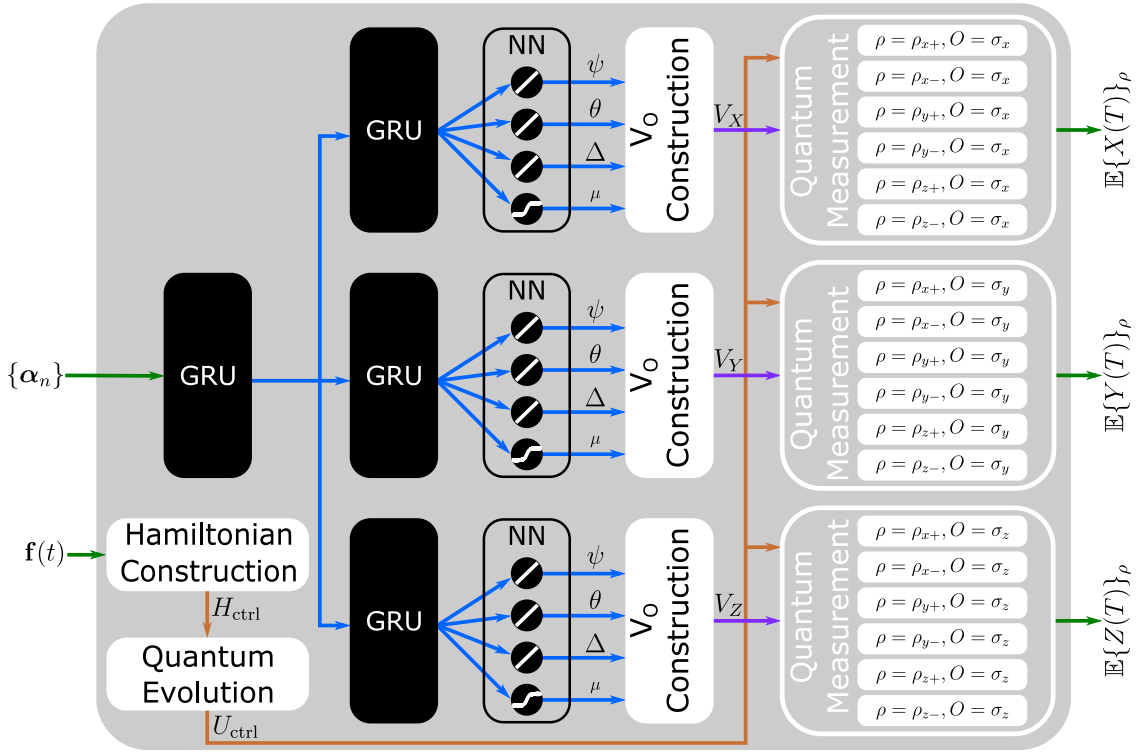


Figure 5.1: The proposed graybox architecture for modeling the noisy qubit following Equation 5.6. The inputs of the model are the sequence of control signal parameters $\{\boldsymbol{\alpha}_n\}$, and the actual time-domain waveform $\mathbf{f}(t)$. The outputs of the model are the expectations over the noise for all the Pauli eigenstates as initial states, and all Pauli's as measurement operators. The black box part of the model consist of two layers of GRU followed by a neuron layer. The output of this layer represents the parameters that can be used to construct the "V_O" operator. There are three different branches corresponding to each of the three possible Pauli observables. The whitebox part of the model is formed from the layers that implement specific formulas known from quantum mechanics. This includes layers for constructing the V_O operators from the parameters generated from the blackbox, constructing the control Hamiltonian from the time domain representation of the control pulse sequence, the time-ordered evolution to generate the control unitary, and the quantum measurement layer. The model is trained using a set of pairs of control pulse sequence and corresponding expectation values of the observables. After training, the model can be used to predict the measurements for new pulse sequences. It can also be probed to estimate one of the "V_O" operators, and thus can be used as a part of a quantum control algorithm to achieve a desired quantum gate.

any pulse sequence is 1. The second is step is the proper formatting of these parameters. We choose to organize the signal parameters of an $n_{\max}$ pulse train in the form of a sequence of vectors $\{\boldsymbol{\alpha}_n\}_{n=1}^{n_{\max}}$, where each vector represents the n^{th} pulse and has r entries representing the normalized pulse parameters (example: Gaussian pulse train will have $r = 3$). For the case of multi-axis control, we concatenate the parametrization along each direction into one vector assuming the controls are independent along each direction. We emphasize here that we take Gaussian and square pulses as examples to demonstrate our ideas, but in general any waveform with any suitable parametrization could be used. This feature is considered global since every parameter affects the whole waveform shape of control sequence.

The second feature used for the model is the actual time domain representation of the pulse sequence, discretized into M steps. In other words, the amplitude of the pulse at each time step. This feature is considered local, since a change in one of the parameters does not affect the whole sequence. This input is only processed by customized whiteboxes. Although in principle we can calculate the time-domain representation from the signal parameters in the first input, it turns out that the overall algorithm performs better if we do not do this calculation directly. In other words treat both features as two "independent inputs" to the model.

The output of the model should be the measurement outcomes. If we initialize our qubit to each of the eigenstates ("up/down") of each Pauli operators (that is a total of 6 states), and measure the three Pauli operators, then we have enough information (tomographically complete) to predict the dynamics for other configurations. So, we need to perform a total of 18 "prepare-measure" experiments and collect their results. And so, our model will have 18 outputs corresponding to each of the measurement settings.

5.3.3.2 Model whiteboxes

As discussed previously, there are lots of known relations from quantum mechanics that we are certain of. It is better in terms of the overall performance to directly implement as much as possible of these relations in non-standard customized layers. This saves the machine from essentially having to learn everything about quantum mechanics from the data, which would be hard and could decrease the overall accuracy. Moreover, this allows us to evaluate physically significant quantities which is one of the most important general advantages of the graybox approach. In our proposed model, we make use of the following whiteboxes.

- Hamiltonian Construction

  This layer takes the discretized time domain representation of the control pulses (which is exactly the second input to the model), and outputs the

control Hamiltonian H_{ctrl} evaluated at each of the M time steps using Equation 5.2. The layer is also parametrized by the energy gap Ω which we fix at the beginning.

- Quantum Evolution

  This layer follows the "Hamiltonian Construction" layer, and thus it takes the control Hamiltonian at each time step as input and evaluates the time-ordered quantum evolution as output (i.e. Equation 5.4). Numerically, this is calculated using the approximation of an infinitesimal product of exponentials

$$
\begin{aligned}
U_{\text{ctrl}} &= \mathcal{T}_{+} e^{-i\int_0^T H_{\text{ctrl}}(s)ds} && (5.28)\\
&= \lim_{M\to\infty} e^{-iH_{\text{ctrl}}(t_M)\Delta T} e^{-iH_{\text{ctrl}}(t_{M-1})\Delta T}\cdots e^{-iH_{\text{ctrl}}(t_0)\Delta T} && (5.29)\\
&\approx e^{-iH_{\text{ctrl}}(t_M)\Delta T} e^{-iH_{\text{ctrl}}(t_{M-1})\Delta T}\cdots e^{-iH_{\text{ctrl}}(t_0)\Delta T}, && (5.30)
\end{aligned}
$$

  where $t_k = k\Delta T$ and $\Delta T = \frac{T}{M}$. The last line follows if M is large enough.

- V_O Construction

  This layer is responsible for reconstructing the V_O operator. It takes the parameters ψ, θ, Δ, and μ as inputs and outputs the V_O following the reconstruction discussed in Section 5.3.2. The blackboxes of the overall model are responsible for estimating those parameters. The output of this layer can be probed to estimate the V_O operator given a control pulse. This allows us to do further operations including noise spectroscopy and quantum control. In the general case for non-unital dynamics, the reconstruction will require the two parameters $\lambda_1 \in [-1, 1]$ and $\Lambda = \lambda_1 + \lambda_2 \in [-1, 1]$, instead of the parameter $\mu \in [0, 1]$.

- Quantum Measurement

  This layer is essentially the implementation of Equation 5.6. So, it takes the V_O operator as input, together with the control unitary, and outputs the trace value. It is parameterized by the initial state of the qubit, as well as the observable to measure. Therefore, in order to calculate all possible 18 measurements, we need 18 of such layers in the model, each with the correct combination of inputs and parameterization. The outputs of all 18 layers are concatenated finally and they represent the model's output.

5.3.3.3 Model blackboxes

The exact calculation of the measurement outcomes requires assumptions on both the noise and control pulse sequence. So, by using the standard ML blackbox layers,

such as Neural Networks (NN) and Gated Reccurent Units (GRU) (see Section 2.1.1 for an overview), we can have an abstract assumption-free representation of the noise and its interaction with the control. This would allow us to estimate the required parameters for reconstructing the V_O operators using a whitebox. The power of such layers comes from their effectiveness in representing unknown maps due to their highly non-linear complex structure. In our proposed model we have three such layers explained as follows.

- Initial GRU

  This layer is connected to the first input of the model, i.e. the parameters of the control pulse sequence. The purpose behind this layer is to have an initial pre-processing of the input features. Feature transformation is commonly used in ML algorithms, to provide a better feature space that would essentially enhance the learning capability of the model. In the modern deep learning paradigm, instead of doing feature transformation at the beginning, we actually integrate it within the overall algorithm. In this way, the algorithm learns the best optimal transformation of features that increases the overall accuracy. In our application, the intuition behind this layer is to have some sort of abstract representation of the interaction unitary U_I. This would depend on the noise as well as the control. In this sense, the input of the layer represents the control pulses, the output represents the interaction evolution operator, and the weights of the layer represent the noise. This does not mean that probing the output layer is exactly related to the actual U_I as the algorithm might have a completely different abstract representation, which is a general feature of blackboxes. In the proposed model, we choose the GRU unit to have 10 hidden nodes.

  The reason behind choosing the GRU, which is a type of a recurrent neural network (RNN), rather than a standard neural network, is to make use of the feedback mechanism. We would like to have a structure that processes a sequence not only term by term, but taking into account previous terms. An RNN-like structure would accomplish this kind of processing. In this sense, we can abstractly model relations that involve convolution operations which is exactly the type of relations obtained when we do theoretical calculations of the dynamics of open quantum systems.

- Final GRU

  This is another GRU layer that is connected to the output of the initial GRU layer. The purpose of this layer is to increase the complexity of the blackboxes so that the overall structure is complex enough to represent our relations. For

our application, this layer serves as a way to estimate the operator $\langle U_I^\dagger O U_I \rangle$ in some abstract representation. And thus we need actually three of such layers to correspond to the three Pauli observables. We choose to have 60 hidden nodes for each of these layers.

- Neural Network

  This is a fully-connected single neural layer consisting of four nodes. The output of the final GRU layer is connected to each of the nodes. The first three nodes have linear activation function and their output represent the actual parameters ψ, θ, and Δ that are used to construct the V_O operator. The last node has a sigmoid activation function and its output corresponds exactly to the μ parameter of the V_O operator. As discussed before the μ parameter has to be in the range $[0, 1]$ which is exactly the range of the sigmoid function. Since the parameters will differ for each of the three observables, we need three of such layers each connected to one of the final GRU layers. In the case of non-unital dynamics, the neural network will change as follows. First, the fourth node will have a hyperbolic tangent (tanh) activation (whose range is $[-1, 1]$) and would represent the parameter λ_1 instead of μ. Second, a fifth node with a hyperbolic tangent activation will be needed to represent the parameter Λ. In this way, the constraints of the V_O operators will be satisfied, and hence the architecture would be suitable for modeling the non-unital dynamics.

5.3.4 Dataset Construction

For any ML algorithm, we need a dataset for training it and testing its performance. A dataset is a collection of "examples" (instances), where each example is a pair of inputs (that should be fed to the algorithm) and corresponding actual outputs or "labels" (that the algorithm is required to predict). The dataset is usually split into two parts, one part is used for training the algorithm while the other part is used for testing its performance. The examples belonging to the testing portion of the dataset are not used in the training process at all. In the application presented in this chapter, the dataset would be a collection of control pulse sequences applied to a quantum system, and the corresponding measured quantum observables. In the experimental situation such a dataset can be constructed as follows. We prepare the qubit in an initial state, apply some a control sequence, then measure the observable. This is repeated for all 18 possible configurations of initial states/observables. The pair consisting of the control pulse sequence parametrization and time-domain representation, and the value of the 18 measurements would correspond to one example in the dataset. To generate more examples, we choose different control sequences and repeat the whole process. In this chapter, we do not have access to an actual

experiment. However, we need to validate the proposed algorithm, so we generate datasets by computer simulation of a noisy qubit (See Sections 5.4 and 2.3.4 for details on how the simulations were done). In a practical application of the algorithm with experimental data, there is no need to use simulated datasets. The algorithm will be trained directly from the acquired data.

