UNIVERSITY OF CALGARY

Evolutionary Algorithm for Adaptive Quantum-Channel Control

by

Pantita Palittapongarnpim

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN PHYSICS AND ASTRONOMY

CALGARY, ALBERTA
JANUARY, 2019

Abstract

The key to successful implementations of quantum technologies is quantum control, whose aim is to steer quantum dynamics such that the desired outcome is achieved. Quantum control techniques rely on models of the quantum dynamics to generate control policies that attain the control targets. In a practical situation, the dynamic model may not match the dynamic in the implementation, and this mismatch can lead to reduced performance or even a failed control procedure.

Data-driven control has been proposed as an alternative to model-based control design. In this approach, measurement outcomes from the system are used to generate a policy, which enables robust control without the need for a noise model. The potential for data-driven quantum control has been demonstrated in the problem of quantum-enhanced adaptive phase estimation. However, the performance and robustness of data-driven policies have never been compared with performance and robustness of model-based control techniques.

In this thesis, we aim to determine the advantages and disadvantages of model-based and data-driven policy generation using a simulated quantum-enhanced adaptive phase estimation as an example of a quantum control task. In the process, we explore the connection between an adaptive quantum-enhanced metrological procedure to a decision-making process, which is an alternative model of the dynamic during the control task. We also devise a robust search algorithm based on an evolutionary algorithm that is ignorant of the properties of the phase noise but is still able to deliver quantum-enhanced precision. We then compare the performances of feedback control policies designed using Bayesian inference, which is a model-based technique, to policies generated using this robust evolutionary algorithm on their performance in both noisy and noiseless interferometers. We also assess the resources used in generating and implementing a control policy and use the complexities of the time and space costs as parts of selecting a practical control procedure.

Preface

The research conducted for this thesis has been published in four papers and one preprint manuscript. The content from these publications is used in appropriate sections of this thesis verbatim or with some modifications. Below is the list of publications from this work [1, 2, 3, 4, 5]:

1. Pantita Palittapongarnpim, Peter Wittek and Barry C. Sanders. Controlling adaptive quantum phase estimation with scalable reinforcement learning. In *Proc. 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016)*: 327–332, Apr 2016.

2. Pantita Palittapongarnpim, Peter Wittek and Barry C. Sanders. Single-shot adaptive measurement for quantum-enhanced metrology. In *Proc. of SPIE Quantum Communications and Quantum Imaging XIV*, **9980**:99800H, Sep 2016. DOI:`http://doi.org/10.1117/12.2237355`. arXiv:1608.06238.

3. Pantita Palittapongarnpim, Peter Wittek, Ehsan Zahedinejad, Shakib Vedaie and Barry C. Sanders. Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics, *Neurocomputing* **268**: 116–126, Apr 2017. DOI:`http://doi.org/10.1016/j.neucom.2016.12.087`. arXiv:1607.03428

4. Pantita Palittapongarnpim, Peter Wittek and Barry C. Sanders. Robustness of learning-assisted adaptive quantum-enhanced metrology in the presence of noise. In *Proc. 2017*

IEEE International Conference on Systems, Man and Cybernetics (2017 SMC): 294–299, Dec 2017. DOI:http://doi.org/10.1109/SMC.2017.8122618.

5. Pantita Palittapongranpim and Barry C. Sanders. Robustness of Adaptive Quantum-Enhanced Phase Estimation. Manuscript submitted to *Physical Review A*, 2018. arXiv:1809.05525

The materials taken from these papers and the modifications made for this thesis are listed below:

- Chapter 1 contains a figure that is reproduced from Ref. [3] and some sentence verbatim from Ref. [5].
- Chapter 3 contains sentences verbatim but not marked from the Background section on quantum-enhanced metrology and adaptive phase estimation in Ref [5].
- The majority of Chapter 4 of this thesis has been published in [2] and is reproduced here verbatim except for the modifications listed below.
  - Materials referring to reinforcement learning are changed to policy search or an evolutionary algorithm to reflect new insights.
  - Sec. 4.1 is taken mostly verbatim from Sec. 2 in Ref. [2].
  - Sec. 4.1.4 is rewritten from Sec. 4 of Ref. [2].
  - Sec. 4.2 contains new material written for this thesis.
  - The figures are reproduced with reference to the figures in the original paper.
  - Sec. 4.4 is modified to include a discussion of the non-entangled state, taken verbatim from Sec. 5 of Ref. [2]
  - Sec. 4.5 is modified from Sec. 7 of Ref. [2].
- Chapter 5 comes from the author's manuscript version of Ref. [3] and is reproduced here verbatim except for the modification below.

  - Materials referring to reinforcement learning are changed to policy search or an evolutionary algorithm.
  - Sec. 4.3 is added for the thesis.
  - The figure is reproduced with reference to the figure in the original paper.
  - Sec. 5.7 is modified in part from Sec. 6 of Ref. [1].

- Chapter 6 is taken from manuscript Ref. [4] and Ref. [5] with modifications as listed below.
  - Sec. 6.1 is taken mostly verbatim from Ref. [5] with a modification to refer to preliminary test in Ref. [4].
  - Sec. 6.2.1 is taken verbatim from Ref. [5] except for the phase estimation algorithm, which is taken from Ref. [4].
  - Sec. 6.2.2 is modified to include additional materials about the statistical criteria.
  - Subsec. 6.3.1 is taken mostly verbatim from Ref. [4].
  - Sec. 6.3.4 is modified from Sec. III.C to include detail of the cost calculations.
  - Sec. 6.4.1 is taken from Ref. [4] with the figure and table reproduced with reference to the ones in the original paper.
  - Sec. 6.4.1 is taken from Ref. [5] with the figure and table reproduced with reference to the manuscript.

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Barry C. Sanders, whose support and advice have been most critical in this endeavor. My PhD has been a massive reeducation that could not have gone well without his guidance and patience. Also, I would like to thank him for all the opportunities he has given or directed to me over the years. This degree wouldn't have been as rich without those experience.

I also would like to thank Dr. Peter Wittek, who has been my collaborator for most of my degree and is instrumental to getting the search algorithm and my understanding of it in shape. Through him, I have been introduced to practices in programming that I would not have come across myself or would not have undertaken for the lack of knowledge and understanding.

Many ideas developed over the course of my PhD research have been born out of discussions with my group mates past and present, especially Ehsan Zahedinejad, Seyed Shakib Vadiae, Eduardo Paez, and Archismita Dalal. Many thanks for their time and hard work.

I would also like to thank people that I had met at INRIA – Paris during my brief summer research internship on quantum control: Dr. Mazyar Mirrahimi who was my supervisor for the project, Dr. Pierre Rouchon who showed me techniques in quantum control theory, Dr. Zaki Leghtas for the lab tour and insights into experiments on superconducting qubits, my most helpful and gracious office mates especially Joachim Cohen, Nicolas Didier, and Shantanu Mundhada. The advice meant a great deal for someone new to the field.

The computational work in this thesis has been supported by Westgrid (www.westgrid.ca)

through Compute Canada/Calcul Canada (www.computecanada.ca). The summer internship at INRIA–Paris was partially supported by MITACS through the Globalink Research Award. I also would like the acknowledge the financial support of AITF and NSERC who partially funded my PhD.

Lastly, I would like to thank my family for the morale support from half the world away. Closer are my friends here in Calgary, whose names are too long to be listed individually here without taking a ridiculous amount of spaces. Thank you for the kindness and inclusiveness even though I have not shown up as much as I should. Their laughter and company will be greatly missed when I move away.

Table of Contents

List of Figures and Illustrations

List of Tables

List of Symbols, Abbreviations and Nomenclature

Symbol or abbreviation	Definition
QEM	Quantum-enhanced metrology
AQEM	Adaptive quantum-enhanced metrology
QEAPE	Quantum-enhanced adaptive phase estimation
MDP	Markov decision process
POMDP	Partially observable Markov decision process
CPU	Central processing unit
GPU	Graphics processing unit
DE	Differential evolution
PSO	Particle swarm optimization
CRLB	Cramér-Rao lower bound
SQL	Standard quantum limit
HL	Heisenberg limit
C	Controller
P	Control plant
u	Control signal
y	Feedback signal
r	Control goal, reference signal
SC	Supervisory controller
z	Signal to supervisory controller from the plant
AIC	Akaike information criterion
$\mathrm{AIC_c}$	Corrected Akaike information criterion
N	Number of particles
O	The Bachmann–Landau notation
ϕ	Unknown parameter
ϕ_0	Mode of a noise distribution
$\widetilde{\phi}$, $\widetilde{\phi}$	Estimate
$\bar{\widetilde{\phi}}$	Mean estimate
$\bar{\theta}$	Mean direction (in directional statistics)
$\Delta\widetilde{\phi}$	Uncertainty
Φ	Controllable parameter
$\hat{\Phi}$	Estimator
x	Measurement outcome

$\boldsymbol{x}$	A string of measurement outcome
V_{H}	Holevo variance
b	(Function) bias, (variable) number of parameters
F	Fisher information
F_Q	Quantum Fisher information
S	Sharpness, fitness function
$\bar{S}$	Mean sharpness
$\mathscr{H}$	Hilbert space
$\mathcal{I}$	Quantum channel
$\mathcal{K}$	Kraus operator for quantum channel
X	Positive-operator-valued measure
U	Unitary operator
C	Combined measurement operator
$\mathcal{L}$	Symmetric logarithm derivative
ρ	Quantum state
P, p	Probability distribution
ϱ	Control policy
$\boldsymbol{\Delta}$	Phase adjustment vector
V	Variance of a noise distribution
γ	Skewness of a noise distribution
η	Loss rate
σ	Standard deviation
v	Number of data
$\wp$	Power-law scaling
N_{p}	population size
C_r	Crossover rate in DE
$\mathcal{F}$	Scaling factor in DE
F'	F-value
R^2	Coefficient of determination
$\overline{R^2}$	adjusted coefficient of determination

Chapter 1

Introduction

1.1 Motivation

Quantum technology exploits quantum resources to enhance the performance of applications such as computation [6], communication [7], and metrology [8]. At the heart of accomplishing these goals is quantum control, whose aim is to steer the quantum system to the desired outcome given the time and resource constraints and the available controllable degrees of freedom [9]. This outcome may be to steer an initial state to the desired final state, which appears in applications such as control of molecular processes [10], or to achieve the desired quantum channel, such as in quantum gate design [11] and adaptive quantum-enhanced metrology [12]. In both cases, the control task can be formulated as approximating the desired quantum dynamic.

The crucial element in achieving the control goal is the design of a controller or more specifically a control policy [13], which is a set of rules used by the controller to compute the time-dependent control signal sent to the actuators in a plant where the quantum process is taking place [14]. The policy has been largely devised using a model of the quantum dynamics and a technique in optimal control theory [15, 16]. Having a model of the quantum dynamics provides insight into the working of the plant and ensures that the policies being generated

are stable and robust. However, these promises hinge on the assumption that the model matches the dynamics of the plant, an assumption that cannot be confirmed in practical applications [17]. In fact, a mismatch is expected in an implementation of a control system as some processes in the plant may be unknown [17]. In the case where the mismatch is minor, the control performance decreases but may still be able to achieve the control goal. Major issue arises when the mismatch leads to instability in the control procedure; i.e., when the control output no longer responds to the signal from the controller [18].

A model of the plant dynamics may be devised to approximate the input and output signal to and from the plant [19], but close approximation of a complex dynamic can lead to a complicated model that is intractable for the control design process [20, 21]. This consideration is particularly important for quantum information technology as quantum supremacy lies in the ability to manipulate large number of particles coherently [22, 23]. Increasing the number of particles, the number of degrees of freedom, the accessible energy levels, and noise and loss all adds to the complexity of the dynamic model.

Data-driven control is an approach in control theory that proposes forgoing the use of a dynamic model and instead devising the control policies directly from data [18]. This idea has been explored under different names since 1970s, such as, unfalsified control [24] and learning control [25], each referring to a particular technique to deal with the unavailability of dynamic models. Learning control formalizes the way to use an optimizer for automatic design of a control policy by augmenting the traditional control loop with another loop containing a supervisory controller that uses data to generate policies [25].

Although the underlying argument for data-driven control is to not rely on models, techniques have been proposed that combine the use of data and models [26]. Model-free methods can be viewed as one extreme of the spectrum where no dynamic model is used or constructed by the supervisory controller. Only the knowledge of the controllable degrees of freedom and the measurement that should be performed are available. Because model-free methods have to interact with the plant numerous times in order to find a policy that

successfully delivers the control target, a combination of model and measurement data has been proposed in order to reduce the number of samples needed [18, 27]. The model-based data-driven techniques has been used to attain a robust control procedure when the dynamic of the plant is not fully known or is time-varying under unknown changes [26, 28].

Policy search is a data-driven technique that uses trial and error to generate a policy. In this technique, the policy is given a task-appropriate mathematical form, and the performance is optimized by searching the space of the policy's parameters [29, 30]. A policy-search method is called model-free if the optimization procedure does not learn a model of the dynamic before updating the policy parameters [30]. One way to devise a model-free policy-search algorithm is by employing a black-box optimization algorithm [31], which samples the performance of the policy either by trying the policy on an experimental setup or a simulation of the setup. Either case, the controller and supervisory controller are ignorant of the origin of the data.

Evolutionary algorithms are black-box optimization algorithms whose procedures are based on biological evolution [32]. An evolutionary algorithm is able to locate an area in a search space where the globally optimal solution resides, but the heuristic nature of the iterative search procedure means that the solution cannot be proven to be optimal. However, the algorithm can be made to search until a policy that delivers the desired performance is found, and this type of algorithms has been used successfully in policy search [29, 33].