An important aspect of this discussion is how to choose the examples constituting the dataset and how large the dataset should be. This is an empirical process, but there are general principles to follow. First, any ML algorithm should be able to generalize, i.e. predict the outcomes for examples that were not in the training portion of the dataset. The way to ensure this is to have the training subset large enough to represent wide range of cases. For instance, consider constructing a dataset of CPMG-like sequences [15], i.e., sequences composed of equally spaced pulses. Then, in order for the model to have the capability of predicting the correct outcomes if the pulses are shifted (maybe due to some experimental errors), we have to provide training examples in which the control pulses are randomly shifted. Similarly, if we want accurate predictions for control pulses that have powers other than π, then we need to include such examples for training. Additionally, the prediction would work for example of the same pulse shape. This means that if all training examples are square pulses then the predictions would be accurate only for square pulses. In case we need the trained algorithm to predict outcomes corresponding to pulses of different shapes, then we must include examples of all desired shapes in the dataset (for example square and Gaussian pulses). This would result in an increase in the complexity of training due to two reasons. First, more computations will be required as we have to increase the size of the dataset to include examples of the different pulse shapes. The second reason is that consequently we need to increase the learning capacity of the blackboxes (by increasing the number of nodes in the neural networks, GRU's,...etc.). However, this is not a problem in practice specially in actual physical experiments. The reason is that usually there are constraints about what types of control sequences can be generated. This would depend on the specifications of the available hardware in the experiment. For example, it might be only possible to generate square pulses with some maximum amplitude, or Gaussian pulses with a fixed pulse duration. In such cases, we will be interested to predict and control the dynamics of a quantum system driven by that particular kind of pulses only. Therefore, the dataset needs to consist of examples of control sequences whose pulse shape is of interest. This would result in a smaller dataset and the training complexity decreases. Thus, experimental constraints arising from the specifications of the available hardware defines what control capabilities are available, which in turn simplifies the process of training the ML algorithm. In this chapter, we do not consider training the algorithm on datasets consisting of different pulse shapes.

However, we construct different datasets and train different "instances" of the algorithm on each dataset independently. This is done for the purpose of proving the concept, and showing that the proposed algorithm works with different pulse shapes. Section 5.4 gives details about the different datasets we constructed in this chapter, as well as the results of training and testing the algorithm on each of the datasets individually.

The size of the dataset (i.e. the number of instances) is usually chosen empirically. The general rule of thumb is that the more training examples are used, the better the generalization of the algorithm is. Moreover, the more testing examples are used in the assessment process, the better the assessment is. So, usually the procedure is iterative where we start with some size for the dataset, train the algorithm, assess the performance (see Section 5.3.5 for details), and then decide whether the size is sufficient or not. Note, however, that we might be constrained by some upper limit on the dataset size before the process of data collection becomes infeasible. This is one of the main challenges facing the design of modern ML algorithms.

5.3.5 Training and Testing

The second step after constructing the dataset, is to choose a loss function for the model. This a function that measures how accurate the outputs predicted by the model compared to the true outputs. This choice depends on the application under consideration. In our case, we shall use the Mean-Square-Error (MSE), averaged over all 18 measurement outcomes. The weights of the model are chosen such that the loss function is minimized. Ideally, we seek a global minimum of the loss function but in practice this might be hard and we probably end up with a local minimum. However, practically, this usually provides sufficient performance.

The third step is to choose an optimization algorithm. The optimization is for finding the weights of the model that minimizes the loss function averaged over all training examples. The standard method used in ML is backpropagation which is essentially a gradient descent based method combined with an efficient way of calculating the gradients of the loss function with respect to the weights. There are many variants of the backpropagation method in the literature, the one we choose to use in this chapter is the Adam algorithm [118]. There exist also other gradient-free approaches such as Genetic Algorithm (GA) based optimization [106].

The fourth step is to actually perform the training. In this case, we initialize the weights of the model to some random values, then apply enough iterations of the optimization algorithm till the loss function reaches a sufficiently small value. In the case of MSE, we would like it ideally to be as close as possible to 0, but this could require infinite number of iterations. So, practically we stop either when we

reach sufficient accuracy or we exceed a maximum number of steps.

A final thing to mention is that because the whiteboxes do not have any trainable parameters, the blackboxes are enforced through the training to generate outputs that are compatible with the whiteboxes, so that we end up with the correct physical quantities.

After executing the aforementioned steps for training the algorithm, it will be able to predict accurately the outputs of the training examples. In our application, this by itself is useful because we can easily probe the output of the V_O layers and use that prediction for various purposes. However, we have to ensure the model is also capable of generalizing to new examples. This is where the portion of the dataset used for testing comes in place. The trained algorithm is used to predict the outcomes of the control pulse sequences of the testing subset, and the average MSE is calculated. Next, the testing MSE is compared to the training MSE. If the testing MSE is sufficiently low then this indicates the model has good predictive power. Ideally, we need the MSE of the testing subset to be as close as possible to the MSE of the training subset. Sometimes this does not happen and we end up with the testing MSE being significantly higher than that of the training MSE. This is referred to as overfitting. In order to diagnose this behaviour we usually plot the MSE of both subsets evaluated at each training iteration on the same axes, versus the iteration number. Note, that the testing examples or the testing MSE do not contribute at all to the training process of the algorithm, they are just used for performance evaluation. If both curves decrease as the number of iterations increase, until reaching a sufficiently low level, then the model has a good fit. If the testing MSE saturates eventually or worse starts increasing again, then the algorithm is overfitting. There are many methods proposed in the classical machine learning literature to overcome overfitting including decreasing the model complexity, increasing the number of training examples, and early stopping (i.e. stop the iterations before the testing MSE of the testing starts to increase). On the other hand, the significance of overfitting on the performance of a model depends on the application, and the required level of accuracy. This means that a model might be experiencing some overfitting behaviour, but the prediction accuracy is still sufficient.

5.4 Simulation Results

In this section, we describe the numerical simulations we performed in order to verify the proposed method. We chose to create six datasets of different pulse configurations to train and test the ML structure. This is described in Section 5.4.1. Next, in Section 5.4.2 we present the performance analysis results regarding the

accuracy of trained models for each of the datasets. In Section 5.4.3, we show the applicability of using our trained model to do standard tasks such as decoherence suppression and quantum control. Finally, we discuss the significance of these results in 5.4.4

5.4.1 Implementation

We implemented the proposed protocol using the "Tensorflow" Python package [7], and its high-level API package "Keras" [8]. The code is publicly available[1]. We also implemented a noisy qubit simulator, to generate the datasets for training and testing. It simulates the dynamics of the qubit using Monte Carlo method rather than solving a master equation, to be general enough to simulate any type of noise. The details of the design and implementation of this simulator are presented in Section 2.3.4. We chose the simulation parameters as shown in Table 5.1.

Parameter	**Description**	**Value**
T	Evolution time	1
M	Number of discrete time steps	4096
K	Number of noise process realizations	1000
Ω	Energy gap	10

Table 5.1: The different simulation parameters used for generating the datasets.

In this chapter, we selected the number of noise realizations based on doing the Monte Carlo simulation of a random pulse sequence, and then observing how much the expectation values change by increasing the number of realizations. As shown in Figure 5.2, the values start to stabilize around 500 realizations, so we chose $K = 1000$ for generating all the datasets.

We created three categories of datasets using the simulator, summarized in Table 5.2, as follows. The datasets are publicly available[2].

1. Qubit with noise on a single-axis and control pulses on an orthogonal axis.

   The Hamiltonian in this case takes the form

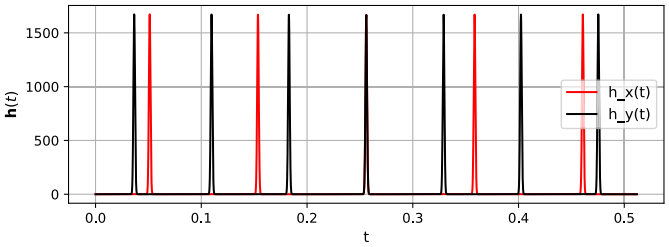
   $$H = \frac{1}{2}\left(\Omega + \beta_z(t)\right)\sigma_z + \frac{1}{2}f_x(t)\sigma_x. \tag{5.31}$$

   We chose the noise to have the following power spectral density (single-side band representation, i.e. the frequency f is non-negative)

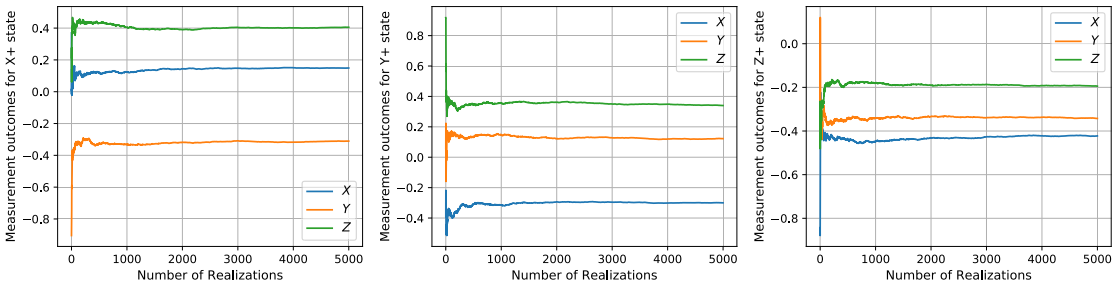
   $$S_Z(f) = \begin{cases} \frac{1}{f+1} + 0.8e^{-\frac{(f-20)^2}{10}} & 0 < f \leq 50 \\ 0.25 + 0.8e^{-\frac{(f-20)^2}{10}} & f > 50 \end{cases} \tag{5.32}$$

[1] `https://github.com/akramyoussry/BQNS`

[2] `https://doi.org/10.6084/m9.figshare.11967465.v1`



(a) Control Pulses



(b) $\rho = X+$ (c) $\rho = Y+$ (d) $\rho = Z+$

Figure 5.2: The effect of the number of realizations on the performance of the Monte Carlo simulation of a noisy qubit driven by the pulses in (a). The three Pauli observables are plotted versus the number of noise realizations over which the averages are calculated. In (b), (c), and (d), different initial states are selected. The plot shows that starting around 500 realizations, the statistical fluctuations start to decrease. We choose 1000 realizations to perform all the simulations in this chapter.

Figure 5.3a shows the plot of this power spectral density. The reason for choosing such a form is to ensure that the resulting noise is general enough, while also covering some special cases (such as $1/f$ noise). Also, the total power of the noise is chosen such the effect of noise is evident on the dynamics (i.e. having coherence < 1). In this category, we generated two datasets. The first one is for CPMG pulse sequences with Gaussian pulses instead of the ideal delta pulses. So, the control function takes the form

$$f(t) = \sum_{n=1}^{n_{\max}} A e^{-\frac{(t-\tau_n)^2}{2\sigma^2}}, \tag{5.33}$$

where $\sigma = \frac{6T}{M}$, and $A = \frac{\pi}{\sqrt{2\pi\sigma^2}}$, and $\tau_n = \left(\frac{n-0.5}{n_{\max}}\right) T$. The highest order of the sequence was chosen to be $n_{\max} = 28$. Now, this means we have a set of 28 examples only in the dataset. In order to introduce more examples in the dataset, we randomize the parameters of the signal as follows. The position of the n^{th} pulse of a given sequence is randomly shifted by a small amount δ_τ chosen at uniform from the interval $[-6\sigma, 6\sigma]$. As a result, we lose the CPMG property that all pulses are equally spaced. However, this can be useful experimentally when there is jitter noise on the pulses. Additionally, we

also randomize the power of the pulse. In this case, we vary the amplitude A by scaling it with randomly by amount δ_A chosen at uniform from the interval $[0, 2]$. For this randomization, we scale all the pulses in the same sequence with the same amount. Again we lose the property of CPMG sequences that they are $\pi-$ pulses, but this is needed so that the algorithm can have sufficient generalization power. With these two sources of randomness, we generate 100 instances of the same order resulting in a total of 2800 examples. Finally, we split randomly the dataset following the 75:25 ratio convention into training and testing subsets.