1.2 Research problems

Evolutionary algorithms have been used to generate policies in quantum control problems, such as quantum-enhanced adaptive phase estimation [34, 35] and quantum gate design [36, 37, 38]. These computational works prove in principle that the data-driven approach can converge on policies that deliver target performance in simulations. However, the performances of these policies have not been compared to ones from a model-based control

technique, and the robustness to the model mismatch has never been tested. As such, the promised advantages of the data-driven control have never been shown in the context of quantum control.

In this thesis, we aim to identify the advantages and disadvantages of using model-free data-driven control using an example of quantum control, namely the quantum-enhanced adaptive phase estimation. The phase estimation setup is simulated on high-performance computers, controlled either by policies generated using a policy-search algorithm [34, 35] or by a model-based feedback designed using Bayes's theorem [39, 40]. Both methods are ignorant of the noise models included in the simulation of the interferometric phase estimation scheme, which is included to emulate the mismatch between the dynamic model used in designing the control policy and the dynamic of the quantum plant. Using this simulation, we test the robustness of these policies and assess the scaling in the computational resources in designing and implementing the control procedure.

1.3 Adaptive quantum-enhanced metrology

The example of quantum channel control that we use in this work is the quantum-enhanced adaptive phase estimation (QEAPE), which is a problem in quantum-enhanced metrology (QEM). QEM aims to improve the precision of a measurement scheme by exploiting quantum resources. In this field, the improvement is measured in terms of the power-law scaling of the imprecision with the number of particles N. Classical techniques, such as using lasers, leads to the lower bound that is known as the standard quantum limit (SQL). An optimal use of the quantum resource results in the lower bound that scales faster than SQL known as the Heisenberg limit (HL). The improvement in precision is of interest to high-precision technologies, such as, the gravitational wave detection [41, 42, 43], atomic clocks [44, 45], and magnetometry [22, 46], where the power tolerance is limited by the hardware and thus must seek precision gain by other means.

Adaptive quantum-enhanced metrology (AQEM) is an approach that attains quantum-enhanced precision by approximating optimal measurement on the quantum state [12]. The discrete-time version of this approach split the N-particle resource into bundles, each interacting with the system containing an unknown parameter ϕ in sequence. After each interaction, the bundle is measured and the measurement outcomes used to adjust a controllable parameter Φ inside the system before the next bundle is injected. In principle, each adjustment would lead to a closer estimate of ϕ than the last as additional information about the unknown parameter is gained from the outcome. However, this assumption hinges upon the feedback control performing a correct adjustment. Therefore, achieving the quantum-enhanced precision in the adaptive approach is a matter of designing a successful feedback procedure.

Phase estimation is widely studied in QEM as its mathematics underlies many applications in the field [47, 48], and hence is an estimation problem whose SQL and HL are well-established and can be used as benchmarks for the adaptive scheme. For a passive interferometric phase estimation, the SQL asymptotically scales as $1/\sqrt{N}$ and HL as $1/N$ [49], which means in the optimal case the precision can be quadratically improved. In this work, we consider the case of two-mode Mach-Zehnder interferometer (Fig. 1.1), where an unknown phase shift $\phi \in [0, 2\pi)$ is in one arm and a controllable phase shifter $\Phi \in [0, 2\pi)$ in another. The N entangled particles are split into packets of single particles and only one particle enters the interferometer at any given time step.

There are at least two methods to design feedback control for QEAPE. One uses Bayesian inference to update the prior of ϕ, and the control procedure selects the value of Φ that best minimizes the imprecision calculated from the prior [40]. We refer to this method as the Bayesian feedback throughout this thesis. The second is the policy-search method that uses black-box optimization to find the policy's parameters that successfully delivers quantum-enhanced precision. Because the latter method does not use a model of the state evolution to devise the policy, the method is considered model-free [50]. Therefore, comparing these

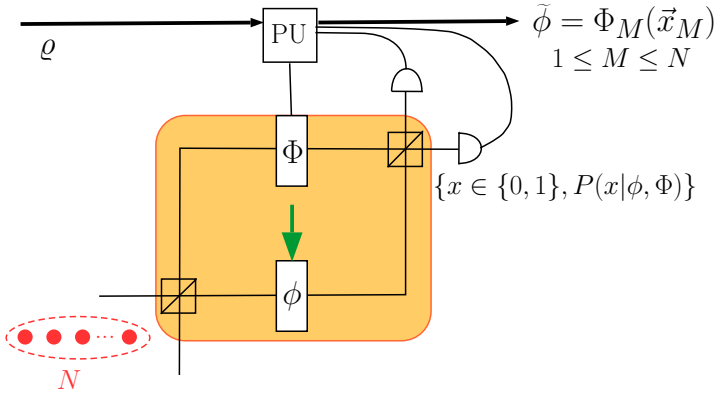


Figure 1.1: An adaptive phase estimation scheme. An N-photon entangled state is separated into single-particle bundles. A particle is injected into a Mach-Zehnder interferometer, representing a two-mode interferometer. Contained in this interferometer is an unknown phase shifter ϕ and a controllable phase shifter Φ, whose value at the m^{th} measurement is Φ_m. The path x_m in which the photon exits is detected by single-photon detectors connected to a processing unit (PU). The PU uses the information to adjust $\Phi_m \to \Phi_{m+1}$ according to policy ϱ. After all the photons are measured, the estimate $\widetilde{\phi}$ is inferred from Φ_M, allowing for loss in photons which leads to $1 \le M \le N$. Hence, the estimate is a function of the history of measurement outcomes $\boldsymbol{x}_M = (x_1 x_2 \cdots x_M)$. [Reproduced from Ref. [3], Fig. 1]

policies can provide some insight into the potential advantages and disadvantages of model-based control and data-driven control for quantum technology.

1.4 Objectives and approaches

The research in this thesis is divided into three parts. The first part of the research aims to understand the factors that contribute to the imprecision of an adaptive measurement and explore the approaches to generate successful policies. By formulating the discrete-time AQEM as a decision-making process and analyzing its characteristics, we are able to explain the use of a policy-search algorithm and determine the methods to make the algorithm noise-resistant and scalable up to a hundred particle in the second part of the work. Lastly, we devise the criteria for comparing AQEM policies based on the ideal performance, robustness, and the time and space complexity of the control implementation. We then use these criteria to compare the Bayesian feedback to the policies from the policy-search algorithm and discuss

the advantages and disadvantages of the two methods.

1.4.1 AQEM as a decision-making process

A decision-making process has been used to represent a discrete-time control problem as can be used as an alternative to the first-principle model used in quantum control [51, 52]. In this part of the research, we formulate a general discrete-time AQEM scheme as a decision-making process and analyze the characteristic of a case study, namely the adaptive phase estimation, to determine the possible policy generation methods.

We also use the decision-making model to derive the Cramér-Rao lower bound [53] (CRLB) for the discrete-time adaptive measurement procedure. Previously, the imprecision lower bound in QEM is defined using the quantum version of CRLB [8, 54, 55] that is formulated for the collective, non-adaptive measurement. Although this provides a bound for ideal adaptive measurement, it does not provide insight into how an input state and a policy contributes to an adaptive measurement being able to achieve quantum-enhanced precision. We explore the role of input state in determining imprecision scaling by using an evolutionary algorithm to attain the minimum imprecision for both the entangled and unentangled state.

1.4.2 Scalable noise-resistant policy-search algorithm

A problem with previous policy-search algorithms for adaptive phase estimation is that it fails to generate policies that deliver power-law scaling at high N. A noise-resistant algorithm based on particle swarm optimization (PSO) is able to generate successful policies up to $N = 45$ [34], whereas the one based on differential evolution (DE) can do so to $N = 93$ but without noise [35]. The second objective is then to create a scalable, noise-resistant algorithm that can deliver a successful policy even in a high-dimensional search space.

To improve the scalability of the algorithm, we introduce a criteria for accepting policies based on its performance instead of constraining the number of generations. Doing so enables

the optimization algorithm to run for as long as needed to produce a successful policy. However, due to the limitation in CPU time, we demonstrate the algorithm up to $N = 100$ which already shows a consistent power-law relationship between the imprecision and N. To make the algorithm noise-resistant, we average out the noise from the fitness values, which measures the performance of the policy [34].

1.4.3 Comparing AQEM policies

In the last part of this research, we devise the criteria for comparing policies in quantum-enhanced adaptive phase estimation focusing on qualities that are of interest in quantum technology. Although performance criteria exist for control engineering, they are focused on the time- and frequency-dependent characteristics of the error signal [56]. These criteria may not be useful in quantum control as the quantum system may not be continuously monitored to avoid the effect of measurements on state evolution. Using the AQEM as a test problem, we propose criteria that are based on the performance in ideal condition, the robustness against noise with unknown properties, and the resources used in generating and implementing a control policy. We then compare the Bayesian feedback and DE-designed policies in QEAPE to show the advantages and disadvantages of both methods.

The noiseless interferometer is the condition that the Bayesian feedback is designed for, and so is the condition that the model matches the dynamics. The performance in this condition is therefore the best either method could achieve. Hence we can use this as a baseline to study how noise affects the model-based feedback control in contrast to policies from the model-free policy search.

The robustness test is executed by introducing phase noise to the simulation of QEAPE while the controller and the policy generating procedure are both ignorant of the properties of this noise. The scheme is determined to be robust if the phase estimation is able to break the SQL for four noise models, namely the normally-distributed noise [57], the random telegraph noise [58], the skew-normal noise [59], and the log-normal noise [60], all of which have the

periodicity of 2π. We also determine the level of noise, quantified by the variance of the noise distribution, that leads to an imprecision scaling equal to or worse than SQL for each of the noise models.

Lastly, we assess the resources used by the controller and policy-generating procedure in order to compare the efficiency of both designing and implementing the control policy. The policy generation is assessed based on the complexity of its time cost [61], which is evaluated by the scaling in the number of operations with the number of particles N. To determine which policy is superior, we compare the complexity in space and time cost for policy implementation [62]. As we seek an efficient procedure that beats the SQL, we choose a policy that requires the least resource to run, given that the target performance can be reached.

1.5 Summary of results

In this section, we briefly describe the results from the three objectives set out in Sec. 1.4. For the first objective, we formulate an AQEM scheme, more precisely the quantum-enhanced adaptive phase estimation, into a partially observable Markov decision process. We also determine that policy search, which updates the entire policy episodically, is suitable for generating a successful policy when the quantum dynamic is not known or trusted. Using the decision-making process, we are also able to derive a CRLB, although we cannot determine the scaling of the imprecision as a function of N due to its dependence on the policy. We utilize a DE algorithm to search for policies that minimize the imprecision for product states, which leads to a SQL for adaptive phase estimation, and sine states, which leads to an imprecision scaling that exceeds the SQL.

The DE algorithm is then made to be noise-resistant by averaging the performance of a policy multiple times before using the mean value for optimization. The algorithm uses two samples for each iteration of the DE algorithm and continues to add samples if the

same policy is selected for the next iteration. Although this number of samples is low and the averaging method leads to comparisons of mean values that have been computed from different sample sizes, the result shows that the mean values are sufficiently different that the DE algorithm is able to locate policies that break the SQL. Combining the noise-resistant DE with accept-reject criteria at high N enables us to deliver imprecision following a power-law relationship that surpasses the SQL up to 100 particles for low-level normally-distributed phase noise.

In comparing the quantum-enhanced adaptive phase estimation schemes, we find that both Bayesian-feedback and DE-designed policies are robust against unknown phase noise. Specifically, the Bayesian method yields imprecision that approaches HL and outperforms DE-designed policies for most noise models. This performance superiority is due to the Bayesian method effectively memorizing the measurement history through the controller's complete knowledge of the quantum state. Storing the entire model yields better imprecision scaling but incurs higher space and operational time costs compared to the DE-derived policy.

Chapter 2

Quantum Control and Evolutionary Algorithms

In this chapter, we review the techniques for devising quantum control policies following the approaches in control theory and engineering [9, 20]. These techniques begin with creating a model the quantum dynamics and then deriving or computing a policy such that the condition of optimality or robustness is met. These techniques have allowed manipulation of molecular structures [10], cooling [63, 64], and quantum gate design [11, 36]. However, a model can be expensive to construct, too complicated to be used efficiently in the designing process, or untrustworthy as observations of a quantum system are limited [20]. These problems are not confined to the field of quantum control as industrial plants can grow to be large and complicated. Data-driven control aims to bypass the use of a model and instead devise a control policy using the available data from the plant [18]. One technique that matches this description is policy search [30], which, as the name suggests, searches for the parameters of a pre-structured policy that can successfully deliver the control target. The search of the parameters can be done using an evolutionary algorithm [65] that treats the plant as a black-box and uses only the outputs to evaluate a policy's performance.

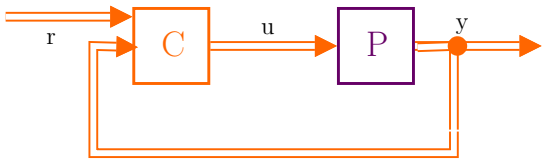


Figure 2.1: A block diagram of a control system. In a quantum control problem, the plant P obeys quantum mechanics, while the controller C and all information channels obey classical mechanics. The reference signal r sets the control goal for the controller, which produces an input u to the plant. The measurement inside the plant sends out signal y, which in closed-loop control is fed back to the controller to be used to determine the next input. In open-loop control, this signal might be monitored but does not influence the control signal.

2.1 Quantum control

In this section, we give an overview of the techniques that have been used to design quantum control procedures. We start with defining a control system and explain the types of control performed on quantum systems, namely open-loop and closed-loop control. We then describe the methods for creating dynamic models for each of these control procedures as the quantum measurement has to be taken into account. We end with reviewing the techniques for designing policies from these models.

2.1.1 Types of quantum control

In this subsection, we explain the components of a control system, as introduced in control engineering, and the difference between open-loop and closed-loop control. Quantum control can be viewed as a case when a part of the control system is governed by quantum mechanics. This assumption leads to a fundamental difference in control design of open-loop and closed-loop (or measurement-based) control on a quantum system due to quantum measurement.

Quantum control uses the same building blocks as a classical control system (Fig. 2.1) [66]. The quantum system is contained in a plant P, which also includes actuators for manipulating the quantum dynamics and measurement devices to observe the quantum system. The observation is sent out in the feedback signal y. The actuators receive an input signal u from a controller C, which holds a policy ϱ. The controller is also given a control goal r

(also known as a reference signal) that determines the desired performance of the control system. The controller and the communication channels in this quantum control scenario are assumed to be classical, and we do not consider the fully-quantum control system [67, 68] as we are interested in the use of classical controllers for the purpose of executing quantum control tasks.

Whereas the distinction between open loop and closed loop is determined by whether y is used by the controller in classical control engineering [69], the distinction in quantum control also includes whether the quantum system is measured during the control task. This difference between the quantum and classical case comes from the assumptions about classical control system, namely that y is a deterministic function of u without added noise [70] and that the measurements do not disturb the state of the plant [71]. These assumptions cannot be satisfied by quantum measurements whose outcome is always probabilistic with incomplete information about the quantum state [72, 73]. Therefore, the control design for a quantum plant with feedback control has to account for the measurement and its effect on the state, whereas the design for open-loop control can forgo this consideration until the end where the measurement is used to ascertain the success of a control policy.

The choice between open-loop and closed-loop control is determined by the application. Control of a chemical process, state transfer [20, 10], and quantum-gate design [11, 36] employ open-loop quantum control as measurements disrupt the intended evolution. On the other hand, cooling of a quantum system [74] and stabilization [75] use measurements to monitor and steer the quantum system. In AQEM, the measurement and feedback afford us the ability to approximate near-optimal measurement [76].

Knowing whether and how the measurement will be used is necessary for generating a quantum control policy as the standard procedure utilizes a model of the plant [71]. The procedure can be divided into two steps: creating a plant model and designing a control policy based on the model. An accurate model of a quantum control system must also specify the type of measurement as well as the type of control as this choice affects the

modelling of the state evolution, which we review in the following subsection.

2.1.2 Devising quantum control procedures

In this subsection, we review the techniques for generating quantum control policies. The design process assumes that the control engineer has at least a heuristic knowledge of the controllable degree of freedoms and measurements necessary to monitor the control procedure [19]. Provided that the control goal, the desired characteristics, and the performance measure have been decided, the first step is to create a model of the plant and its dynamics, which in this case follows the formalism of quantum mechanics. The model is then used to derive or compute the control policy subjected to constraints and specifications of the control problem.

Modeling of quantum systems

In control engineering, the model of the plant's dynamics is derived either from the physical laws (first-principle model) or by mathematically emulating the response of the plant [77]. The purpose of the model is to relate the input signal u to the output y, which can then be used to predict the performance of a policy. The standard practice in quantum control is to construct a first-principle model of the quantum system that describes the evolution of the quantum state over time with u manipulating one or more coefficients of the Hamiltonian [78, 20, 71]. The choice of the mathematical description depends on whether the control is an open loop or a closed loop and how the measurement is implemented.

Using a first-principle model has the advantage that the model can provide insight into the mechanism behind the plant's response, but without the correct coefficients or simulation of noise and loss, the model may not be able to predict the observed data [17, 79]. The mathematical models can be made to match the input and output data from the plant, but the mathematical structure has to be chosen by the engineer and the coefficients estimated from experimental data, which introduces uncertainties to the model [77]. As such, the

model is never guaranteed to match the dynamics of the plant, and in some cases, a perfect match is not desirable because models can become too complex to be used in devising the policy [17].

Since the dynamics of the plant are known to be quantum mechanics, the models are based on the existing descriptions of quantum evolution. For open-loop control, master equations are used to describe the time-dependent state evolution. If the dynamic is assumed to be devoid of decoherence and noise, a Schrödinger equation can be used [80, 78], but otherwise, a Lindblad master equation is used when the dynamic is Markovian [81].

For closed-loop control, the quantum measurement causes the state to jump conditioned upon a random outcome, and this randomness must be captured in the dynamic equation in order to give accurate prediction [71]. When a continuous measurement is used, the evolution of the state is described by a stochastic master equation or a stochastic Schrödinger equation that includes a Wiener process, which is used to describe a stochastic process in Itô calculus [82, 83, 84]. The formalism can also be applied in the Heisenberg picture where an operator evolves following the Itô rule instead of the state [71], known as quantum filtering as the operator evolves according to the measurement outcomes.

In some applications, the quantum system is measured at certain points in its evolution, causing an instantaneous, discontinuous change to the state. The control task can then be divided into time steps, each step including a continuous evolution and a measurement. This state evolution can then be captured in a Schrödinger equation as a term governing an evolution in time and a jump conditioned on the measurement [71]. The discontinuous evolution can also be captured in another way by simulating each step of the measurement using a measurement operator corresponding to a random measurement outcome [39, 40].

Devising a control policy

Once the quantum dynamics is modelled, a policy can be generated for the control task. The first step in doing so is defining a performance measure and using the model to create

an objective function [15, 20]. The policy is optimized either analytically or numerically based on this function [85]. Here we give examples of the techniques without attempting to be complete as new techniques can be invented to address new challenges and constraints presented in a quantum control problem.

In open-loop quantum control, the control policies are often devised using quantum optimal control theory [15]. In this approach, the objective function is a combination of expected values of desired observables and constraints. The function is derived using the quantum dynamics such that the input to the function is a parameterized control sequence in time u(t) [20]. If the function is differentiable and simple enough to be solved analytically, then an analytical optimization method is applied. If finding an analytical solution is not possible, the time to complete the control task $[0, \tau]$ is discretized into time steps and numerical optimization is used to find an optimal or a feasible solution. A popular iterative optimization algorithm for this task is GRadient-Ascent Pulse Shaping (GRAPE), which is a gradient-based iterative method that uses the dynamic model to arrive at a feasible solution [86]. Other numerical optimization methods can also be applied [38].

For closed-loop control, the techniques are divided based on whether the state evolution is modelled using continuous- or discrete-time measurements. For a system that uses continuous measurement, an analytical approach to devising the policy is possible because a stochastic dynamic equation is available [82, 84]. The feedback using discrete-time measurement lends itself to a discretized numerical method, such as dynamic programming [83]. However, an analytical method is also possible based on Bayes's theorem, such as the one used in Berry and Wiseman's design of an adaptive interferometric phase estimation [39, 40]. Using the step-wise model, the Bayes's theorem is used to determine the prior of the unknown parameter ϕ after each measurement, and this prior is used to compute the imprecision. In this case, the imprecision is the objective of the control task, and the optimization is done in a step-wise fashion to minimize imprecision.

2.2 Data-driven control

In this section, we review the framework and approaches in data-driven control, an approach to control engineering that uses the output signal y from a plant to generate a policy ϱ [18]. The purpose of the data-driven approach is to circumvent the need for a first-principle model, which may not match the plant's dynamics. The automatic adjustment of ϱ is performed by an additional controller called supervisory controller SC, which can utilize a model that is created using the input and output signals from the plant or does not use a model at all. We also give examples of these techniques, including the policy search, which is the technique we use in this thesis.

2.2.1 Motivation and framework

In this subsection, we explain the motivation and framework of data-driven control. Although the first-principle models can give an insight into the physics of the plant, having an inaccurate model can lead to a policy that fails to deliver the target performance or at the worst fail to control the plant at all [18], and so data-driven control is proposed as an alternative to the model-based control engineering by adjusting ϱ based on its performance when applied to the plant.

The main concern with model-based control design is when the assumptions about the model do not match the plant's dynamics, known as the model mismatch problem [17]. A method to identify or quantify a mismatch does not exist, and the only way to discover the problem is to see an unpredicted output from applying the policy to the plant. This problem can stem from a structure or a parameter mismatch, which can be rectified by building a better model based on the input-output data [18]. However, the process of building an accurate model can take up more time and resources than is practical to invest.

If an accurate model can be practically obtained, the model-based approach should be chosen, but there is another model-related issue that might prevent a model-based method

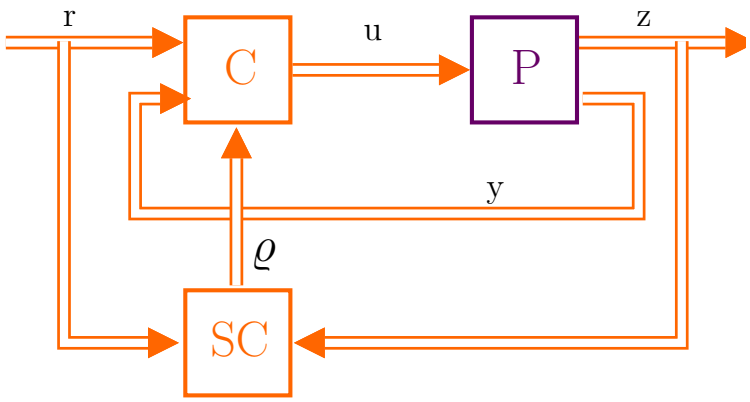


Figure 2.2: A block diagram of a control system with a loop for devising control policy using data from the plant P. In quantum control, the plant's dynamics obeys quantum mechanics. The supervisory controller SC devises policy ϱ using the information from the plant z such that the control goal r is satisfied and sends the policy to the controller C. The controller produces an input u to the plant based on the measurement outcome y, which may or may not be the same as z. If the control is open-loop, y does not influence the control signal.

from being successful. An accurate but complex model, especially with many nonlinear terms may make devising a policy intractable [20, 21], which is why linearizing methods exist although this again leads to model mismatch. This problem cannot be addressed within the framework of model-based control [18].

Data-driven control is proposed as an alternative approach to dealing with systems that cannot be efficiently modelled [87]. In this approach, the input-output data is used to automatically improve the policy by a supervisory controller SC that is augmented to the control loop (Fig. 2.2). The supervisory controller takes the control goal r and the output z from the plant as its input. The output z can include the measurement outcome y, which is used by the controller, but in general, y might not be the only output the SC requires for generating ϱ.

As the definition of data-driven control is broad, many techniques can be classified as data-driven techniques, some of which are already used in quantum control. Adaptive feedback control, for example, is a technique that adjusts a policy for open-loop quantum control based on its performance on a plant [20]. The mention of *feedback* in this approach refers to the SC loop and not the y channel. Reinforcement learning is another approach that

has been applied in quantum control to generate entangled state [88]. Reinforcement learning has been developed to solve decision-making problems where an agent decides on its next action based on the current state of the control system [52, 89], and so can be used to generate policies for discrete-time closed-loop control problems. When the state is only partially observable and the set of possible actions is large, a direct policy search technique is a scalable alternative to reinforcement learning [30]. This approach has been applied to adaptive phase estimation [90, 34] and bias estimation in quantum walk [35]. In the next subsection, we cover this policy search technique in more detail as we use the policy from this method in the comparison of quantum control policies.

2.2.2 Direct policy search

One technique that fits into the idea of data-driven control is policy search, which adjusts policy's parameters by trial and error [30]. In this subsection, we give an overview of the policy-search method, the control problems it solves, and the use of evolutionary algorithms as the optimization method to arrive at a set of parameters that deliver the control target.

Overview of policy search

Here we explain policy search and the problems it has been applied to. Policy search has been proposed as an alternative to reinforcement learning, and so the discussion of the control problems solved by this method is based on the framework of reinforcement learning, which we introduce briefly here by reviewing Markov decision process (MDP) and partially observable Markov decision process (POMDP). However, the idea of optimizing a pre-structured policy is not unique to solving MDP problems, we review the applications of policy search, including those that are not formulated as a decision-making process.

In the policy-search approach, a policy is given a task-appropriate structure, i.e., a parameterized mathematical function [29, 30, 91]. The policy is then optimized based on its performance when applied to the control plant. A policy-search method can either be model-

based or model-free depending on whether the method constructs a dynamic model before finding a successful policy or using the input-output data to directly construct a policy [30]. Here we do not attempt to review both techniques but focus on the model-free method, which is used for this work.

The idea behind the policy search can be applied to diverse control scenarios, but the term *policy search* is used mostly in the context of decision-making processes, specifically when the decision is Markovian [92]. The mathematical representation of an MDP[52, 89] consisted of

- a state set S, where state s describes the plant and the controller,
- an action set A, containing all possible actions u which the controller can take; it is also possible to create a subset $A(s)$ where $s \in S$ are the actions an agent can take in state s,
- transition probabilities $P(s_{m+1}|s_m, u_m)$, where m denotes the time step from the start of the task, and
- a reward $\mathcal{R}(s_{m+1}, u_m, s_m)$.

The trial-and-error approach to generating a policy is used when the MDP model is not complete [52], and so the dynamics have to be extracted by interacting directly with the plant.