The second dataset in this category is very similar with the only difference being the shapes of the pulses. Instead of Gaussian pulses we have square pulses with finite width. The control function takes the form

$$f(t) = \sum_{n=1}^{n_{\max}} Au(t - \tau_n - 0.5\sigma)u(\tau_n + 0.5\sigma - t), \tag{5.34}$$

where $u(\cdot)$ is the unit step function, $\sigma = \frac{6T}{M}$, and $A = \frac{\pi}{\sigma}$. The same scheme for randomization and splitting is used in this dataset.

2. Qubit with multi-axis noise, and control pulses on two orthogonal directions.

   The Hamiltonian in that category takes the form

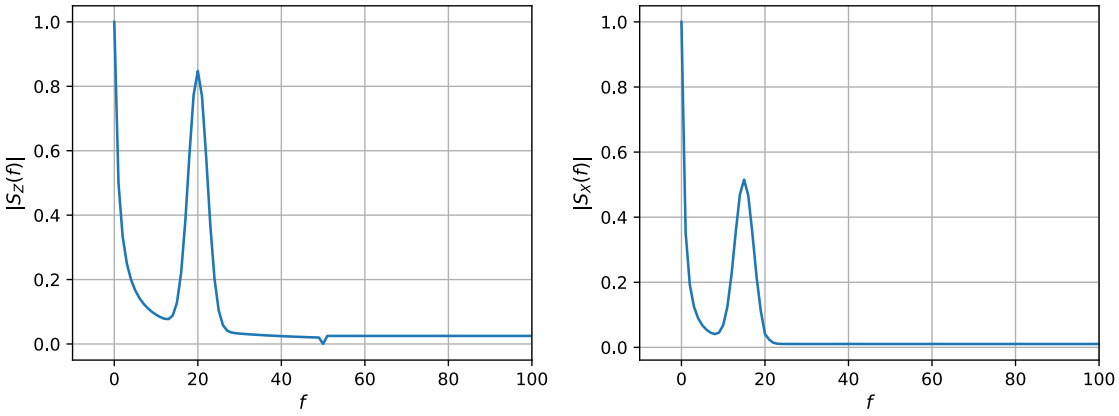
$$H = \frac{1}{2}\left(\Omega + \beta_z(t)\right)\sigma_z + \frac{1}{2}\left(f_x(t) + \beta_x(t)\right)\sigma_x + \frac{1}{2}f_y(t)\sigma_y \tag{5.35}$$

   We chose the noise along $z-$ axis to have the same power spectral density as in Equation 5.36, while the noise along the x-axis has the power spectral density

$$S_X(f) = \begin{cases} \frac{1}{(f+1)^{1.5}} + 0.5e^{-\frac{(f-15)^2}{10}} & 0 < f \leq 20 \\ (5/48) + 0.5e^{-\frac{(f-15)^2}{10}} & f > 20 \end{cases} \tag{5.36}$$

   Figure 5.3b shows the plot of this power spectral density. This category consists of two datasets. The first one consists of CPMG sequences of maximum order of 7 for the $x-$ and $y-$ directions. We take all possible combinations of orders along each direction. This leaves us with 49 possible configurations. We follow the same randomization scheme discussed before applied to the pulses along the $x-$ and $y-$ directions separately. We generate 100 examples per each configuration and then split into training and testing subsets. The second dataset is similar with the only difference that we do not randomize over the pulse power, we just randomize over the pulse positions.

3. Qubit without noise (i.e. a closed quantum system), and pulses on two or-



(a) Z-axis noise

(b) X-axis noise

Figure 5.3: Powers Spectral Density of the noise that was used to generate the datasets in categories 1 and 2.

thogonal directions.

The Hamiltonian takes the form

$$H = \frac{1}{2}\Omega\sigma_z + \frac{1}{2}f_x(t)\sigma_x + \frac{1}{2}f_y(t)\sigma_y \tag{5.37}$$

This category has only datasets as well which follow the same scheme of pulse configuration and randomization as the second category dataset. The only difference is that the absence of noise.

Cat.	**Name**	**Pulse Shape**	**Noise**	**Control**	**Train**	**Test**
1	CPMG_G_X_28	Gaussian	(z)	(x)	2100	700
1	CPMG_S_X_28	Square	(z)	(x)	2100	700
2	CPMG_G_XY_7	Gaussian	(x, z)	(x, y)	3625	1225
2	CPMG_G_XY_pi_7	Gaussian	(x, z)	(x, y)	3625	1225
3	CPMG_G_XY_7_nl	Gaussian	–	(x, y)	3625	1225
3	CPMG_G_XY_pi_7_nl	Gaussian	–	(x, y)	3625	1225

Table 5.2: The datasets generated by computer simulations of a noisy qubit for training and assessing the performance of the proposed machine learning algorithm. The datasets are categorized into 3 categories. The first category is for qubits with noise along $z-$axis, and control pulses along $x-$ axis. The second category is for qubits with noise along $z-$ and $x-$ axes, and control pulses along $x-$ and $y-$ axes. The final category is for noiseless qubits with pulses along $x-$ and $y-$ axes.

It is worth emphasizing here that since we are generating the datasets by simulation, we had to arbitrarily chose some noise models and pulse configurations. In a physical experiment however, we do not assume any noise models and just directly measure the different outcomes. Moreover, the pulse configurations should be chosen according to the capability of the available experimental setup.

Dataset	# Iterations	MSE Training	MSE Testing
CPMG_G_X_28	3000	9.86×10^{-5}	1.03×10^{-4}
CPMG_S_X_28	3000	1.05×10^{-4}	1.14×10^{-4}
CPMG_G_XY_7	3000	4.12×10^{-4}	4.36×10^{-4}
CPMG_G_XY_pi_7	3000	2.47×10^{-4}	2.38×10^{-4}
CPMG_G_XY_7_nl	3000	8.32×10^{-5}	8.31×10^{-5}
CPMG_G_XY_pi_7_nl	3000	1.50×10^{-6}	1.52×10^{-6}

Table 5.3: The average MSE evaluated for the various datasets at the end of the training process. The average is calculated over the mutually exclusive training and testing subsets. The results show that the model has "good fit" since both MSE's for each dataset are close.

5.4.2 Results

The proposed algorithm was trained on each of the different datasets to assess its performance in different situations. The number of iterations is chosen to be 3000. Table 5.3 summarizes the MSE evaluated at the end of the training stage for both training and testing examples. Figure 5.4 shows the history of the training procedure for each of the datasets. The plot shows the MSE evaluated after each iteration for both the training and testing examples. For the testing examples, the MSE evaluated is just recorded and does not contribute to the calculation of the gradients for updating the weights. Figure 5.5 shows a violin plot of the MSE compared across the different datasets; while Supplementary Figure 5.8 shows the boxplot. Supplementary Figures 5.9 to 5.14 show the square of the prediction errors for measurement outcome in the best case, average case, and worst case examples of the testing examples of each dataset.

5.4.3 Applications

5.4.3.1 Dynamical decoupling and quantum control

For the model trained on the single-axis Gaussian dataset "CPMG_G_X_28", we tested the possibility of using it perform some quantum control tasks. Particularly, we implemented a simple numerical optimization-based controller that aims to find the optimal set of signal parameters to achieve some target quantum gate G. We used the cosine similarity as an objective function, which is defined for two $d \times d$ matrices U and V as

$$\tilde{F}(U,V) = \left| \frac{\mathrm{tr}\left[U^\dagger V\right]}{\sqrt{\mathrm{tr}\left[U^\dagger U\right]\mathrm{tr}\left[V^\dagger V\right]}} \right|^2, \tag{5.38}$$

satisfying that $0 \leq \tilde{F}(U, V) \leq 1$. This is a generalized definition for fidelity that reduces to the standard definition of gate fidelity when U and V are unitary,

$$F(U, V) = \frac{1}{d^2} |\operatorname{tr}[U^\dagger V]|^2. \tag{5.39}$$

Ideally, we target four objectives listed as follows:

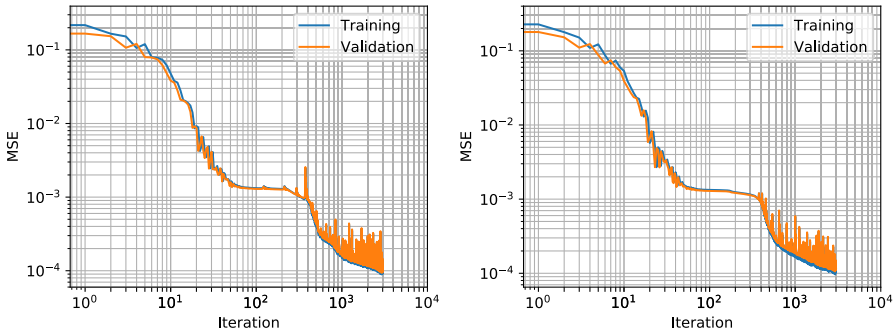
$$\tilde{F}(V_O, I) = 1, \quad \forall O \in \{X, Y, Z\} \tag{5.40}$$

$$\tilde{F}(U_{\text{ctrl}}, G) = 1, \tag{5.41}$$

where V_O and U_{ctrl} are estimated from the trained model. The first three conditions are equivalent to getting rid of the effects of noise, while the last one is equivalent to having achieve evolution described by quantum gate G. Practically, it is hard to completely remove the noise effects, so what we want to do is to find the set of optimal pulse parameters $\{\boldsymbol{\alpha}_n^*\}$ such that

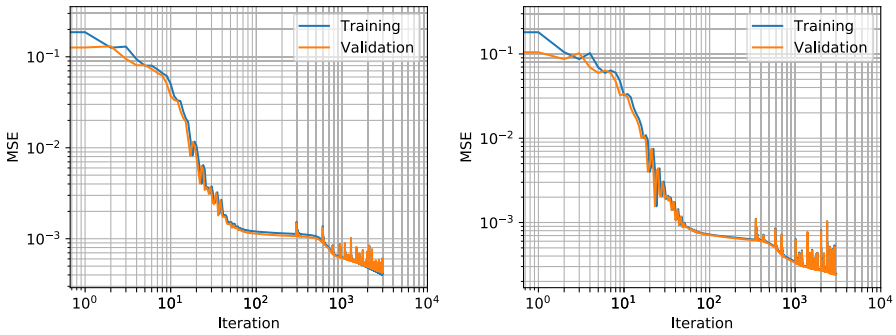
$$\begin{aligned} \boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} &(\tilde{F}(V_X[\boldsymbol{\alpha}], I) - 1)^2 \\ &+ (\tilde{F}(V_Y[\boldsymbol{\alpha}], I) - 1)^2 + (\tilde{F}(V_Z[\boldsymbol{\alpha}], I) - 1)^2 + (\tilde{F}(U_{\text{ctrl}}[\boldsymbol{\alpha}], G) - 1)^2. \end{aligned} \tag{5.42}$$

Then using this objective function we can numerically find the optimal pulse sequence. Utilizing this formulation allows us to treat the problem of dynamical decoupling exactly the same, with $G = I$. It is important to mention that this is just one method to do quantum control which might have some drawbacks because of its multi-objective nature. For instance, the optimization could result in one or more of the objectives having sufficient performance, while the others are not. An example of this case is where U_{ctrl} becomes so close to G, while the V_O operators are still far from the identity. This means that the overall evolution will not be equivalent to G. There are ways to overcome this problem. For example, we can optimize over the observables instead of the operators or optimize over the overall noisy unitary U. However, this is a separate issue, and we defer it to the future work of this chapter. We present these results as a proof of concept that it is possible to use the trained model as a part of a quantum control algorithm. We tested this idea to implement a set of universal quantum gates for a qubit. The resulting fidelities are shown in Table 5.4. The control pulses obtained from the optimization procedures are shown in Figure 5.6.



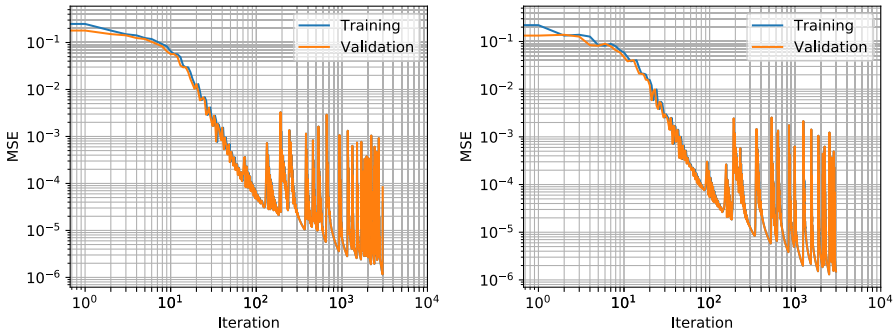
(a) CPMG_G_X_28

(b) CPMG_S_X_28



(c) CPMG_G_XY_7

(d) CPMG_G_XY_pi_7



(e) CPMG_G_XY_7_nl

(f) CPMG_G_XY_pi_7_nl

Figure 5.4: The average MSE evaluated for the training and testing examples versus the iteration number for the various datasets. The plots show a "good fit" model since both lines are close for each datasets, and they are decreasing as the number of iterations increase. The small fluctuations at the end of the plots result from the gradients of the loss function being noisy, and are exaggerated due to the logarithmic scale of the plot.

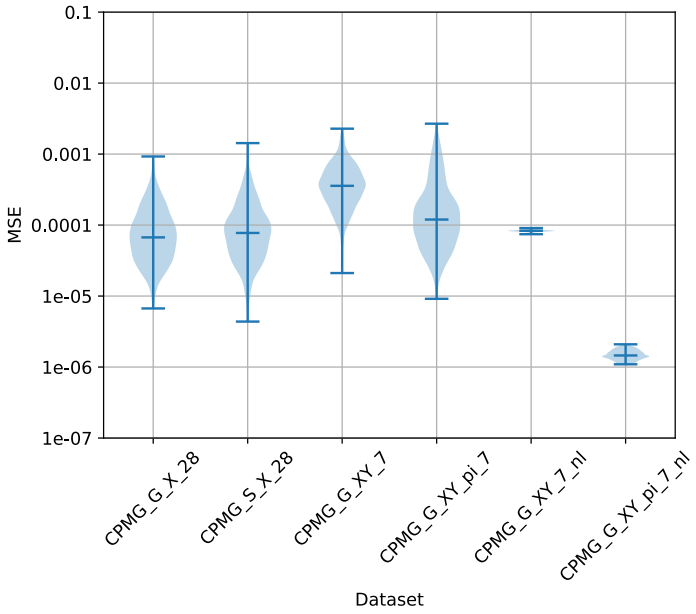


Figure 5.5: Comparison between the MSE of the various datasets at the end of the training phase, evaluated over the testing subsets only. The plot depicts an empirical kernel density estimate of data distribution, which is referred to as a Violin plot. The bottom horizontal line represents the minimum value, the middle line represents the median, and the top line represents the maximum. The plot shows how the data is "concentrated" with respect to its order statistics.

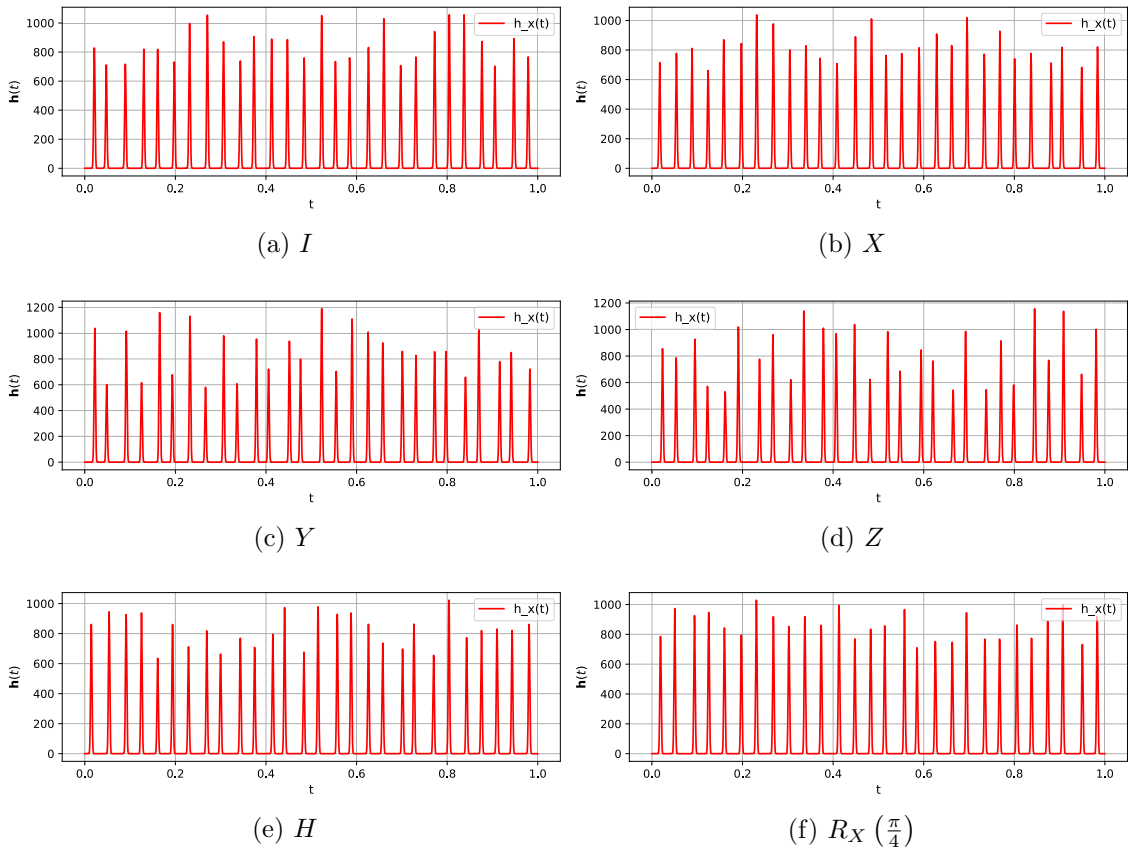
5.4.3.2 Quantum noise spectroscopy

It is also possible to use the trained model to estimate the power spectral density of the noise using the standard Alvarez-Suter (AS) method [23]. In this case, we use the trained model to predict the coherence of the qubit (that is the expectation of the X observable for the $X+$ initial state $\mathbb{E}\{X(T)\}_{\rho=X+}$) for a set of CPMG sequences at the correct locations and powers. Then, from the predicted coherence we can find the power spectrum that theoretically produces these values. In order to do so we have to assume the noise is stationary and Gaussian. Here, we have trained a separate model with CPMG sequences up to order 50. Since the evolution time T is fixed, the higher the order of the sequence is, the higher the accuracy of the estimated spectrum would be specially at high frequencies. On the other hand, because the pulses still have finite width, there is a maximum we could apply during the evolution time and thus we can only probe the spectrum up to some frequency. Figure 5.7 shows the plot of the estimated PSD of the noise versus the theoretical one, as well as the coherences obtained from predictions of the model as well as the theoretical ones. We emphasize here that the point of presenting this work is to develop a method that is more general than the standard QNS method. However, we show in this application that we can still utilize the conventional methods combined with our proposed one. Also, in this experiment the focus was on showing the possibility

G	$1-\tilde{F}(V_X, I)$	$1-\tilde{F}(V_Y, I)$	$1-\tilde{F}(V_Z, I)$	$1-\tilde{F}(U_{\text{ctrl}}, G)$
I	1.43×10^{-4}	9.27×10^{-5}	5.87×10^{-5}	4.26×10^{-4}
X	2.64×10^{-5}	1.09×10^{-4}	1.57×10^{-5}	2.69×10^{-4}
Y	9.42×10^{-5}	3.82×10^{-5}	5.34×10^{-5}	9.27×10^{-4}
Z	2.78×10^{-5}	4.37×10^{-4}	1.23×10^{-5}	6.64×10^{-4}
H	3.14×10^{-5}	1.33×10^{-4}	5.88×10^{-8}	7.46×10^{-4}
$R_X\left(\frac{\pi}{4}\right)$	5.41×10^{-5}	1.87×10^{-4}	8.52×10^{-5}	7.23×10^{-5}

Table 5.4: The resulting "generalized" infidelity between the predicted V_X, V_Y, V_Z, and U_{ctrl} from the machine learning model trained on the "CPMG_G_X_28" dataset, and the corresponding targets (which would be I, I, I, and G respectively). The control pulses were obtained by numerical optimization of the cost function taken to be the average of the generalized infidelities of the four targets, utilizing the trained model.

of doing spectrum estimation. We did not use the trained models discussed in the previous section as they are limited to 28 pulses which prevents the probing of the spectrum using the (AS) method to high frequencies.

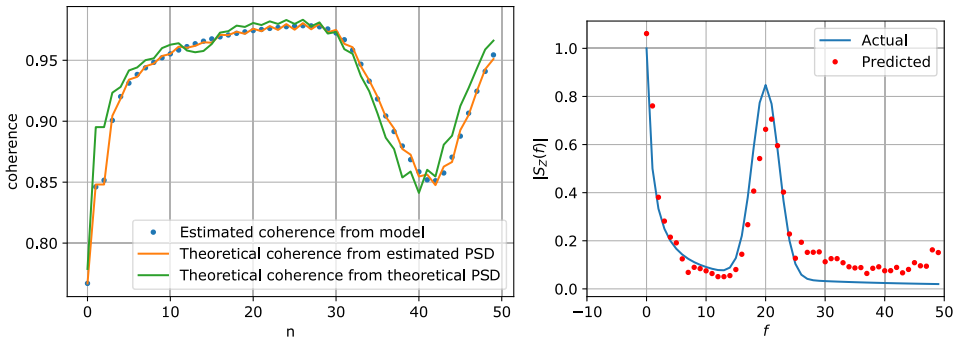


(a) I (b) X

(c) Y (d) Z

(e) H (f) $R_X\left(\frac{\pi}{4}\right)$

Figure 5.6: The control pulses to implement various quantum gates obtained by numerical optimization of the cost function taken to be the average of the generalized infidelity between the V_X, V_Y, V_Z and U_{ctrl} and the target operators I, I, I, and G respectively. The trained model of the "CPMG_G_X_28" dataset is utilized.



(a) Coherences (b) Power Spectrum

Figure 5.7: The estimated and actual theoretical coherence measurements and noise power spectrum using model trained on the "CPMG_G_X_28" dataset. 50 CPMG sequences with finite width pulses were used to measure the Pauli X observable with the positive Pauli X eigenstate as initial state. The power spectrum is reconstructed using Alvarez-Suter method using the predicted coherences from the trained model.

5.4.4 Discussion

The plots presented in the previous subsection are useful to assess the performance of the proposed method. First, we can see that for all datasets, the average MSE curve evaluated over the training examples versus iterations decreases on average with the number of iterations. This means that the structure is able to learn some abstract representation of the each of the V_O operators as function of the input pulses. For the testing subsets, we see that the MSE curves goes down following the training MSE curves. This indicates that the model is able to predict the observables for the training and testing examples, and thus the fitting is good. Second, The violin plot and boxplot of the MSE of the testing subsets show that there exists some minor outliers which we would expect anyway from a machine learning based algorithm. However, most of the points are concentrated at or below the median. This indicates that for most testing examples, the prediction was accurate (Note, the lower the MSE the better the prediction is). The best performing cases of the algorithm in terms of the spread of the testing MSE around the median was category 3 datasets. This is expected since in that category the quantum system is closed. Category 2 dataset had a slightly higher median compared to category 1 datasets. This can be explained due to the fact that in category 2, the qubit had multi-axis noise, which means the overall noise strength is higher than the category 1 datasets. This is a general observation that we would expect that the stronger the noise is, the harder it is to predict the outcomes.

Finally, if we look into the worst-case examples, we see that they are actually performing well in terms of accuracy for the different datasets. The overall conclusion from this analysis is that proposed model is able to learn how to predict

the measurement outcomes with high accuracy. The noisy multi-axis datasets had slightly less accuracy than the single-axis and the noiseless datasets, which might be worth investigation and is subject to the future work.

The results of the applications of the trained model are also very promising. The fidelities obtained for the different quantum gates are above 99% including the identity gate which equivalent to dynamical decoupling. This indicates that we can use numerical quantum control methods combined with our proposed one. We were also able to show the possibility of estimating the spectrum of the noise using the AS method. These results could be enhanced by including longer pulse sequences which requires increasing the overall time of evolution. Thus, the proposed framework is general enough to be used for different tasks in quantum control.

5.5 Conclusion

In this chapter, we presented a machine-learning based method for characterizing and predicting the dynamics of an open quantum system (and eventually controlling it) based on measurable information. We followed a graybox approach that allows us to estimate the V_O operators, which are generally difficult to calculate analytically without assumptions on noise and control signals. We showed the method is applicable to the general case where the noise is defined by a non-unital quantum map, but restricted the numerical experiments in this chapter on the case of classical noise (unital dynamics). The numerical results show good performance in terms of prediction accuracy of the measurement outcomes.