Typically, the transition probabilities are missing when the MDP is incomplete, but one or more parameters that describe the state s may also be missing. Assuming that a measurement can be performed on the plant that yields an observable o related to s, the task can be described using a POMDP [93],

- a state set S,
- an action set A,

- transition probabilities $P(s_{m+1}|s_m, u_m)$,
- a reward $\mathcal{R}(s_{m+1}, u_m, s_m)$,
- an observable set $\mathcal{O}$ where observable o_m is an outcome that is obtained with probability $P(s_{m+1}, u_m, o_m)$.

The history of observations can be used by the controller to choose u or to infer a belief state before determining the appropriate action [94].

Policy search is known to be advantageous when applied to a decision-making process that presents certain characteristics, one of which is partial observability [95] and is claimed to be an 'asymptotically optimal method for episodic POMDP' [96]. Another characteristic that is considered is the size of the action space. When the set of actions the controller can make is large or continuous, policy search provides a scalable approach for devising policies [97]. These observations do not mean that all problems that exhibit these characteristics are best solved by policy-search methods, only that policy search is an attractive option that should be considered alongside other possible methods [29].

Because of the performance in POMDP and the large action space, policy search is widely applied in robotics, where the controllable parameters are often continuous in principle [30, 91] and sometimes in conjunction with reinforcement-learning methods [97]. However, a policy-search technique can also be found in other areas, such as in tuning PID controllers [33], which is an example of a model-free controller that needs to be tuned for a specific application. One group of algorithms that have been used to automatically tune the parameters is the evolutionary algorithms, which we review below.

Evolutionary algorithms for policy search

Evolutionary algorithms are global optimization algorithms that use the heuristic of biological evolution to maximize the performance (fitness) of a solution [65]. Because the optimization does not depend on the dynamic model, evolutionary algorithms are versatile

and can be easily applied in many fields, including control engineering [33]. Here we explain the idea behind evolutionary algorithms, the outline that all evolutionary algorithm share, and their applications in control engineering, especially in policy search.

The heuristics in evolutionary algorithms follow Darwin's theory of evolution by natural selection [98]; i.e., individuals with traits that help them thrive in an environment have a good chance of surviving and passing on these traits to their offspring. An individual's chance of survival is determined by its fitness with the fittest individual having the most chance of surviving and passing on its genetic material. After many generations, the favourable traits become dominant in the population as they are selected through competition between individuals.

The procedure of an evolutionary algorithm is analogous to this natural selection process [65]. An individual becomes a solution candidate, initialized randomly in the solution space to provide the diversity needed to generate a successful solution. The fitness of a candidate is determined using a fitness function, which outputs a real number indicating the performance of the candidate in solving the problem. One or more offspring is then generated for each candidate by combining the existing candidates. The fitness of each offspring is calculated, and a portion of the offspring and the parent population are selected for the next generation according to a selection rule. The size of the offspring population, the combination rule, and the selection rule are unique to the variant of the evolutionary algorithm. These rules along with the algorithm's parameters determine the algorithm's ability to explore the solution space and converge on a globally optimal solution although the convergence is not guaranteed [32].

Many evolutionary algorithms treat the fitness functions as a black box, i.e., the optimization does not use an explicit description of the fitness function only the output from the function, and thus evolutionary algorithms can be applied in various areas of control engineering [33]. Most relevant to this work is the use of evolutionary algorithms to optimize the parameters of model-free controllers, such as PID and PI controllers. The algorithms

have also been used to update the structure of the controller itself.

Being a black-box optimization also has drawbacks, especially when employed to search for policy's parameters. A black-box optimization ignores the contribution of individual actions to the success of a policy but evaluates the policy as a whole, which leads to a large number of trials needed to find a successful solution [89]. This observation, although true, does not mean evolutionary algorithms are always at a disadvantage to action-based optimization [95], and therefore need to be tested on a case by case basis. One case that policy-based optimization is the only possible method is when no information is available to evaluate the effectiveness of individual actions and only a delayed reward is available (also known as credit assignment problem [52]).

Chapter 3

Quantum-Enhanced Metrology

In this chapter, we review a field that quantum control has been applied to, i.e., quantum-enhanced metrology (QEM). Closed-loop quantum control systems have been employed in adaptive metrological schemes such that quantum-enhanced precision is achieved. We begin by describing the setup of QEM and explain the imprecision lower bounds obtained from using the classical and quantum resource, namely the standard quantum limit (SQL) and the Heisenberg limit (HL). We then turn to adaptive quantum-enhanced metrology (AQEM), which aims to achieve the goal of QEM using a sequence of adaptive measurements. One example of the AQEM scheme is the adaptive interferometric phase estimation, whose policy has been derived using a model-based Bayesian inference [39] and the data-driven policy-search approach [90]. We aim to compare these approaches when they are applied to interferometers including a phase-noise model that is not accounted for in the control design.

3.1 Quantum-enhanced metrological setup

The field of QEM aims to study high-precision estimation of an unknown parameter ϕ or parameters $\boldsymbol{\phi}$ by exploiting quantum resources [48]; i.e., particles that are in a non-classical collective state. In this section, we explain the framework of a QEM setup, in particular of

an indirect-measurement setup where a probe is used to interact with a system $\mathcal{I}(\phi)$ and ϕ is estimated from the change in the state of the probe. We review the types of input states that the probe is prepared in before describing the procedures that exploit this resource to achieve a quantum-enhanced precision.

3.1.1 Quantum resources

The use of a quantum resource to improve measurement precision was first considered for gravitational wave detection [41, 42, 43] and soon found other applications, such as in atomic clocks [45] and magnetometry [46, 99]. The appeal of QEM is in its ability to improve the precision without having to increase the power of the probe, which is a limiting factor for high-power measurement setups similar to the gravitational wave detector. The constraint on the power tolerated by the hardware also applies to low-energy applications, such as in quantum information technology where the hardware is designed to perform at the few-photons level. In this subsection, we give a brief overview of the quantum resources considered for both the high-power and low-power applications.

The necessary resource that enables quantum-enhanced precision is a probe that has been prepared in a non-classical state [100], either as a squeezed state [101] or an entangled state [102]. A squeezed state describes a quantum state whose uncertainty falls below the ground state for one observable, e.g. the position, and rises in another non-commuting variable, e.g. the momentum [103]. An entangled state describes a quantum system where there are correlations between subsystems, and so the state of the system cannot be described as a product of its parts. Although the terminology divides the resource in such a manner, squeezing and entanglement are connected [104], both are ways of describing quantum correlation that cannot be reproduced by classical correlation.

In QEM, the use of a squeezed state appears when the input state consists of a high number of particles, such as in the case of gravitational wave detection [42, 43] or atomic clocks [45], as these applications utilize probes that consist of millions of particles. Cur-

rent technology allows the generation of a squeezed state for a number of particles as high as 10^6 [105], whereas entangling the same number of particles is currently impossible [106, 107]. The quantum-enhanced precision is attained by exploiting the squeezed variable, enabling a measurement with reduced imprecision.

In this thesis, we employ a QEM scheme that utilizes N d-level entangled particles prepared in a collective state

$$\rho \in \mathcal{S}\left(\mathscr{H}_d^{\otimes N}\right), \tag{3.1}$$

which is in the space of positive-definite, trace-class, self-adjoint linear operators acting on a tensor product of N copies of a d-dimensional Hilbert space $\mathscr{H}_d$ [108]. How an entangled state can lead to enhanced precision is less straightforward than the case of the squeezed state. Because of the connection between entanglement and squeezing, the enhancement can be explained as a reduction of an uncertainty [22]. Another perspective attributes the advantage to the cumulative effect of ϕ on the entangled state, giving the same effect as interacting a single-particle probe N times with the system [109]. This perspective can explain certain estimation schemes such as two-mode phase estimation using a N00N state.

For both the squeezed state and the entangled state, the input state ρ has to be optimized to achieve minimal imprecision as the imprecision depends in part to the strength of the interaction between ρ and the system, described by a quantum channel $\mathcal{I}(\phi)$ [110]. To put it another way, in order to infer ϕ, the difference between the output state coming from ϕ and $\phi + \mathrm{d}\phi$ must be as different from each other as possible. As $\mathcal{I}(\phi)$ is in most cases assumed to be fixed, the optimization of the interaction depends on using the optimal input state for the particular channel. If there is a mismatch between the quantum channel used the find the optimal input state and the physical system, the performance of the QEM scheme can be greatly diminished. An example of this is the N00N state, which is known to be susceptible loss [111]. In this work, we use a known optimal state for interferometric phase estimation with a loss-resistant property and do not attempt to find an optimal state for when the phase noise is included as we attempt to test the robustness of the scheme when

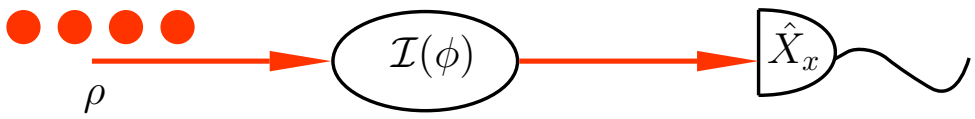


Figure 3.1: A diagram of a measurement procedure using quantum resource.

the noise model is not known during the control design.

3.1.2 Quantum measurement procedure

In this subsection, we explain the QEM procedure (Fig. 3.1), which can be divided into three parts: the preparation of the input state, the interaction between the input state and the system, and the measurement of the output state [8]. The last two steps are determined by the measurement strategy used. We focus mainly on the collective measurement strategy in this subsection as it is the approach used in the calculation of the imprecision lower bound.

A collective strategy is only one of many ways the quantum resource can be used in QEM, two others being a sequential and an adaptive strategy. The collective strategy has all the particles in the probe collectively interacting with the system and then measured. The sequential strategy is where the probe interacts with the system multiple times before a measurement is performed [112]. Although this strategy achieves high-precision by accumulating the effect of ϕ rather than using the collective behaviour of a quantum state, this strategy can be extended by entangling the probe with ancilla qudits, which can then be used with a quantum error correction algorithm to mitigate the effect of noise and loss [113]. The adaptive strategy incorporates a feedback control to a sequential strategy, and in some cases only a part of the input state is interacting with the system at any given time step. Once the output state, or part of the output state, is measured, the feedback loop changes the controllable parameter Φ according to the outcome [100, 76].

For the rest of this subsubsection, we focus on the mathematical formalism of the collective measurement strategy used in conjunction with an N particle entangled state. All the N

d-level particles interact with the system represented by a quantum channel (completely-positive trace-preserving map) $\mathcal{I}(\phi)$. This map acts on any ρ, with a single unknown parameter ϕ in the channel [114], and can be represented using Kraus operators $\{\mathcal{K}_0, \mathcal{K}_1, \ldots \mathcal{K}_\mu\}$ acting on $\mathscr{H}_d^{\otimes N}$ such that

$$\mathcal{I}(\phi)\rho = \sum_i^\mu \mathcal{K}_i(\phi)\rho\mathcal{K}_i^\dagger(\phi) \tag{3.2}$$

and $\sum_{i=0}^{\mu} \mathcal{K}_i(\phi)\mathcal{K}_i^\dagger(\phi) = \mathbb{1}$. The size of the set of Kraus operators is bounded by the dimension of the Hamiltonian, i.e., $\mu \leq \dim \mathscr{H}_d^{\otimes N}$ [115]. In the special case of an isolated system, the channel is represented by a unitary transformation [47]

$$\mathcal{I}(\phi)\rho = U(\phi)\rho U^\dagger(\phi) \tag{3.3}$$

for a unitary operator U.

After the particles exit the system, they are measured, and this measurement is described as a positive-operator-valued measure [116, 117], which is a positive semidefinite operator

$$X_x : \mathscr{H}_d^{\otimes N} \to \mathscr{H}_d^{\otimes N}, \sum_x X_x = \mathbb{1}, \tag{3.4}$$

assuming the measurement outcomes $\{x\}$ are in a finite set. This outcome is random with probability

$$P(x|\phi) = \mathrm{tr}\left(X_x \mathcal{I}(\phi)\rho\right). \tag{3.5}$$

The measurement X_x is repeated multiple times to sample the distribution (Eq. 3.5) sufficiently well, and ϕ is then inferred from these samples.

3.2 Quantum imprecision

In this section, we explain the imprecision of an estimate of ϕ obtained from a QEM scheme, which is determined by the procedure for inferring ϕ and from the distribution of the mea-

surement outcomes. We begin by reviewing quantum parameter estimation, which calculates the estimate $\tilde{\phi}$ from the measurement outcomes. This procedure can increase the imprecision $\Delta\tilde{\phi}$, and so the lower bound of the imprecision is computed from the distribution of the outcomes rather than the estimate. This lower bound is formally known as the Cramér-Rao lower bound (CRLB), which in the quantum version depends on the quantum channel and the input state. The choice of the classical and quantum resource leads to the imprecision limits, the SQL and the HL, that are used as benchmarks for QEM procedures.

3.2.1 Quantum parameter estimation

Parameter estimation is a study of how to use the measurement outcomes $\boldsymbol{x}_M = x_1 x_2 \cdots x_M$ to infer an unknown parameter ϕ. The quantum version of parameter estimation uses $\boldsymbol{x}_M$ that are generated by a quantum mechanical process, which may negate certain assumptions about the probability distribution of the outcomes, such as the assumption that the outcomes are sampled from an independent and identical distribution (i.i.d.). In this subsection, we explain the connection between the distribution of the outcomes and the distribution of the estimate without imposing such a condition.

The estimate $\tilde{\phi}$ is obtained from an estimator [118],

$$\tilde{\phi} := \hat{\Phi}\left(\boldsymbol{x}_M\right), \tag{3.6}$$

which we assume here to be a deterministic function. Because $\boldsymbol{x}_M$ is a finite string of random variables, $\tilde{\phi}$ is also random. From multiple instances of estimating ϕ, the probability $P\left(\widetilde{\phi}|\phi\right)$ can be reconstructed (Fig. 3.2). Assuming the distribution to be unimodal, the width of $P\left(\widetilde{\phi}|\phi\right)$ determines the imprecision $\Delta\tilde{\phi}$, and the difference between the peak $\bar{\tilde{\phi}}$ and ϕ is the bias.

For a deterministic estimator, the distribution of the estimate can be written in terms of

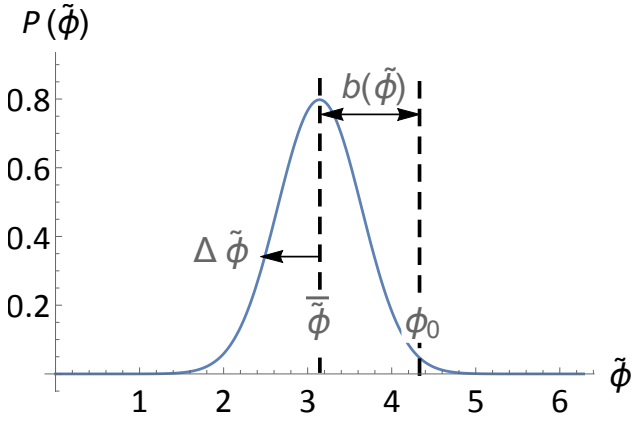


Figure 3.2: An example of a distribution of the estimate $\tilde{\phi}$. The value ϕ_0 indicates the value to be estimated which in the text is ϕ, whereas $\bar{\tilde{\phi}}$ is the peak of the distribution, which is also the mean of the estimate for a normally-distributed $\tilde{\phi}$. In some estimation scheme, $\bar{\tilde{\phi}}$ may be reported as the final estimate. The bias is $b(\tilde{\phi}) = \bar{\tilde{\phi}} - \phi_0$ and $\Delta\tilde{\phi}$ is the imprecision of $\tilde{\phi}$.

the distribution of the data,

$$P(\widetilde{\phi}|\phi) = \sum_{\boldsymbol{x}_M \in \hat{\Phi}^{-1}(\widetilde{\phi})} P(\boldsymbol{x}_M|\phi). \tag{3.7}$$

Each x_m is obtained independently from a measurement of an N-particle quantum state. Thus the distribution of $\boldsymbol{x}_M$ can be written as

$$P(\boldsymbol{x}_M|\phi) = \prod_{m=0}^{M} P(x_m|\phi). \tag{3.8}$$

The key to attaining quantum-enhanced precision is by manipulating Eq. 3.8 through the selection of an appropriate quantum state and measurements. For $M = 1$, the imprecision scales as

$$\Delta\tilde{\phi} \in O\left(N^{-\wp}\right). \tag{3.9}$$

This power-law relationship can be shown by calculating the quantum CRLB, which takes into account that the distribution Eq. 3.5 is computed from particles that obey quantum

mechanics. We review this procedure below.

3.2.2 Quantum Cramér-Rao lower bound

The Cramér-Rao lower bound defines the lower limit in the imprecision that can be obtained from a set of measurement outcomes using the Fisher information, which shows the average change in the distribution $P(\boldsymbol{x}_M|\phi)$ (Eq. 3.8) in the locality of ϕ [119]. The quantum version of Fisher information can be derived using the quantum version of $P(\boldsymbol{x}_M|\phi)$, and from that the optimal POVM can be calculated. However, the optimal POVM is not guaranteed to be practical.

Specifically, Cramér-Rao lower bound is defined for the variance of the estimate and is different for an unbiased and a biased estimator. For unbiased estimator the inequality is

$$\left(\Delta\tilde{\phi}\right)^2 \geq \frac{1}{F(\phi)}, \tag{3.10}$$

where $F(\phi)$ is the Fisher information:

$$F(\phi) := \sum_{\forall \boldsymbol{x}_M} P(\boldsymbol{x}_M|\phi) \left(\frac{\partial P(\boldsymbol{x}_M|\phi)}{\partial \phi}\right)^2. \tag{3.11}$$

The biased estimator contains a correction,

$$\left(\Delta\tilde{\phi}\right)^2 \geq \frac{1 + \frac{\partial b(\phi)}{\partial \phi}}{F(\phi)}, \tag{3.12}$$

where the bias is $b(\phi) = \bar{\tilde{\phi}} - \phi$. This correction is to account for the difference between the variance of a biased and an unbiased estimator, and not a fundamental change in Fisher information as the quantity only depends on the distribution of the outcomes. Intuitively, Fisher information measures how easily the $\boldsymbol{x}_M$ generated at ϕ can be differentiated from the $\boldsymbol{x}_M$ generated at $\phi + \mathrm{d}\phi$. The more easily the data can be differentiated, the higher the Fisher information and the smaller the imprecision.

The Cramér-Rao lower bound is also extended to quantum parameter estimation. In the case where the collective QEM scheme is used, the quantum Fisher information is derived for an instance of the phase estimation using a Hermitian operator $\mathcal{L}_\phi$ known as symmetric logarithmic derivative [54], which is defined to be

$$\frac{\partial \rho_\phi}{\partial \phi} = \frac{\mathcal{L}_\phi \rho_\phi + \rho_\phi \mathcal{L}_\phi}{2}. \tag{3.13}$$

This definition is substituted into the Fisher information calculation, which leads to [55]

$$\begin{aligned} F_Q(\phi) &= \sum_{\forall x} \frac{\mathrm{Re}(\mathrm{tr}(X_x X_x^\dagger \rho_\phi \mathcal{L}_\phi))^2}{\mathrm{tr}(X_x X_x^\dagger \rho_\phi)} \\ &\leq \sum_{\forall x} \left| \frac{\mathrm{tr}(X_x X_x^\dagger \rho_\phi \mathcal{L}_\phi)}{\sqrt{\mathrm{tr}(X_x X_x^\dagger \rho_\phi)}} \right|^2 \\ &\leq \mathrm{tr}\left(\left(\sum_{\forall x} X_x X_x^\dagger \right) \mathcal{L}_\phi \rho_\phi \mathcal{L}_\phi \right) \\ &\leq \mathrm{tr}(\mathcal{L}_\phi \rho_\phi \mathcal{L}_\phi). \end{aligned} \tag{3.14}$$

Hence, quantum Fisher information is independent of the measurement but instead depends on the input state and the quantum channel.

In practice, the quantum CRLB can only be achieved if an optimal measurement is performed. The POVM of this measurement can be calculated from the quantum Fisher information by finding the eigenstates of the Hermitian $\mathcal{L}_\phi$, which may not be feasible to be implemented. The implication, therefore, is that a certain QEM scheme may not achieve the lower bound due to the limitation in collective measurements that can be performed. The idea behind adaptive measurement is to approximate the optimal collective measurement by a sequence of measurements [12]. Hence, the lower-bound of the adaptive procedure is assumed to be the same as the one calculated from the collective measurements, which we review in the following subsection.

3.2.3 Imprecision limits

Using the quantum CRLB, the SQL and HL can be derived for a QEM scheme by considering the use of a classical and a quantum resource respectively. The derived lower bound is a function of N, which asymptotically takes the form of a power-law relationship (Eq. 3.9). The SQL asymptotically approaches the power-law scaling of $\wp = 1/2$, whereas HL exceeds this number and is used to claim optimality of a QEM scheme. In this subsection, we discuss how these bounds originated and in particular we give the SQL and the HL of the interferometric phase estimation.

Standard quantum limit

The SQL is the imprecision lower bound when a classical resource is used, which means that the input state (Eq. 3.1) is separable, i.e., unentangled [120]. The simplest case of such a separable state is a tensor product of N independent particles [48],

$$\rho = \rho_1^{\otimes N}. \tag{3.15}$$

Interacting this state with the quantum channel leads to the output state

$$\left(\mathcal{I}(\phi)\rho_1\right)^{\otimes N}. \tag{3.16}$$

Measuring the output state leads to a distribution in Eq. 3.8 where

$$P(x_m) = \mathrm{tr}\left(X_{x_m}\mathcal{I}(\phi)\rho_1\right), \tag{3.17}$$

is independent and identical for $m \in \{1, 2, \ldots, M\}$.

Due to Fisher information's additive property under the i.i.d. condition,

$$F^{(N)}(\phi) = \sum_{n=1}^{N} F_n^{(1)}(\phi) = NF^{(1)}(\phi). \tag{3.18}$$

Therefore,

$$(\Delta\tilde{\phi})^2 \geq \frac{1}{NF^{(1)}(\phi)}, \tag{3.19}$$

for an unbiased estimator. Asymptotically, the standard quantum limit is

$$\Delta\widetilde{\phi} \propto O\left(\frac{1}{\sqrt{N}}\right) \tag{3.20}$$

irrespective of the quantum channel.

Heisenberg Limit

If quantum resources are employed, e.g., squeezed state [101] or entangled state [102], the SQL can be surpassed [43, 22]. Since this lower bound is computed from quantum CRLB [54], which depends on the input state and the quantum channel [55], the HL is specific to the QEM scheme [121] unlike the SQL. In the case of interferometric phase estimation, the HL is known to be [49]

$$\Delta\widetilde{\phi} \propto O\left(\frac{1}{N}\right), \tag{3.21}$$

although this limit can only be attained through the use of optimal measurement. Because the optimal measurement as defined by the POVM's may not be physically implementable, adaptive phase estimation schemes have been proposed to approximate this lower bound [76].

3.3 Adaptive quantum-enhanced metrology

AQEM is proposed as an approach to approximate an optimal measurement for quantum-enhanced metrology [12]. AQEM employs a closed-loop control to adapt the measurement, which can either be continuous-time measurements [122, 123, 124, 125], such as homodyne detection, or discrete-time measurements [39, 76, 90], such as single-photon detectors. In this work, we focus on the discrete-time construction of an AQEM procedure whose evolution is described by jumps in the quantum state. Here we lay down the general procedure, which we

later turn into a decision-making process in order to obtain a successful policy (Chapter 4).

3.3.1 Adaptive measurement procedure

In a discrete-time adaptive procedure, the N-entangled particles are split into bundles, and the outcome of a measurement on one of the bundles is used to adjust the measurement for when the next one arrives [76]. In this setup, the measurement and the parameter estimation are not two separate stages but interweaved with a guess of the estimate being made after each measurement. As such, the feedback procedure is also the estimator, and hence the precision and accuracy are determined by the feedback policy.

Instead of a collective measurement, we can consider measuring subsets of particles, which we call bundles, and, at the extreme limit, which is of interest here, the individual-measurement case of measuring a single particle at a time. Mathematically, we split the particles into M bundles of L particles where $N = ML$ [2] so the Hilbert space can be expressed as

$$\left(\underbrace{\mathscr{H}_d^{\otimes L}}_{\text{bundle}}\right)^{\otimes M}. \tag{3.22}$$

In this case, both $\mathcal{I}(\phi)$ and X_x act on $\mathscr{H}_d^{\otimes L}$. For localized measurements on each bundle, the POVM is

$$\bigotimes_{m=1}^{M} X_{x_m}^{(m)},\ X_{x_m}^{(m)} : \mathscr{H}_d^{\otimes L} \to \mathscr{H}_d^{\otimes L} \tag{3.23}$$

with outcomes from this tensor-product POVM being concatenations of M length L strings of d-dimensional digits,

$$\boldsymbol{x}_M = x_1 x_2 \cdots x_M \in \mathbb{N}_{d^L}^{\otimes M}, \tag{3.24}$$

where

$$x_m \in \mathbb{N}_{d^L} := \{0, 1, 2, \ldots, d^L - 1\} \tag{3.25}$$

is measured from the m^{th} bundle.

In one extreme case, each bundle contains only one particle, which leads to $M = N$ and

$L = 1$. The string of outcomes becomes

$$\boldsymbol{x}_N = x_1 x_2 \cdots x_N \in \mathbb{N}_d^{\otimes N}. \tag{3.26}$$

The POVM is

$$\bigotimes_{m=1}^{N} X_{x_m}^{(m)},\ X_{x_m}^{(m)} : \mathscr{H}_d \to \mathscr{H}_d, \tag{3.27}$$

which is a tensor product of N qudit POVMs.

If the particles are two-level particles, the state (Eq. 3.1) is simplified to

$$\rho \in \mathcal{S}\left(\mathscr{H}_2^{\otimes N}\right), \tag{3.28}$$

and the POVM simplifies from Eq. 3.27 to

$$\bigotimes_{m=1}^{N} X_{x_m}^{(m)},\ X_{x_m}^{(m)} : \mathscr{H}_2 \to \mathscr{H}_2. \tag{3.29}$$

The outcome (Eq. 3.26) is simplified to

$$\boldsymbol{x}_N \in \{0,1\}^{\otimes N}, \tag{3.30}$$

which is an N-bit string.

3.3.2 Feedback control and inference

AQEM employs quantum feedback control to create adaptive measurements [84], but the control signal may not be attached directly to the measurement device. Rather, the signal is connected to a controllable component in front of a measurement device, and the combination of the component and the device emulates a POVM. However, the controllable component can instead be modifying a quantum channel while the measurement device remains static through the procedure, and the probability of x_m would still be the same as having an

adaptive measurement.