As discussed in Chapter 4, there are two main limitations of using ML models: scalability and generalizability. The scalability is definitely a major challenge for the proposed method because the full characterization requires models (and thus measurements) that go as the square of the dimension of the Hilbert space of the system. This directly results in the requirement of extremely large datasets for the training which might be difficult to prepare. This challenge, however, can be avoided in many applications. In quantum error correction applications, a partial characterization could be only be required. Thus, the model needs to predict few observables only rather than an informationally complete set. In quantum computing applications, the common practice is to optimize quantum gates for 1-qubit and 2-qubit systems rather than multi-qubit systems. The second challenge, which is generalization to arbitrary inputs, can also be avoided since the experimental setup for characterizing and operating the device is usually fixed. Therefore, under the same conditions of the experiment the models will be able to predict different inputs.

In the next chapter, we conclude the thesis and introduce the possible research directions extending the presented results.

5.6 Supplementary Figures

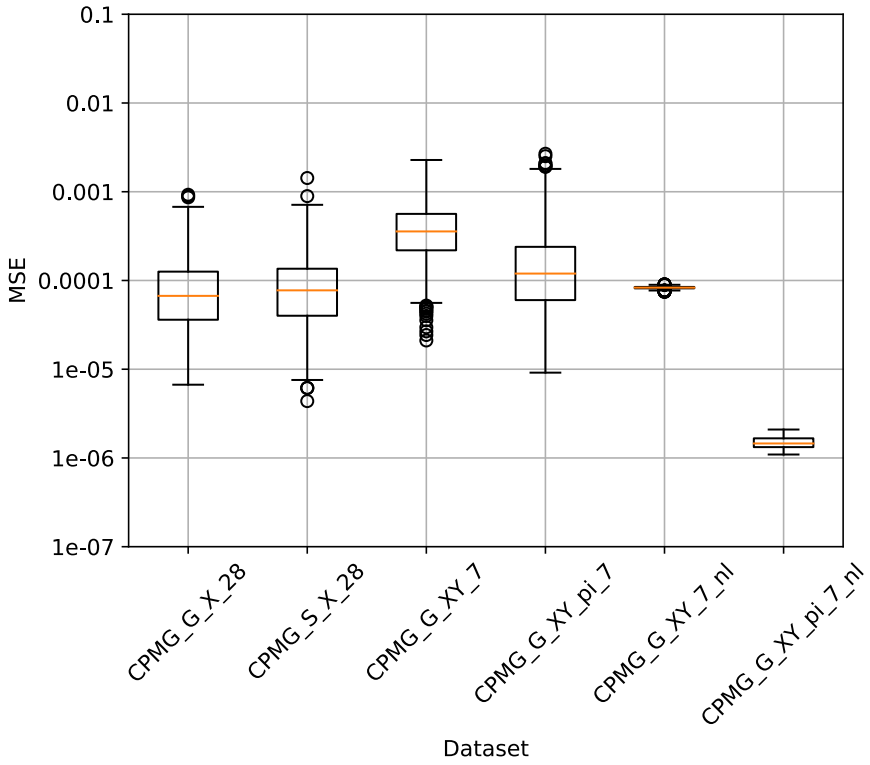
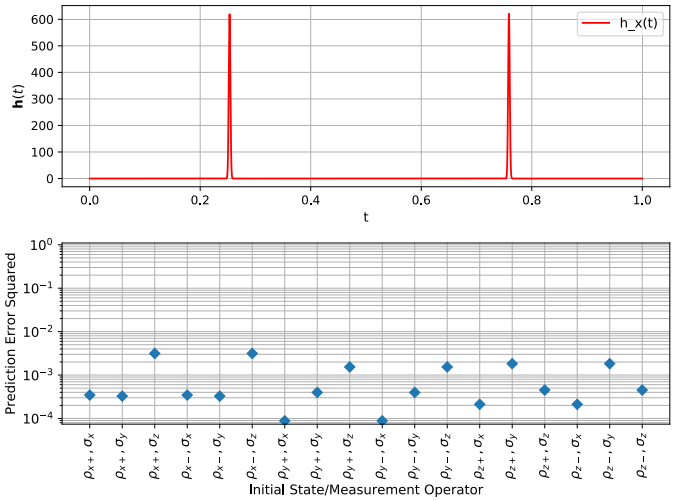
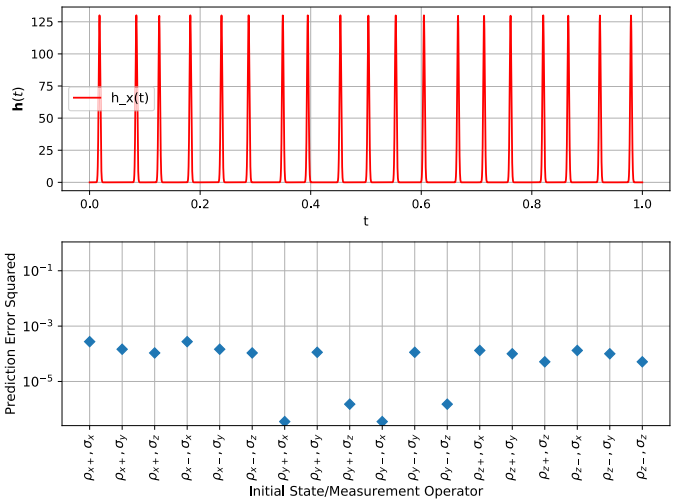


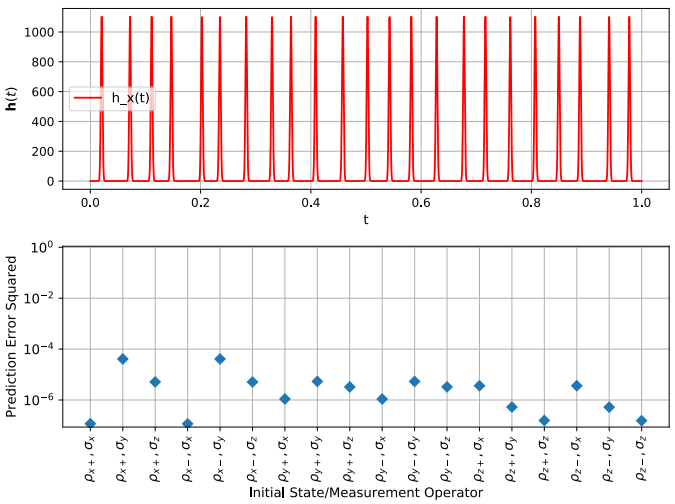
Figure 5.8: Comparison between the MSE of the various datasets at the end of the training phase, evaluated over the testing subsets only. The box is drawn between the first and third quartiles of the data, and thus the height represents the interquartile range. The middle horizontal line represents the median, the bottom and upper horizontal lines represent the minimum and maximum respectively. The maximum and minimum are defined to be the points that are at a distance of 1.5 $\times$ the interquartile range measured starting from the upper and lower quartiles respectively. Any point outside that range is referred to as an outlier and is represented as a small circle in the plot. The plot shows how the data is "concentrated" with respect to its order statistics, as well as the distribution of outliers.



(a) Worst case

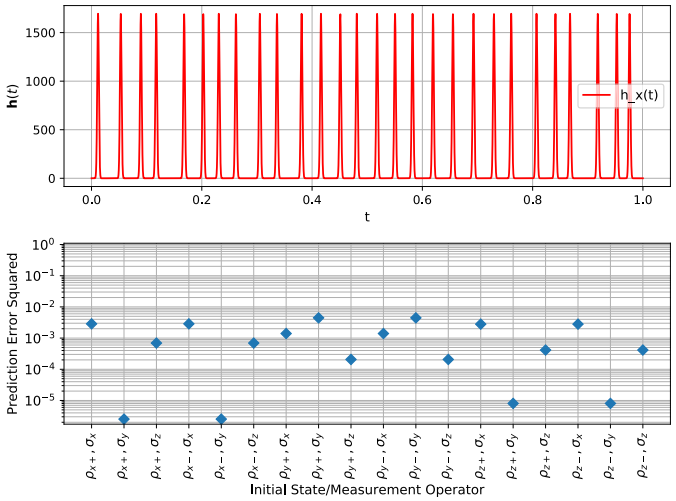


(b) Average Case

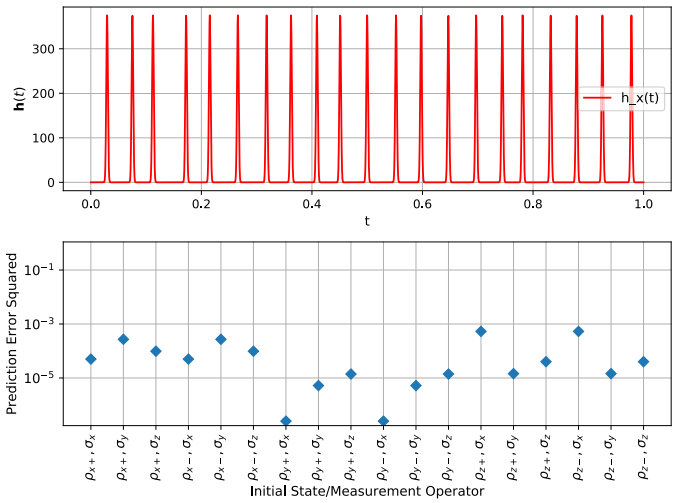


(c) Best Case

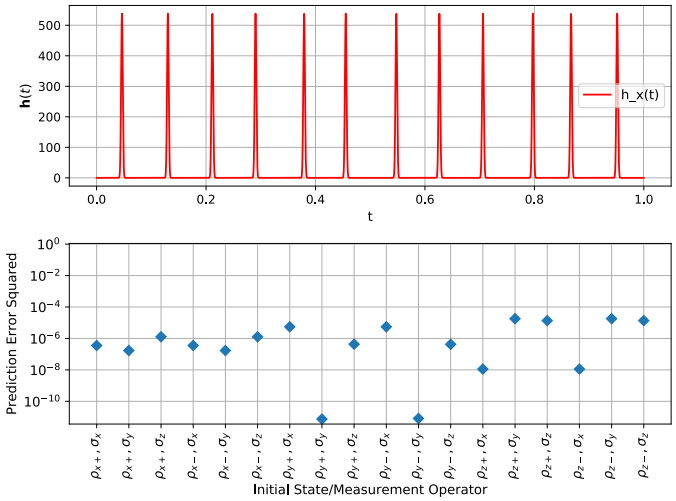
Figure 5.9: The worst, average, and best case examples for the CPMG_G_X_28 testing dataset.



(a) Worst case

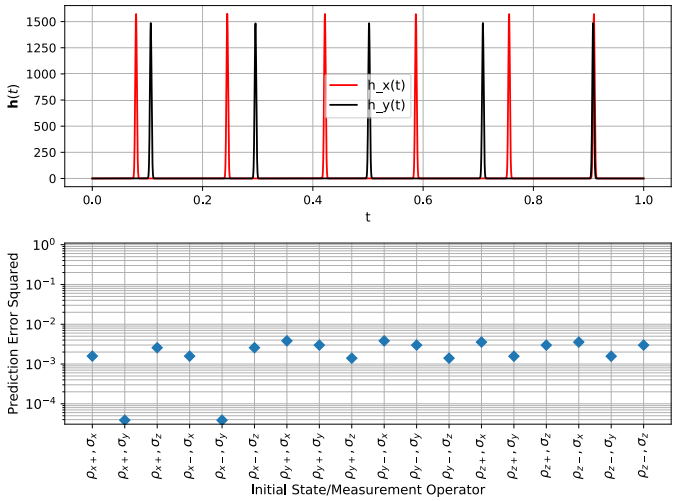


(b) Average Case

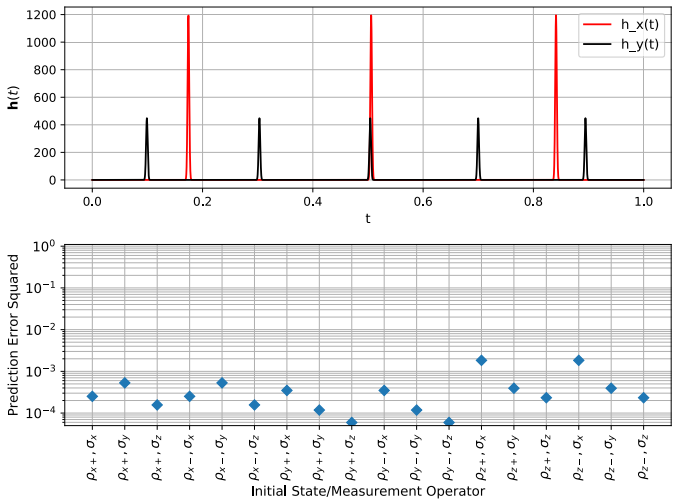


(c) Best Case

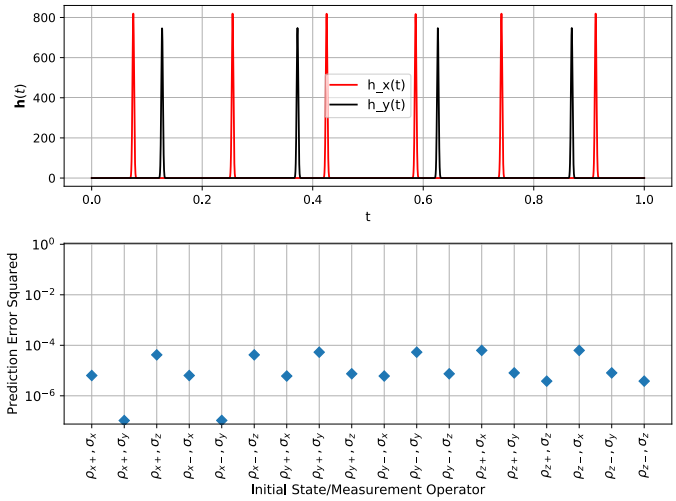
Figure 5.10: The worst, average, and best case examples for the CPMG_S_X_28 testing dataset.