In this thesis, we assume that the signal is connected to a controllable parameter Φ inside a quantum channel. Hence, the channel acting on the m^{th} bundle is $\mathcal{I}(\phi;\Phi_m)$. Measurement of the m^{th} bundle leads to an update of the control parameter to Φ_{m+1} for the next bundle. The control-parameter update is determined by a policy

$$\varrho : \Phi_m \mapsto \Phi_{m+1} \tag{3.31}$$

according to $\boldsymbol{x}_m = x_1 x_2 \cdots x_m$ of outcomes. The estimate is

$$\tilde{\phi} := \Phi_{M+1}, \tag{3.32}$$

and therefore an AQEM procedure can be used for single-shot measurement, i.e., inferring ϕ from one instance of the measurement procedure.

Inferring $\tilde{\phi}$ from Φ_{M+1} is a choice that we implement in our procedure and is not the only or the best option for an estimator. An adaptive measurement procedure that uses the Bayesian inference, for example, infers $\tilde{\phi}$ from the prior of ϕ, which is updated based on the measurement outcomes and is independent of Φ_{M+1} [12]. Whereas calculating a prior uses a model of the quantum dynamics, Φ_{M+1} is a known parameter and thus inferring $\tilde{\phi}$ using Eq. 3.32 can be done without a need for a model.

3.4 Adaptive phase estimation

Phase estimation underlies many QEM applications [43, 47, 48] and thus is widely used for devising quantum-enhanced techniques, including several AQEM schemes [76, 90, 123]. Here we explain quantum-enhanced adaptive phase estimation (QEAPE), which we numerically simulate to determine its imprecision. The QEAPE scheme is controlled by Bayesian feedback [39, 40] or DE-designed policies [34, 35, 3], which we compare in terms of robustness

and resource consumption for control.

3.4.1 Quantum-enhanced adaptive phase estimation

One method of estimating phase is to use an interferometer, which infers phase shifts from the interference between two or more modes [126]. In particular, we use QEAPE based on a Mach-Zehnder interferometer, which has two modes and therefore we are looking at the case of $d = 2$ representing the modes. The mathematics of Mach-Zehnder interferometry applies to other forms of SU(2) interferometry, such as Ramsey, Sagnac and Michelson interferometry [48, 47, 127]. In this subsection, we present the input state, adaptive channel, detection, feedback, inference and imprecision.

Input state

For non-adaptive quantum interferometry with collective measurement, the unitary interferometric transformation is in the Lie group SU(2) with irrep (Casimir-invariant label) $j = N/2$. For adaptive quantum interferometry or individual measurements, the interferometric unitary transformation is SU(2^N) for N particles and two paths. However, the two descriptions converge if the input state is permutationally symmetric; technically, Schur-Weyl duality dictates that the applicable transformation is SU(2) with irrep $j = N/2$ [128].

Notationally, modes are labelled by

$$\varepsilon_m \in \{0, 1\}, \tag{3.33}$$

which conveys which of the two paths, such as input or output port or intra-interferometric path, pertains. Thus, the state $|\epsilon_m\rangle$ refers to the m^{th} photon being in path ϵ_m. The multi-photon basis is the tensor-product state

$$|\boldsymbol{\epsilon}_N\rangle = \bigotimes_{m=1}^{N} |\epsilon_m\rangle \,. \tag{3.34}$$

For $\mathrm{ham}\,\boldsymbol{\epsilon}$ the Hamming weight, i.e., sum of bits, of $\boldsymbol{\epsilon}$, the permutationally-symmetric basis is

$$|n, N_a - n\rangle = \binom{N_a}{n}^{-1/2} \sum_{\mathrm{ham}\,\boldsymbol{\epsilon}_{N_a}} |\boldsymbol{\epsilon}_{N_a}\rangle \tag{3.35}$$

for N_a the total number of particles in mode a.

The sine state serves as a loss-tolerant symmetric state that minimizes phase-estimation imprecision [129, 124, 125], and is expressed as [39, 40]

$$\begin{aligned} |\psi\rangle_N = &\left(\frac{N}{2}+1\right)^{-1/2} \sum_{n,k=0}^{N} \sin\left(\frac{k+1}{N+2}\pi\right) \mathrm{e}^{\mathrm{i}\pi(k-n)/2} \\ &\times d^{N/2}_{n-N/2,k-N/2}\left(\frac{\pi}{2}\right) |n, N-n\rangle\,, \end{aligned} \tag{3.36}$$

for $d^{j}_{m,m'}(\beta)$ the Wigner-d function [130]. This state also has the advantage of being robust against photon loss [128], which is a desirable property for practical QEM and so is used in the QEAPE procedures.

Adaptive Channel

The particles in the sine state (Eq. 3.36) are divided into single-particle bundles ($L = 1$ case), each of which passes through the Mach-Zehnder interferometer. For a noiseless interferometry, the quantum channel for one photon is

$$U_1(\phi; \Phi_m) = \exp(\mathrm{i}(\phi - \Phi_m)\hat{\sigma}_y),\ \phi, \Phi_m \in [0, 2\pi) \tag{3.37}$$

for $\hat{\sigma}_y$ a Pauli matrix [131]. Therefore, the channel is

$$U(\phi; \Phi_m) = U_1(\phi; \Phi_m) \otimes \cdots \otimes \mathbb{1}^{(N)} \tag{3.38}$$

acting on the state space (Eq. 3.28).

In a physical implementation of an interferometer, mechanical disturbances, air-pressure

changes and the thermal fluctuations induce optical-path fluctuations. These effects randomize the phase difference

$$\phi - \Phi_m \tag{3.39}$$

according to prior distribution $p(\phi)$. The quantum channel is thus [34]

$$\begin{aligned} &\mathcal{I}(\phi;\Phi_m) : \mathcal{S}\left(\mathscr{H}_2^{\otimes(N-m)}\right) \to \mathcal{S}\left(\mathscr{H}_2^{\otimes(N-m)}\right) : \\ &\rho_m \mapsto \int_{\phi=0}^{2\pi} \mathrm{d}\phi\, p(\phi) U^\dagger(\phi;\Phi_m)\rho_m U(\phi;\Phi_m), \end{aligned} \tag{3.40}$$

where ρ_m is the state after the $(m-1)^{\text{th}}$ photon is measured.

Detection, feedback, and inference

After the m^{th} photon passes through the interferometer, the photon is detected by one of the single-photon detectors positioned outside the output ports. The information of the exit port is $x_m \in \{0,1\}$, which is given to a controller. The controller then uses this information to compute Φ_m from ϱ before the next photon arrives. The procedure of simulating the injection of the next photon from the sine state (Eq. 3.36) followed by the action of the channel (Eq. 3.38) and then measurement at the output ports is repeated until all photons are consumed, and the estimate is inferred from Φ_{N+1}, assuming no loss of photons.

Imprecision

Imprecision of the estimate (Eq. 3.9) is related to the Holevo variance [124]

$$\left(\Delta\tilde{\phi}\right)^2 = V_{\text{H}} := S^{-2} - 1 \tag{3.41}$$

by the sharpness function

$$S = \frac{1}{K}\left|\sum_{k=1}^{K} \exp\left[\mathrm{i}(\phi_0^{(k)} - \tilde{\phi}^{(k)})\right]\right|, \tag{3.42}$$

which estimates the width of a distribution $P(\tilde{\phi}|\phi)$ over a periodic variable. As this distribution is determined by $P(\boldsymbol{x}_N|\phi)$ (Eq. 3.7), the ideal method of calculating S is to determine the probability of every $\boldsymbol{x}_N$ given a policy ϱ. However, due to the exponentially number of $\boldsymbol{x}_N$ with respect to N, we instead numerically compute the sharpness by simulating the adaptive phase estimation for $K = 10N^2$ samples of ϕ_0, which is the unknown (noiseless) interferometric phase shift taken uniformly from $[0, 2\pi)$ [90]. We will discuss the how this sampling method is done in detail in Chapter 4 once we establish the discrete-time QEAPE as a decision-making process.

In principle, the sharpness has a definite value for a particular $P(\tilde{\phi}|\phi)$. However, as we estimate the sharpness by sampling, the resulting S from Eq. 3.42 is a random variable sampled from a distribution $P_{\text{sharp}}(S)$ with a mean

$$\bar{S} = \sum S P_{\text{sharp}}(S). \tag{3.43}$$

The distribution $P_{\text{sharp}}(S)$ is also affected by phase noise as this additional undertainty increases the width of $P(\tilde{\phi}|\phi)$. As such, we use the mean sharpness (Eq. 3.43), which average out the increased sampling noise, as the fitness for the policy-search algorithm, as we will explain in Chapter 5.

3.4.2 Policy design

In this subsection, we explain how the policy ϱ (Eq. 3.31) is obtained. Specifically, we consider two approaches. First we consider the Bayesian approach, based on Bayes theorem [132], which applies if the controller has complete knowledge about the quantum system

and updates this knowledge based on measurement outcomes [133]. The second approach is to execute a direct policy search for an optimal or feasible ϱ [30], which has the advantage over the Bayesian approach that a trusted model is not required.

Bayesian Feedback

Early examples of AQEM employ the Bayesian approach [133, 39, 40], which involves a fully trusted complete model and then performs computationally intensive decision making as measurement data arrive, leading to updates of the prior. For each particle exiting the interferometer, the probability for the particle being detected in output port given by outcome x_m is computed by assuming a perfectly known input state and known quantum dynamics such as unitary interferometric evolution (Eq. 3.38). The prior for ϕ_0 is then updated using Bayes's theorem. The width of this prior is quantified by $\sqrt{V_{\mathrm{H}}}$ (Eq. 3.41) at each measurement step; the optimal Φ_m that minimizes V_{H} for the next particle is then computed from this trusted model. Although the Bayesian method approaches the HL [40], ϱ might not be robust to input-state variability or to system noise, which is a problem for designing controllers [77, 79].

Policy search using differential evolution

Policy search provides an attractive alternative to the Bayesian approach when a trusted model is lacking or if the computational overhead associated with the Bayesian approach is excessive. In this approach, the policy is given a task-appropriate structure and the performance is optimized by searching the space of the policy's parameters [29, 30]. A policy-search method is called model-free if the optimization process does not include learning a model of the dynamic but directly updates the policy parameters [30], which can be achieved by employing a black-box optimization algorithm [31] using only policy performance. Evolutionary algorithms form a class of black-box optimization algorithms whose policy update procedure is inspired by models of biological evolution [32]. We use DE to search for a feasible policy

for QEAPE. Once the policy is found, no further training is necessary.

For the task of QEAPE, we assume a logarithmic-search heuristic Markovian update rule [90]

$$\Phi_{m+1} = \Phi_m - (-1)^{x_m}\Delta_m, \tag{3.44}$$

with phase-adjustment vector

$$\boldsymbol{\Delta} := (\Delta_1, \Delta_2, \ldots, \Delta_N)\,, \tag{3.45}$$

where $\Delta_m \in [0, 2\pi)$. This update rule (Eq. 3.44) corresponds to the controller turning the "phase knob" up or down by a fixed amount Δ_m after the m^{th} photon, subject only to the previous outcome and ignoring the full measurement history. The action that the controller performs is therefore

$$u_m = (-1)^{x_m}\Delta_m \tag{3.46}$$

in this particular feedback procedure.

During the training stage where we optimize $\boldsymbol{\Delta}$, an individual in DE is initially assigned a random set of parameters $\boldsymbol{\Delta} \in [0, 2\pi)^N$. The performance of the policy is evaluated using the mean sharpness (Eq. 3.43) [34], as the interferometer is assumed to contain noise and this performance measure averages out noise. In practice, due to the high cost of sampling $P_{\text{sharp}}(S)$ (Eq. 3.43), few runs are performed to obtain $\bar{S}$. A DE algorithm then uses this performance as the fitness to search iteratively for $\boldsymbol{\Delta}$ such that $\bar{S}$ is maximized [3].

The use of DE incurs a time cost for generating ϱ, quantified by a loop analysis of the algorithm assuming N photons are used in the phase estimation. The algorithm is assumed to run on a single processor, which leads to a serial evaluation of $\boldsymbol{\Delta}$ for every iteration of DE. The scaling of time cost with respect to N determines the complexity, which has been shown to be polynomial [35], and so the degree of the polynomial convey the complexity for generating ϱ.

Previous policy-search algorithms that are designed for adaptive phase estimation have

upper bounds in the number of particles that a successful policy can still be found. The algorithm that is designed for normal-distribution noise is able to deliver $\wp > 1/2$ up to $N = 45$ [34], whereas the algorithm that utilizes differential evolution is capable of generating successful policies up to $N = 98$ but is not noise-resistant [35]. We address this issue by devising a scalable, noise-resistant policy-search algorithm in Chapter 5.

Chapter 4

AQEM as a Decision-Making Process

In this chapter, we present an alternative model of a discrete-time AQEM, i.e., a decision-making process, which enables us to connect AQEM problems to known techniques in policy generation, such as reinforcement learning and policy search. This representation also enables us to formulate a method for calculating the CRLB given a policy, which makes it difficult to calculate explicitly. Here we use adaptive phase estimation from Sec. 3.4 as an example for discrete-time quantum feedback control. We formulate this problem as a decision-making process and analyze its characteristics to show that policy search is a suitable approach for generating a successful policy. We then simulate the QEAPE schemes without phase noise that use either an N-particle product state, which is an example of a quantum state without a correlation between particles, or a sine state, which is an N-particles entangled state, in estimating an unknown phase shift. We use the imprecisions from these simulations to demonstrate the role of entanglement in reducing QEAPE's imprecision.

4.1 A generalized AQEM

In this section, we describe an adaptive measurement scheme that divides N particles into bundles of equal size, expanding on the explanation given in Sec. 3.3. We then conceive this scheme as a decision-making process, allowing us to represent the procedure as a decision

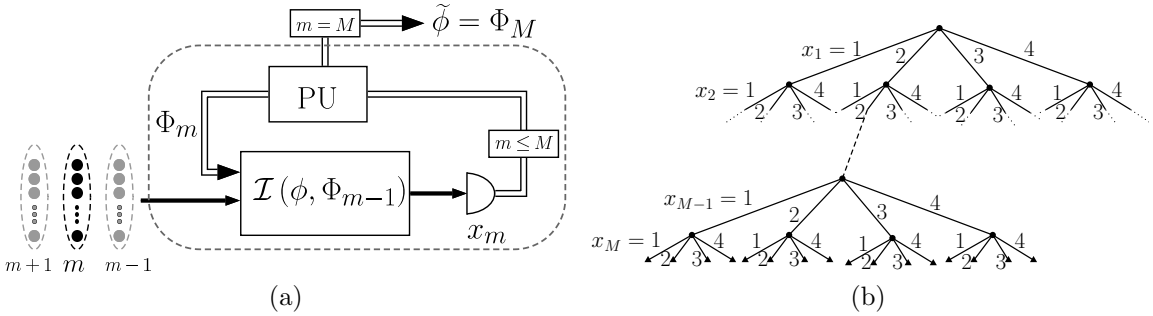


Figure 4.1: (a) A diagram of AQEM procedure. The input state ρ_1 is divided into M bundles of L particles, each being injected into the process one after another. After the detection of the m^{th} bundle, the processing unit (PU) uses the history of the measurement outcomes up to that point, $\boldsymbol{x}_m = x_1 x_2 \dots x_m$, in order to determine the control parameter Φ_{m+1}, taking the process $\mathcal{I}(\phi, \Phi_m) \mapsto \mathcal{I}(\phi, \Phi_{m+1})$ before the $(m+1)^{\text{th}}$ bundle passes through. This process is repeated until all bundles are measured and Φ_{M+1} is reported as the estimate. (b) An example of a decision tree representing an adaptive measurement using M bundles. Each measurement (a node) produces one of the $d^L = 4$ possible outcomes represented by a branch. The number of leaves grows exponentially with depth M. Here we only expand one of the subtrees at the depth $M-1$ and M. [Reproduced from Ref. [2], Fig. 1]

tree. We then discuss the characteristics of this decision-making process and approaches to generating a policy. The formalism leads to an imprecision and a lower bound that are dependent on the policy.

4.1.1 Adaptive measurement procedure

In this subsection, we describe a scheme for single-shot quantum-enhanced adaptive measurement (Fig. 4.1a). The aim of the procedure is to estimate an unknown parameter ϕ governing a process $\mathcal{I}$. We consider the case where N d-level particles are used as a resource and are divided equally into M bundles of L particles. Each bundle is subjected to the process $\mathcal{I}(\phi, \Phi)$, where Φ is a controllable parameter. After the m^{th} bundle is measured, resulting in an outcome $x_m = \{1, \dots, d^L\}$, the value of Φ_m is adjusted according to the feedback policy ϱ and the history of outcomes leading up to the m^{th} measurement, $\boldsymbol{x}_m = x_1 x_2 \dots x_m$. Here we do not restrict the feedback to be Markovian [52]. The estimate $\widetilde{\phi}$ is inferred from $\Phi_{M+1}(\boldsymbol{x}_M)$ once all the bundles are measured, barring any loss of particles.

The N particles are injected into the measurement scheme as bundles of L particles as

a strategy to increase the information obtained from the measurement [134]. The bundles can be entangled, and so the state is ρ_1, acting on state space $\mathcal{S}\left(\mathscr{H}_d^{\otimes N}\right)$. The particles can be temporally or spatially separated and, hence, can be labelled. In particular, we restrict the input state to those which are symmetric under particle reordering. A permutationally symmetric state is advantageous for an adaptive scheme because the effect of the loss of particle on the imprecision is reduced. In this work, however, we do not include loss.

The m^{th} bundle is injected into a process $\mathcal{I}(\phi, \Phi_m)$, which is a quantum channel, i.e., a completely-positive trace-preserving map [114]. In this formalism, the unitary channel becomes a special case for when no noise or loss is present [47]. The map is represented by a set of Kraus operators $\{\mathcal{K}_i(\phi, \Phi_m)\}$, acting on the subspace $\mathscr{H}_d^{\otimes L}$, such that the condition

$$\sum_i \mathcal{K}_i(\phi, \Phi_m)\mathcal{K}_i^\dagger(\phi, \Phi_m) = \mathbb{1} \tag{4.1}$$

is satisfied.

Upon its exit, the bundle is measured, and the outcome $x_m \in \{0, 1, 2, \ldots, d^L - 1\}$ is obtained. The measurement on one of the bundles is a projection-valued measure (PVM) [117, 116] $X(x_m) = |x_m\rangle \langle x_m|$, where the outcome x_m is obtained with a probability

$$P(x_m|\phi, \Phi_m, \rho_m) = \text{tr}\left(X(x_m)\mathcal{K}_i(\phi, \Phi_m)\rho_m\mathcal{K}_i^\dagger(\phi, \Phi_m)X(x_m)\right), \tag{4.2}$$

omitting the tensor product $\prod_{m' \neq m} \otimes \mathbb{1}^{(m')}$ on the operators.

We create a shorthand for the m^{th} measurement by combining the quantum channel operator and the measurement operator into

$$C_{x_m}(\phi, \Phi_m) = X(x_m)\mathcal{K}(\phi, \Phi_m). \tag{4.3}$$

The unnormalized state after this step is

$$\rho_{m+1} = C_{x_m}(\phi, \Phi_m)\rho_m C_{x_m}^{\dagger}(\phi, \Phi_m). \tag{4.4}$$

As the measurement procedure starts from the input state ρ_1, ρ_m is determined by the previous measurement outcomes, and so we rewrite the notation $P(x_m|\phi, \Phi_m, \rho_m)$ as $P(\boldsymbol{x}_m|\phi, \Phi_m)$ to indicate the probability of obtaining x_m given the previous outcomes being $x_1 \dots x_{m-1}$.

The most general feedback procedure uses the entire history $\boldsymbol{x}_m$ along with the policy ϱ to determine the adjustment $\Phi_m \mapsto \Phi_{m+1}$. Therefore, the controllable parameter is a function $\Phi_{m+1}(\boldsymbol{x}_m)$. We initialize $\Phi_1 \equiv 0$ at the beginning of the procedure. The estimate $\widetilde{\phi}$ is only inferred after all the particles have been measured from the controllable parameter, $\widetilde{\phi} \equiv \Phi_{M+1}(\boldsymbol{x}_M)$, and therefore Φ_{M+1} is an estimator that is an injective function but not necessarily bijective. This function takes a string of discrete random variables as input and hence the range of Φ_{M+1} is also discrete. The estimate $\widetilde{\phi}$ is then a discrete approximation of a continuous parameter $\phi \in [0, 2\pi)$.

4.1.2 Adaptive measurement as a decision-making process

In this subsection, we describe the adaptive measurement procedure in Sec. 4.1.1 as a decision-making process, which is one way of modelling the discrete-time closed-loop control system. For one instance of phase estimation, the procedure can be represented as a decision tree, whose number of leaves reflects the number of possible scenarios and therefore the size of the policy so that every scenario is taken into account. However, this number scales up exponentially, which makes non-Markovian policy computationally expensive to devise.

How the policy determines the imprecision has been explained from two perspectives. An adaptive measurement scheme is thought to achieve the minimum imprecision by approximating the theoretical optimal measurement [76]. In this view, an optimal policy steers the AQEM scheme to approximate this measurement within some error bound. Another expla-

nation of the working of AQEM is to consider the procedure as a Bayesian measurement, where the prior for ϕ is updated according to the outcomes [135]. In this case, a successful policy can be said to maximize the amount of information about ϕ as represented by the prior. The Bayesian model is useful as it provides a framework to generate successful policies by minimizing the width of the prior after each measurement[40].

An AQEM procedure can also be delineated as a decision-making process, where the function $\Phi_{m+1}(\boldsymbol{x}_m)$ makes the decision of what the value of Φ_{m+1} should be based on the outcomes. If the outcome x_m is the only information used, the feedback is Markovian [136, 52], but we do not restrict our formalism to this case. Considering the adaptive measurement scheme as a decision-making process allows us to visualize the procedure as a decision tree, where each branch represents a single-shot procedure obtaining a unique string $\boldsymbol{x}_M$ occurring with probability $P(\boldsymbol{x}_M|\phi, \varrho)$. This decision tree can also represent a policy, making it possible to calculate an important property of a policy, namely the size, which contributes to the difficulty in devising the optimal adaptive procedure.

For an AQEM scheme that utilizes M bundles of L d-level particles, there are M measurements in a single-shot procedure, each with d^L possible outcomes. This process can be represented by a decision tree of depth M with d^L branches stemming from each node (Fig. 4.1b). Each branch, from root to leaf, corresponds to an adaptive measurement producing a unique string of outcomes $\boldsymbol{x}_M$. Given the estimator Φ_{M+1}, this leads to an estimate $\widetilde{\phi}$, which is not necessarily unique.

Assuming the feedback is a deterministic procedure, the probability of obtaining a sequence of $\boldsymbol{x}_M$ is determined by the quantum measurement. Because each outcome is generated with a probability $P(\boldsymbol{x}_m|\phi, \varrho)$, the probability of obtaining $\boldsymbol{x}_M$ is then

$$P(\boldsymbol{x}_M|\phi, \varrho) = \prod_{m=1}^{M} P(\boldsymbol{x}_m|\phi, \Phi_m). \tag{4.5}$$

This probability is also the probability for a single-shot adaptive measurement to deliver an

estimate $\widetilde{\phi} = \Phi_{M+1}(\boldsymbol{x}_M)$.

A decision tree can also represent a policy, providing the rule for mapping $\Phi_m \mapsto \Phi_{m+1}$ according to where on the decision tree the scheme currently resides. Hence, the size of the decision tree; i.e., the number of branches, gives the maximum size of the policy. For a measurement procedure using M bundles of L d-level particles, the number of branches is

$$\sum_{m=1}^{M} (d^L)^m = d^L \frac{(d^N - 1)}{d^L - 1}. \tag{4.6}$$

That is, the maximum size of a non-Markovian policy scales exponentially with N, making a non-Markovian strategy difficult to devise.

4.1.3 Imprecision of an adaptive scheme

In this subsection, we discuss the imprecision determined from the decision tree. Whereas there are many ways of quantifying the imprecision of a measurement scheme [137, 121], all of them can be said to quantify the width of a distribution of the estimate $P(\widetilde{\phi}|\phi, \varrho)$. Using the estimator $\Phi_{M+1}(\boldsymbol{x}_M)$, we can relate the distribution of the estimate to the distribution of the outcomes $P(\boldsymbol{x}_M|\phi, \varrho)$. Minimization of the imprecision is then achieved by optimizing the probabilities associated with the branches of the decision tree.

The imprecision of a single-shot adaptive measurement can be ascertained from multiple estimations of the same value of ϕ (Fig. 4.2). The data set$\{\widetilde{\phi}\}$ forms a distribution that peaks at $\bar{\widetilde{\phi}}$, assuming the distribution is Gaussian. For an unbiased scheme, $\bar{\widetilde{\phi}} = \phi$, but for small M this is not the case because the estimates have discrete values. The bias of a scheme can be quantified by the difference $\phi - \bar{\widetilde{\phi}}$.

We define the imprecision to be the spread of the distribution around $\bar{\widetilde{\phi}}$ regardless of whether the estimate is biased or unbiased. Because of the deterministic relationship between $\boldsymbol{x}_M$ and $\widetilde{\phi}$, we can relate the probability distribution of the outcomes to the probability

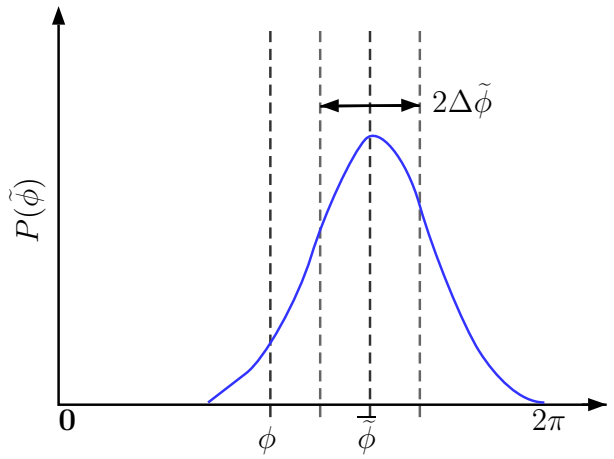


Figure 4.2: Distribution of sampled values of phase estimate $\widetilde{\phi}$. The actual phase shift ϕ, the mean $\overline{\widetilde{\phi}}$ and the points $\overline{\widetilde{\phi}} \pm \Delta\widetilde{\phi}$ one standard deviation $\Delta\widetilde{\phi}$ above and below the mean $\widetilde{\phi}$ are shown on the abscissa. [Reproduced from Ref. [2], Fig. 2]

distribution of the estimate

$$P\left(\widetilde{\phi}|\phi,\varrho\right) = \sum_{\boldsymbol{x}_M \in \Phi_{M+1}^{-1}(\widetilde{\phi})} P\left(\boldsymbol{x}_M|\phi,\varrho\right), \tag{4.7}$$

where Φ_{M+1}^{-1} is the inverse function of Φ_{M+1}. By substituting Eq. 4.5 in Eq. 4.7, we obtain a distribution that explicitly depends on the policy.