(a) Worst case

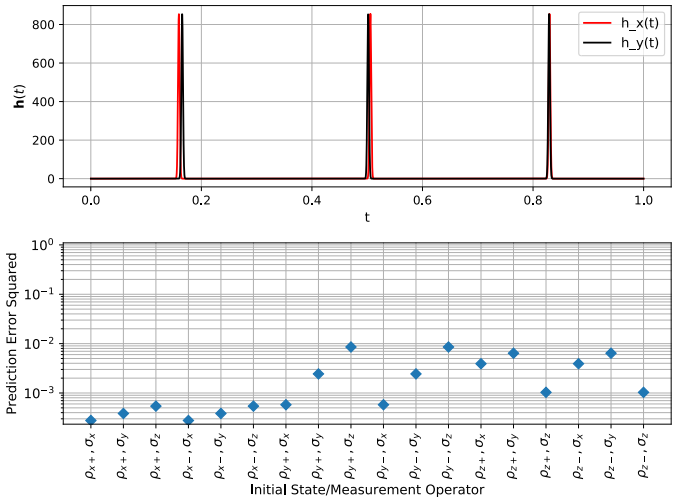


(b) Average Case

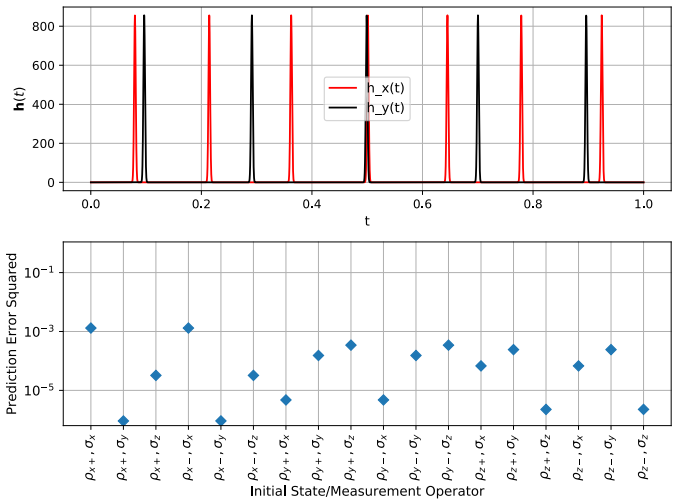


(c) Best Case

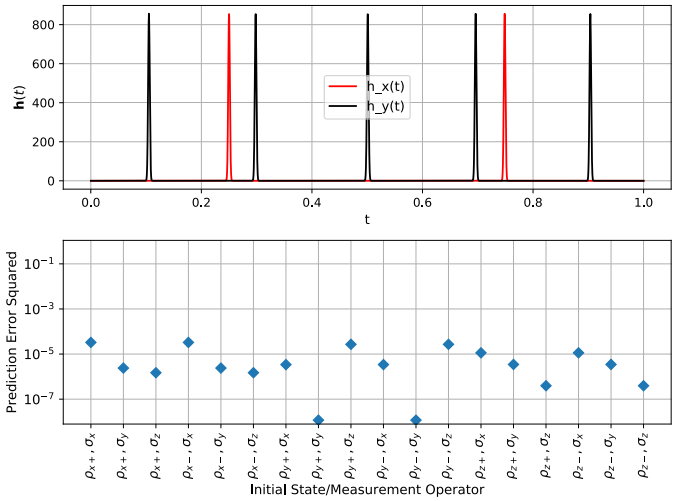
Figure 5.11: The worst, average, and best case examples for the CPMG_G_XY_7 testing dataset.



(a) Worst case

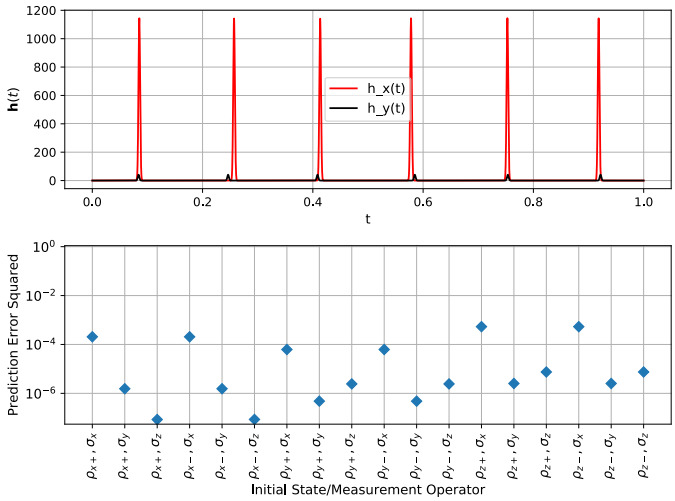


(b) Average Case

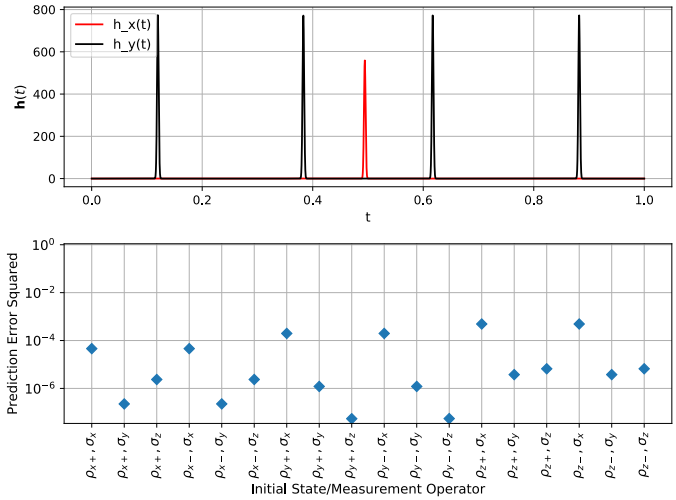


(c) Best Case

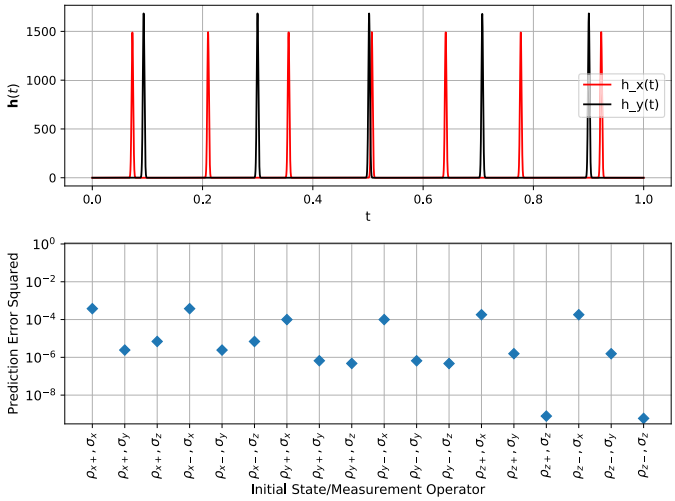
Figure 5.12: The worst, average, and best case examples for the CPMG_G_XY_pi_7 testing dataset.



(a) Worst case

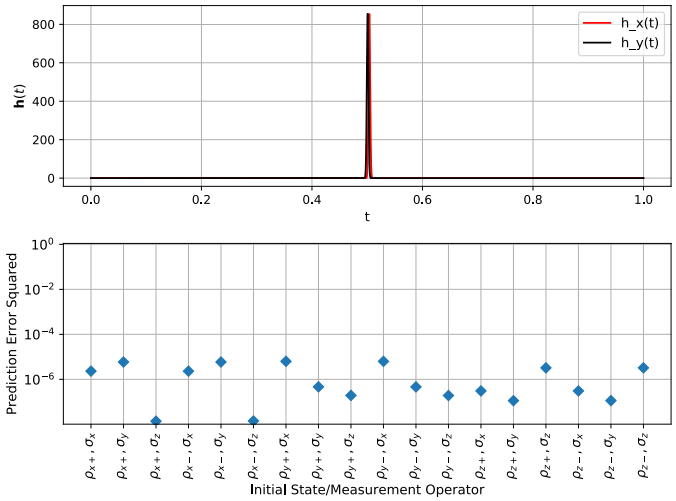


(b) Average Case

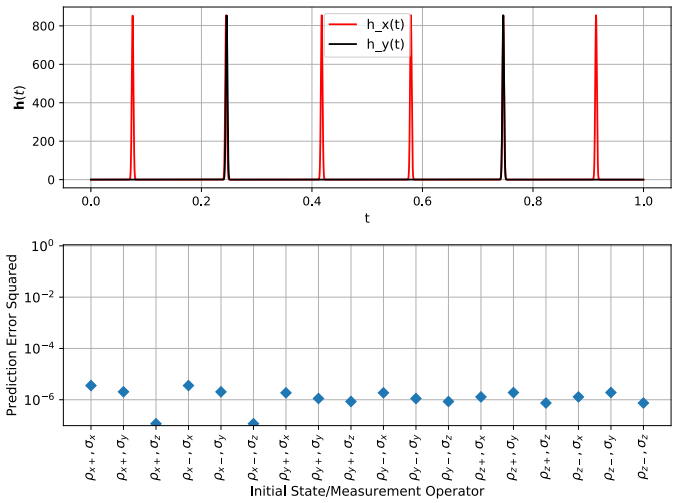


(c) Best Case

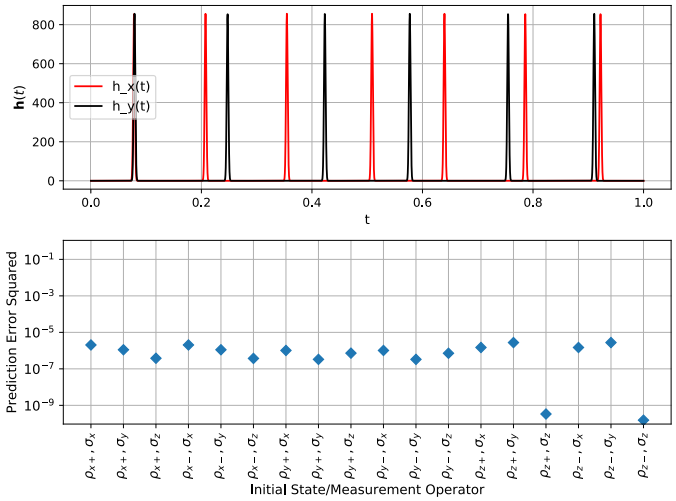
Figure 5.13: The worst, average, and best case examples for the CPMG_G_XY_7_nl testing dataset.



(a) Worst case



(b) Average Case



(c) Best Case

Figure 5.14: The worst, average, and best case examples for the CPMG_G_XY_pi_7_nl testing dataset.

Chapter 6

Conclusions and Future Work

In this thesis, we developed machine learning based frameworks for characterization and control of quantum systems. In Chapter 3, we presented an online quantum state estimation algorithm inspired by the matrix exponentiated gradient method. In Chapters 4 and 5, we developed novel graybox deep learning based models for quantum systems (for open and closed dynamics). The models take into consideration experimental constraints, and do not need assumptions on the control or the environment. We tried to present the problem in a way that that would facilitate for researchers from the machine learning community to explore and contribute to this field. We emphasized the importance of understanding the theoretical aspects so one can know the limitations and assumptions of various tools. This is the essence of using the graybox approach, as opposed to using only a blackbox that prevents the extraction of any physical insights from the trained models.

The presented methods in this thesis show great potential in solving problems in quantum characterization and control. Nonetheless, there are inevitable limitations that stem from the utilization of ML techniques as well as the difficulty of encoding quantum information classically. In Chapter 3, the main limitation is performing the state estimation on large-scale quantum systems. This is related however to the nature of the quantum tomography problem. In other words, the problem persists independent of the design of the algorithm. Thus, there is a need to develop fundamentally new frameworks to characterize large-scale systems beyond the standard tomography paradigm. In Chapters 4 and 5, the limitations arise from the use of ML techniques which can be summarized as follows. The first problem is scalability. For large-scale quantum systems, preparing a training dataset will be a challenging task. The second problem is generalizability. The trained ML models will be able to generalize only for examples that are sampled from the same distribution from which the training examples were sampled. This means models work only for the same experimental situations. If the hardware conditions change (such as changing some devices, or changing the control pulse characteristics), then a new dataset has

to be constructed and the model have to be re-trained. Therefore, a proper life-cycle management and fine-tuning needs to be maintained for the models to remain effective. In conclusion, the presented methods in this thesis are suited more for characterization and control of small-scale quantum devices, which is the dominant scale of current technology. One interesting approach that could be explored in the future is optimizing small-scale devices, and then use them directly to either control or become the building blocks of large-scale systems.

On the theoretical side, there are many extensions to the presented work. In Chapter 3, we considered only fixed measurements to do the state estimation. Considering adaptive measurements could further improve the performance, and proving convergence in that case would be interesting to explore. Additionally, proving convergence for the projected-gradient decent method as another online estimation algorithm is worth considering. In Chapter 4, it would be interesting to study controllability of the chip under model constraints such as limited control voltage to know the set of gates that are possible to implement. In Chapter 5, there are many interesting points to explore. The first point is studying the controllability of open quantum systems using the effective observable dynamics formalism, and also under constraints of control pulses. Another related direction is utilizing geometric techniques to study different problems of quantum control of open quantum systems.

On the numerical side, it would be interesting to explore other possible machine learning techniques in the classical literature and investigate their applicability in the quantum setting. In Chapter 4, fidelity could be used as cost function to do the training rather than the MSE, and study whether it would yield better results. The blackboxes introduced in Chapters 4 and 5 are not fully optimized. They were designed to prove the concept but there is a room for improvement in terms of accuracy and generalization. Particularly in Chapter 5, the blackboxes can be further optimized to increase the accuracy specially for the noisy multi-axis datasets. There are other architectures that one could exploit such as Convolutional Neural Networks. Moreover, one could make use of existing results in machine learning that deals with incomplete training data [119–122]. These could be leveraged to reduce the number of required experiments which would be useful particularly for higher dimensional systems.

On the experimental side, it would be interesting to test the ideas presented in this thesis while embedded in a real experiment. The methods would require fine-tuning according to a particular experimental situation, but nonetheless they are general enough for any quantum realization technology.

On the applications side there are many ideas to explore. In Chapter 3, the proposed method could be extended to quantum process tomography. It is also suitable for quantum state tracking, where a slowly time-varying quantum state

(due to experimental drifts for example) is tracked through the measurements. The utilization of the graybox approach in Chapters 4 and 5, could be extended to modelling of continuously-monitored systems, systems with feedback control, and continuous-variable systems. Another interesting application is modelling the data-spectator architecture of quantum systems where some redundant qubits are used to monitor the noise affecting the qubits that are used for the actual quantum computations. Finally, there is noise detection, where there exist a discrete set of possible noise models affecting a quantum system, and the problem is how to find optimal control pulses and measurements as well as a classifier to decide which noise model is affecting the system.

Bibliography

[1] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[2] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019.

[3] S. Bravyi, D. Gosset, R. König, and M. Tomamichel, "Quantum advantage with noisy shallow circuits," *Nat. Phys.*, vol. 16, no. 10, pp. 1040–1045, Jul. 2020.

[4] J. Morris, F. A. Pollock, and K. Modi, "Non-Markovian memory in IBMQX4," *arXiv:1902.07980*, 2019.

[5] K. Tsuda, G. Rätsch, and M. K. Warmuth, "Matrix exponentiated gradient updates for on-line learning and Bregman projection," *J. Mach. Learn. Res.*, vol. 6, no. Jun, pp. 995–1018, 2005.

[6] A. Globerson, T. Y. Koo, X. Carreras, and M. Collins, "Exponentiated gradient algorithms for log-linear structured prediction," in *Proc. ICML '07*. ACM Press, 2007, pp. 305–312.

[7] M. Abadi *et al.*, "TensorFlow: large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[8] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io

[9] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[11] F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, "Non-Markovian quantum processes: complete framework and efficient characterization," *Phys. Rev. A*, vol. 97, p. 012127, Jan. 2018.

[12] O. Oreshkov, F. Costa, and Č. Brukner, "Quantum correlations with no causal order," *Nat. Commun.*, vol. 3, no. 1, Jan. 2012.

[13] G. Chiribella, G. M. D'Ariano, and P. Perinotti, "Theoretical framework for quantum networks," *Phys. Rev. A*, vol. 80, p. 022339, Aug. 2009.

[14] F. Costa and S. Shrapnel, "Quantum causal modelling," *New J. Phys.*, vol. 18, no. 6, p. 063032, Jun. 2016.

[15] H. Y. Carr and E. M. Purcell, "Effects of diffusion on free precession in nuclear magnetic resonance experiments," *Phys. Rev.*, vol. 94, no. 3, pp. 630–638, May 1954.

[16] L. Viola and S. Lloyd, "Dynamical suppression of decoherence in two-state quantum systems," *Phys. Rev. A*, vol. 58, pp. 2733–2744, Oct. 1998.

[17] Ł. Cywiński, R. M. Lutchyn, C. P. Nave, and S. Das Sarma, "How to enhance dephasing time in superconducting qubits," *Phys. Rev. B*, vol. 77, p. 174509, May 2008.

[18] G. S. Uhrig, "Keeping a quantum bit alive by optimized π-pulse sequences," *Phys. Rev. Lett.*, vol. 98, p. 100504, Mar. 2007.

[19] K. Khodjasteh and D. A. Lidar, "Fault-tolerant quantum dynamical decoupling," *Phys. Rev. Lett.*, vol. 95, p. 180501, Oct. 2005.

[20] J. Řeháček, Z. Hradil, E. Knill, and A. I. Lvovsky, "Diluted maximum-likelihood algorithm for quantum tomography," *Phys. Rev. A*, vol. 75, no. 4, p. 042108, Apr. 2007.

[21] B. Qi, Z. Hou, L. Li, D. Dong, G. Xiang, and G. Guo, "Quantum state tomography via linear regression estimation," *Sci. Rep.*, vol. 3, Dec. 2013.

[22] J. A. Smolin, J. M. Gambetta, and G. Smith, "Efficient method for computing the maximum-likelihood quantum state from measurements with additive gaussian noise," *Phys. Rev. Lett.*, vol. 108, no. 7, p. 070502, Feb. 2012.

[23] G. A. Álvarez and D. Suter, "Measuring the spectrum of colored noise by dynamical decoupling," *Phys. Rev. Lett.*, vol. 107, p. 230501, Nov. 2011.

[24] N. Wiebe, C. Granade, C. Ferrie, and D. G. Cory, "Hamiltonian learning and certification using quantum resources," *Phys. Rev. Lett.*, vol. 112, no. 19, p. 190501, May 2014.

[25] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory, "Robust online Hamiltonian learning," *New J. Phys.*, vol. 14, no. 10, p. 103013, Oct. 2012.

[26] J. Wang *et al.*, "Experimental quantum Hamiltonian learning," *Nat. Phys.*, vol. 13, no. 6, p. 551, Mar. 2017.

[27] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, "Quantum Hamiltonian learning using imperfect quantum resources," *Phys. Rev. A*, vol. 89, no. 4, p. 042314, Apr. 2014.

[28] R. Blume-Kohout, J. K. Gamble, E. Nielsen, J. Mizrahi, J. D. Sterk, and P. Maunz, "Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit," *arXiv:1310.4492*, 2013.

[29] D. Greenbaum, "Introduction to quantum gate set tomography," *arXiv:1509.02921*, 2015.

[30] K. Jacobs and D. A. Steck, "A straightforward introduction to continuous quantum measurement," *Contemp. Phys.*, vol. 47, no. 5, pp. 279–303, Sep. 2006.

[31] J. A. Gross, C. M. Caves, G. J. Milburn, and J. Combes, "Qubit models of weak continuous measurements: markovian conditional and open-system dynamics," *Quantum Sci. Technol.*, vol. 3, no. 2, p. 024005, Feb. 2018.

[32] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms," *J. Magn. Reson.*, vol. 172, no. 2, pp. 296–305, Feb. 2005.

[33] P. de Fouquieres, S. Schirmer, S. Glaser, and I. Kuprov, "Second order gradient ascent pulse engineering," *J. Magn. Reson.*, vol. 212, no. 2, pp. 412–417, Oct. 2011.

[34] G. Ciaramella, A. Borzì, G. Dirr, and D. Wachsmuth, "Newton methods for the optimal control of closed quantum spin systems," *SICOMP*, vol. 37, no. 1, pp. A319–A346, Jan. 2015.

[35] T. Caneva, T. Calarco, and S. Montangero, "Chopped random-basis quantum optimization," *Phys. Rev. A*, vol. 84, no. 2, p. 022326, Aug. 2011.

[36] M. Abdelhafez, D. I. Schuster, and J. Koch, "Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation," *Phys. Rev. A*, vol. 99, p. 052327, May 2019.

[37] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, "Speedup for quantum optimal control from automatic differentiation based on graphics processing units," *Phys. Rev. A*, vol. 95, p. 042318, Apr. 2017.

[38] H. Haas, D. Puzzuoli, F. Zhang, and D. G. Cory, "Engineering effective Hamiltonians," *New J. Phys.*, vol. 21, no. 10, p. 103011, Oct. 2019.

[39] M. Paris and J. Řeháček, Eds., *Quantum state estimation.* Springer Berlin Heidelberg, 2004.

[40] Z. Hradil, "Quantum-state estimation," *Phys. Rev. A*, vol. 55, no. 3, pp. R1561–R1564, Mar. 1997.

[41] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing," *Phys. Rev. Lett.*, vol. 105, no. 15, p. 150401, Oct. 2010.

[42] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert, "Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators," *New J. Phys.*, vol. 14, no. 9, p. 095022, 2012.

[43] C. Granade, J. Combes, and D. G. Cory, "Practical bayesian tomography," *New J. Phys.*, vol. 18, no. 3, p. 033024, Mar. 2016.

[44] E. Bolduc, G. C. Knee, E. M. Gauger, and J. Leach, "Projected gradient descent algorithms for quantum state tomography," *npj Quantum Inf.*, vol. 3, no. 1, p. 44, Oct. 2017.

[45] J. Shang, Z. Zhang, and H. K. Ng, "Superfast maximum-likelihood reconstruction for quantum tomography," *Phys. Rev. A*, vol. 95, no. 6, p. 062336, Jun. 2017.

[46] Y.-H. Li and V. Cevher, "A general convergence result for the exponentiated gradient method," *arXiv:1705.09628*, 2017.

[47] S. Aaronson, "The learnability of quantum states," in *Proc. Math. Phys. Eng. Sci.*, vol. 463, no. 2088. The Royal Society, 2007, pp. 3089–3114.

[48] ——, "Shadow tomography of quantum states," in *Proc. STOC 2020.* ACM Press, 2018, pp. 325–338.

[49] S. Aaronson, X. Chen, E. Hazan, S. Kale, and A. Nayak, "Online learning of quantum states," *J. Stat. Mech: Theory Exp.*, vol. 2019, no. 12, p. 124019, Dec. 2019.

[50] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, "Sample-optimal tomography of quantum states," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5628–5641, Sep. 2017.

[51] H. Umegaki, "Conditional expectation in an operator algebra," *Kodai Math. Sem. Rep.*, vol. 14, pp. 59–85, 1962.

[52] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth, "A comparison of new and old algorithms for a mixture estimation problem," *Mach. Learn.*, vol. 27, no. 1, pp. 97–119, Apr. 1997.

[53] S. Golden, "Lower bounds for the Helmholtz function," *Phys. Rev.*, vol. 137, no. 4B, pp. B1127–B1128, Feb. 1965.

[54] C. J. Thompson, "Inequality with applications in statistical mechanics," *J. Math. Phys.*, vol. 6, no. 11, pp. 1812–1813, Nov. 1965.

[55] M. Adam, "Applications of unitary k-designs in quantum information processing," Master's thesis, Masarykova univerzita, Fakulta informatiky, 2013.

[56] R. A. Low, "Pseudo-randomness and learning in quantum computation," *arXiv:1006.5227*, 2010.

[57] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *Proc. STOC '99*. ACM Press, 1999, pp. 507–516.

[58] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numer. Math.*, vol. 108, no. 1, pp. 59–91, Oct. 2007.

[59] N. Friis *et al.*, "Observation of entangled states of a fully controlled 20-qubit system," *Phys. Rev. X*, vol. 8, no. 2, p. 021012, Apr. 2018.

[60] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Phys. Rev. A*, vol. 48, pp. 1687–1690, Aug. 1993.

[61] A. Peruzzo *et al.*, "Quantum walks of correlated photons," *Science*, vol. 329, no. 5998, pp. 1500–1503, Sep. 2010.

[62] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Trans. Signal Inf. Process.*, vol. 3, 2014.

[63] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, "Automated search for new quantum experiments," *Phys. Rev. Lett.*, vol. 116, no. 9, p. 090405, Mar. 2016.

[64] A. A. Melnikov *et al.*, "Active learning machine learns to create new quantum experiments," *PNAS*, vol. 115, no. 6, pp. 1221–1226, Jan. 2018.

[65] L. O'Driscoll, R. Nichols, and P. Knott, "A hybrid machine learning algorithm for designing quantum experiments," *Quantum Mach. Intell.*, pp. 1–11, Mar. 2018.

[66] H. Zhou, Y. Zhao, X. Wang, D. Gao, J. Dong, and X. Zhang, "Self-learning photonic signal processor with an optical neural network chip," *arXiv:1902.07318*, 2019.

[67] Y. Ming, C.-T. Lin, S. D. Bartlett, and W.-W. Zhang, "Quantum topology identification with deep neural networks and quantum walks," *npj Comput. Mater*, vol. 5, no. 1, pp. 1–7, Aug. 2019.

[68] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, "Universal quantum control through deep reinforcement learning," *npj Quantum Inf.*, vol. 5, no. 1, Apr. 2019.

[69] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, "Reinforcement Learning in Different Phases of Quantum Control," *Phys. Rev. X*, vol. 8, p. 031086, Sep. 2018.

[70] M. Ostaszewski, J. Miszczak, L. Banchi, and P. Sadowski, "Approximation of quantum control correction scheme using deep neural networks," *Quantum Inf. Process.*, vol. 18, no. 5, p. 126, 2019.

[71] A. Youssry, C. Ferrie, and M. Tomamichel, "Efficient online quantum state estimation using a matrix-exponentiated gradient method," *New J. Phys.*, vol. 21, no. 3, p. 033006, Mar. 2019.

[72] M. Niemiec, "Error correction in quantum cryptography based on artificial neural networks," *Quantum Inf. Process.*, vol. 18, no. 6, p. 174, Apr. 2019.

[73] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. Beenakker, "Machine-learning-assisted correction of correlated qubit errors in a topological code," *Quantum*, vol. 2, p. 48, Jan. 2018.

[74] D. Lennon *et al.*, "Efficiently measuring a quantum device using machine learning," *npj Quantum Inf.*, vol. 5, no. 1, pp. 1–8, Sep. 2019.

[75] S. Yamada and M. Minakata, "DC drift phenomena in $LiNbO_3$ optical waveguide devices," *Jpn. J. Appl. Phys.*, vol. 20, no. 4, p. 733, Apr. 1981.

[76] C. Gee, G. Thurmond, H. Blauvelt, and H. Yen, "Minimizing dc drift in $LiNbO_3$ waveguide devices," *Appl. Phys. Lett.*, vol. 47, no. 3, pp. 211–213, Aug. 1985.

[77] H. Nagata and J. Ichikawa, "Progress and problems in reliability of Ti: $LiNbO_3$ optical intensity modulators," *Opt. Eng.*, vol. 34, no. 11, pp. 3284–3294, Nov. 1995.

[78] Y. Bromberg, Y. Lahini, R. Morandotti, and Y. Silberberg, "Quantum and classical correlations in waveguide lattices," *Phys. Rev. Lett.*, vol. 102, no. 25, p. 253904, Jun. 2009.

[79] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[80] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, Oct. 2012.

[81] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, "Tunable, flexible, and efficient optimization of control pulses for practical qubits," *Phys. Rev. Lett.*, vol. 120, p. 150401, Apr. 2018.

[82] J. Bylander *et al.*, "Noise spectroscopy through dynamical decoupling with a superconducting flux qubit," *Nat. Phys.*, vol. 7, no. 7, pp. 565–570, May 2011.

[83] F. K. Malinowski *et al.*, "Spectrum of the nuclear environment for GaAs spin qubits," *Phys. Rev. Lett.*, vol. 118, p. 177702, Apr. 2017.

[84] K. W. Chan *et al.*, "Assessment of a silicon quantum dot spin qubit environment via noise spectroscopy," *Phys. Rev. Applied*, vol. 10, p. 044017, Oct. 2018.

[85] L. M. Norris, G. A. Paz-Silva, and L. Viola, "Qubit noise spectroscopy for non-gaussian dephasing environments," *Phys. Rev. Lett.*, vol. 116, p. 150503, Apr. 2016.

[86] G. Ramon, "Trispectrum reconstruction of non-gaussian noise," *Phys. Rev. B*, vol. 100, p. 161302, Oct. 2019.

[87] Y. Sung *et al.*, "Non-gaussian noise spectroscopy with a superconducting qubit sensor," *Nat. Commun.*, vol. 10, no. 1, p. 3715, 2019.

[88] V. M. Frey *et al.*, "Application of optimal band-limited control protocols to quantum noise sensing," *Nat. Commun.*, vol. 8, no. 1, p. 2189, Dec. 2017.

[89] G. A. Paz-Silva, L. M. Norris, and L. Viola, "Multiqubit spectroscopy of gaussian quantum noise," *Phys. Rev. A*, vol. 95, p. 022121, Feb. 2017.

[90] G. A. Paz-Silva, L. M. Norris, F. Beaudoin, and L. Viola, "Extending comb-based spectral estimation to multiaxis quantum noise," *Phys. Rev. A*, vol. 100, p. 042334, Oct. 2019.

[91] L. M. Norris, D. Lucarelli, V. M. Frey, S. Mavadia, M. J. Biercuk, and L. Viola, "Optimally band-limited spectroscopy of control noise using a qubit sensor," *Phys. Rev. A*, vol. 98, p. 032315, Sep. 2018.

[92] Ł. Cywiński, "Dynamical-decoupling noise spectroscopy at an optimal working point of a qubit," *Phys. Rev. A*, vol. 90, p. 042307, Oct. 2014.

[93] V. Frey, L. M. Norris, L. Viola, and M. J. Biercuk, "Simultaneous spectral estimation of dephasing and amplitude noise on a qubit sensor via optimally band-limited control," *Phys. Rev. Applied*, vol. 14, p. 024021, Aug. 2020.

[94] P. Szańkowski, G. Ramon, J. Krzywda, D. Kwiatkowski, and Ł. Cywiński, "Environmental noise spectroscopy with qubits subjected to dynamical decoupling," *J. Phys. Condens. Matter*, vol. 29, no. 33, p. 333001, Aug. 2017.

[95] J. Krzywda, P. Szańkowski, and u. Cywiński, "The dynamical-decoupling-based spatiotemporal noise spectroscopy," *New J. Phys.*, vol. 21, no. 4, p. 043034, Apr. 2019.

[96] J. H. Cole and L. C. L. Hollenberg, "Scanning quantum decoherence microscopy," *Nanotechnology*, vol. 20, no. 49, p. 495401, Nov. 2009.

[97] T. Yuge, S. Sasaki, and Y. Hirayama, "Measurement of the noise spectrum using a multiple-pulse sequence," *Phys. Rev. Lett.*, vol. 107, p. 170504, Oct. 2011.

[98] C. L. Degen, F. Reinhard, and P. Cappellaro, "Quantum sensing," *Rev. Mod. Phys.*, vol. 89, p. 035002, Jul. 2017.

[99] M. M. Müller, S. Gherardini, and F. Caruso, "Noise-robust quantum sensing via optimal multi-probe spectroscopy," *Sci. Rep.*, vol. 8, no. 1, Sep. 2018.

[100] C. Benedetti, F. Salari Sehdaran, M. H. Zandi, and M. G. A. Paris, "Quantum probes for the cutoff frequency of Ohmic environments," *Phys. Rev. A*, vol. 97, p. 012126, Jan. 2018.

[101] G. Gordon, G. Kurizki, and D. A. Lidar, "Optimal dynamical decoherence control of a qubit," *Phys. Rev. Lett.*, vol. 101, p. 010403, Jul. 2008.

[102] M. J. Biercuk, A. C. Doherty, and H. Uys, "Dynamical decoupling sequence construction as a filter-design problem," *J. Phys. B*, vol. 44, no. 15, p. 154002, Jul. 2011.

[103] T. Chalermpusitarak, B. Tonekaboni, Y. Wang, L. M. Norris, L. Viola, and G. A. Paz-Silva, "Frame-based filter-function formalism for quantum characterization and control," *arXiv:2008.13216*, 2020.

[104] A. Youssry, R. J. Chapman, A. Peruzzo, C. Ferrie, and M. Tomamichel, "Modeling and control of a reconfigurable photonic circuit using deep learning," *Quantum Sci. Technol.*, vol. 5, no. 2, p. 025001, Jan. 2020.

[105] H. Chen, M. Vasmer, N. P. Breuckmann, and E. Grant, "Machine learning logical gates for quantum error correction," *arXiv:1912.10063*, 2019.

[106] J. Bausch and F. Leditzky, "Quantum codes from neural networks," *New J. Phys.*, vol. 22, no. 2, p. 023005, Feb. 2020.

[107] C. Ferrie, C. Granade, G. Paz-Silva, and H. M. Wiseman, "Bayesian quantum noise spectroscopy," *New J. Phys.*, vol. 20, no. 12, p. 123005, Dec. 2018.

[108] L. Viola, E. Knill, and S. Lloyd, "Dynamical decoupling of open quantum systems," *Phys. Rev. Lett.*, vol. 82, pp. 2417–2421, Mar. 1999.

[109] K. Khodjasteh and D. A. Lidar, "Fault-tolerant quantum dynamical decoupling," *Phys. Rev. Lett.*, vol. 95, p. 180501, Oct. 2005.

[110] K. Khodjasteh and L. Viola, "Dynamically error-corrected gates for universal quantum computation," *Phys. Rev. Lett.*, vol. 102, p. 080501, Feb. 2009.

[111] M. H. Levitt and R. Freeman, "Compensation for pulse imperfections in NMR spin-echo experiments," *J. Magn. Reson.*, vol. 43, no. 1, pp. 65 – 80, Apr. 1981.

[112] C. Kabytayev, T. J. Green, K. Khodjasteh, M. J. Biercuk, L. Viola, and K. R. Brown, "Robustness of composite pulses to time-dependent control noise," *Phys. Rev. A*, vol. 90, p. 012316, Jul. 2014.

[113] H.-V. Do *et al.*, "Experimental proof of quantum Zeno-assisted noise sensing," *New J. Phys.*, vol. 21, no. 11, p. 113056, Nov. 2019.

[114] S. Kotler, N. Akerman, Y. Glickman, and R. Ozeri, "Nonlinear single-spin spectrum analyzer," *Phys. Rev. Lett.*, vol. 110, p. 110503, Mar. 2013.

[115] Y. Wang *et al.*, "Single-qubit quantum memory exceeding ten-minute coherence time," *Nat. Photonics*, vol. 11, no. 10, pp. 646–650, Sep. 2017.

[116] D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty- I," *Bell Syst. Tech. J.*, vol. 40, no. 1, pp. 43–63, Jan. 1961.

[117] G. A. Paz-Silva and L. Viola, "General transfer-function approach to noise filtering in open-loop quantum control," *Phys. Rev. Lett.*, vol. 113, p. 250501, Dec. 2014.

[118] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv:1412.6980*, 2015.

[119] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, no. 5-6, pp. 519–533, May 2003.

[120] E. Acuña and C. Rodriguez, "The Treatment of Missing Values and its Effect on Classifier Accuracy," in *Classification, Clustering, and Data Mining Applications*, D. Banks, F. R. McMorris, P. Arabie, and W. Gaul, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 639–647.

[121] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural. Comput. Appl.*, vol. 19, no. 2, pp. 263–282, Sep. 2010.

[122] V. C. Raykar *et al.*, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, Aug. 2010.