From Eq. 4.7, we have shown that in AQEM minimizing the imprecision is done by optimizing the distribution $P(\boldsymbol{x}_M|\phi,\varrho)$. This task is difficult because there are d^N possible values of $\boldsymbol{x}_M$. Furthermore, each string is obtained through a unique state trajectory, and taking all of the possible trajectories into account in order to compute $P(\widetilde{\phi}|\phi,\varrho)$ is resource consuming. As a result, generating a feasible policy is a primary challenge for devising an AQEM scheme. By analyzing the characteristics of the decision-making process, we can make connections between an AQEM scheme to possible methods of generating successful policies.

4.1.4 Generating policies for AQEM

In this section, we discuss the characteristics of a decision-making process derived from an AQEM procedure and the approaches for generating a policy based on these characteristics. We base this discussion on the framework of MDP and POMDP (Sec. 2.2.2). The Markovianity here refers to the outcome of the decision being determined only by the current state and action and not the history of states and actions [52, 89]. Markovian policy, on the other hand, refers to a process that uses only the current state or observable to make a decision [138].

To determine whether an AQEM scheme is Markovian, we first assume that the input state and the quantum dynamics can be known accurately and precisely, and therefore, the state, action, transition probabilities can be calculated. The state s_m at time step m is determined by ϕ, Φ_m, and ρ_m, and the actions that the controller can choose are $\{u_m\}$, which takes $\Phi_m \mapsto \Phi_{m+1}$. If the quantum channel is completely known, then knowing s_m would allow us to calculate the transition probabilities to s_{m+1} from Eq. 4.2 without having to know what the prior states and actions are. Hence, the dynamics of an AQEM procedure can be described by an MDP.

The controller, however, does not have access to this MDP model as the value of ϕ is unknown in an AQEM task, rendering both the states and transition probabilities incalculable. Instead, the controller observes the measurement outcome x_m, which puts an AQEM problem into the category of POMDP rather than MDP. Solving a POMDP is possible if the model of the state evolution is available. One such example is to update a prior of ϕ, which is assumed to be uniformly probable at the beginning, based on x_m using Bayesian inference [40].

If a trusted dynamic model is unavailable, the alternative is to interact with the POMDP in order to gather information about its dynamic [52]. Within this group of techniques, we can divide the methods into ones that evaluate the actions after each time step and ones that evaluate the entire policy after one or more episode of the AQEM task. Reinforcement

learning is an approach in the first category [89] that utilizes step-wise rewards to generate a policy. If step-wise rewards are not available, then policy-based techniques that use a long-term reward are used, and one such technique is direct policy search (2.2.2). When combined with a heuristic optimization algorithm, a policy-search algorithm does not need to construct a model of the POMDP, and the information from the interactions are used only for policy generation.

4.1.5 Cramér-Rao lower bound for AQEM

Whether a quantum metrology scheme attains a quantum-enhanced precision depends on its optimal performance, i.e., the minimum imprecision of the scheme. The lower bound is defined in parameter estimation by the CRLB (Sec. 3.2.2), which gives the minimum variance in terms of Fisher information [139]. In this section, we discuss the lower bound for the imprecision of an AQEM scheme using the distribution derived from our decision-making model. In particular, we consider the case where the bundles of particles are not entangled, and the lower bound is simplified. Because the distribution depends on the policy, the lower bound can only be computed when an optimal policy is given.

In the classical picture, the lower bound to imprecision $\Delta\widetilde{\phi}$ arises from imperfection in the measurement device. Therefore, in principle, the precision of a classical measurement scheme can be improved indefinitely. When the quantum nature of the measurement scheme is taken into account, the uncertainty principle prevents an improvement to the lower bound beyond that allowed by quantum mechanics. To describe the lower bounds for an AQEM scheme, we consider a measurement where there is no imperfection in the measurement scheme and the imprecision is an effect of the quantum-mechanical nature of the input state.

The Cramér-Rao lower bound, which is defined as the lower bound for the variance,

$$(\Delta\widetilde{\phi})^2 = \int_0^{2\pi} \mathrm{d}\widetilde{\phi} P(\widetilde{\phi}|\phi)\,(\phi - \widetilde{\phi})^2\,, \tag{4.8}$$

is derived most generally for multiple-parameter estimation [53, 140]. Here we consider the case where a single parameter is measured, and the estimate is unbiased. Under these conditions, the CRLB becomes

$$\Delta\widetilde{\phi} \geq \frac{1}{\sqrt{F(\phi, \varrho)}}, \tag{4.9}$$

where $F(\phi, \varrho)$ is the Fisher information

$$F^{(M)}(\phi, \varrho) := \sum_{\{\boldsymbol{x}_M\}} P(\boldsymbol{x}_M|\phi, \varrho) \left(\frac{\partial \log P(\boldsymbol{x}_M|\phi, \varrho)}{\partial \phi} \right)^2. \tag{4.10}$$

The Fisher information shows the degree for which the outcomes $\{\boldsymbol{x}_M\}$ generated from ϕ is differentiable from the outcomes generated by $\phi + \mathrm{d}\phi$. The information increases as the two sets of outcomes become distinct from one another.

When all the bundles are entangled, the Fisher information cannot be simplified from Eq. 4.10, although the probability for $\boldsymbol{x}_M$ is a product of the probabilities for x_m. That is because the distribution depends on the history of the outcomes through the back-action on the state. If the bundles are independent of one another, the back-action does not affect the subsequent measurements. In this case, the Fisher information can be simplified to

$$F^{(M)}(\phi, \varrho) = \sum_{m=1}^{M} F^{(1)}(\phi, \Phi_m). \tag{4.11}$$

If $F^{(1)}(\phi, \Phi_m)$ are identical for all $m \in \{1, \ldots, M\}$, the information become $F^{(M)}(\phi, \varrho) = MF^{(1)}(\phi, \Phi_1)$, and the SQL is derived if $MF^{(1)}(\phi, \Phi_m) \propto N$. However, this condition is not satisfied in the presence of a feedback control, which changes the distribution of $P(\boldsymbol{x}_m|\phi, \Phi_m)$ in subsequent measurements. A statement about the lower bound can only be made when the optimal policy is known. Hence we turn to adaptive phase estimation as the toy problem To this end, we turn to adaptive phase estimation as an example of AQEM procedure.

4.2 Adaptive phase estimation as a POMDP

In this section, we cast a QEAPE (Sec. 3.4) into a decision-making process and discuss possible approaches for generating successful policies based on the problem's characteristics. In QEAPE, the N-particle state is split into N bundles of two-state particles, and hence the procedure can be represented as a binary decision tree with the estimates $\tilde{\phi}$ as the leaves of the tree. Due to the randomness of the measurement outcomes, the policy does not control the trajectory that the procedure takes during an instance of estimating ϕ, but rather the probabilities of the trajectories. As the string $\boldsymbol{x}_m$ alone cannot be used to estimate the trajectory probability or determine how close Φ_m is to ϕ, we decide that a policy-generation technique that uses policy-based evaluation rather than action-based evaluation is the most appropriate method for this problem.

4.2.1 QEAPE as a binary decision tree

Here we explain in the properties of the binary decision tree and how the task of QEAPE can be understood in this representation. As the full state description of the decision tree would require knowing ϕ, the tree represents a POMDP with each branch representing a sequence of observables $\boldsymbol{x}_N$ and the action that the controller choose is the tweak to the controllable phase shifter Φ_m, which for the policy-search feedback is $u_m = (-1)^{x_m}\Delta_m$ for $\Delta_m \in [0, 2\pi)$ (Eq. 3.44). In this picture, the task of the feedback control in QEAPE is to manipulate the quantum dynamics such that the path leading to $\tilde{\phi} \approx \phi$ is maximally probable, and the spread of the classification error, i.e., the imprecision, is minimized.

An instance of estimating an unknown ϕ can be viewed as a random walk from the root of the decision tree to the leaf with the path chosen determined by the probabilistic nature of the quantum measurements (Fig. 4.3). For any given ϱ, there are $2^N - 1$ paths, corresponding to every possible sequences of $\boldsymbol{x}_N$. If the measurement outcome is not probabilistic, then an optimal policy should lead to a unique trajectory $\boldsymbol{x}_M$ that results in $\tilde{\phi} \approx \phi$ for any

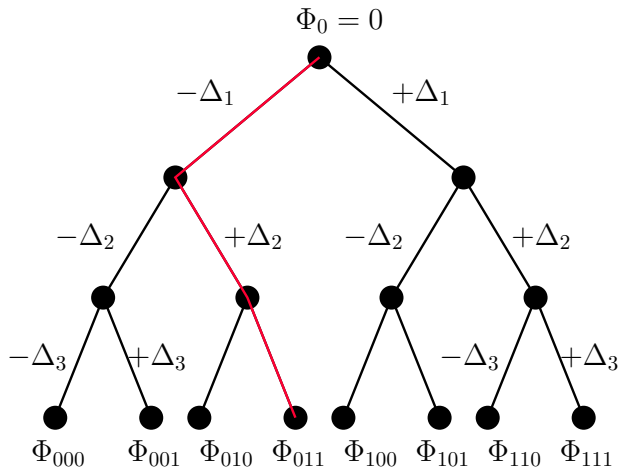


Figure 4.3: An example of a binary decision tree representing a QEAPE using three particles and a logarithmic-search feedback (Eq. 3.44). The red line represents one instance of running the QEAPE scheme, which is randomly simulated based on the quantum dynamics. The initial controllable phase shifter Φ_0 is set to zero and is adjusted by Δ_m or $-\Delta_m$ depending on $x_m \in \{0, 1\}$. Once the particles are used up, the value of Φ_3, which in this example is $\Phi_3 = \Phi_{011}$ is used to infer the estimate $\tilde{\phi}$.

ϕ. However, the quantum nature of the measurement means that even when an optimal policy is used, the trajectory generated by the procedure may not necessarily lead to a close approximation of ϕ. In fact, the trajectory would be random each time, governed by the probability of the particular sequence occurring (Eq. 4.5), and this probability is determined both by ϕ and ϱ. As such, the role of the policy is better stated as maximizing the probability of $\tilde{\phi} \approx \phi$ for $\phi \in [0, 2\pi)$. The imprecision captures the spread of the probability distribution $P(\tilde{\phi}|\phi)$ (Eq. 4.7), which we measure using the Holevo variance (Eq. 3.41) from many sampled trajectories of the same policy.

In this work, we assume that a trusted model of the quantum dynamics is not available due to an addition of phase noise, which we simulate by turning ϕ into a random number in $[0, 2\pi)$ that is sampled from a unimodal distribution with the peak at ϕ_0. Under this constraint, a trial-and-error approach to policy generation is needed as we would need a method of determining the probabilities of the paths with incomplete information. However, the trail-and-error has a drawback as the number of paths scales exponentially with N, and

this number of paths is only for one set of policy parameters, for example a particular $\boldsymbol{\Delta} \in [0, 2\pi)^N$ (Eq. 3.45) whose space contains an infinite number of possible policy vectors. As such, determining the probabilities for each and every combination of $\boldsymbol{x}_N$ and ϱ is impractical. We therefore estimate the spread of the error, formally quantified by the Holevo variance (Eq. 3.41), by having the simulation sample a subset of paths. We do not explicitly specify which path the QEAPE follows but instead leave the path to be randomly chosen based on the quantum dynamics.

4.2.2 Optimizing the QEAPE

Now that we have casted a QEAPE scheme into a POMDP, we explore the possible approaches to minimize the imprecision. We discuss whether an action-based evaluation or a policy-based evaluation is suitable given the limited information we can infer from x_m. As determining the preferred actions for this task is not possible without a model of the quantum dynamics, we choose the approach of policy-based evaluation and uses a black-box optimization algorithm to find a successful policy.

As mentioned in the previous subsection, QEAPE aims to maximize the chance of $\tilde{\phi} \approx \phi$ and minimize the imprecision $\Delta\tilde{\phi}$, which can be intuitively computed using many trials of the same policy ϱ. In other word, the most intuitive approach to evaluating a QEAPE policy is to evaluate the entire policy multiple times. The question is whether we can isolate the actions that leads to achieving minimal imprecision [52]. Action-based evaluation is attractive in the case where the search space is large as the approach exploits a possible structure in the control task to effectively reduce the size of the space that the optimization algorithm focuses on [89]. Action-based evaluation can therefore lead to a faster search using a smaller training set than a policy-based evaluation method [95].

To determine the type of techniques that is suitable for the problem, we turn our attention to the information contained in $\boldsymbol{x}_m$ and whether this information can be used to assign credits that reflect the contribution from each action to minimizing the imprecision. As we use a

measure of imprecision to assign long-term reward to the policy and this imprecision is tied to the probabilities of the outcomes, we consider whether the probabilities $P(x = 1)$ and $P(x = 0)$ can be estimated from $\boldsymbol{x}_m$. If there is no phase difference, i.e. $\phi - \Phi = 0$, then the particles would be detected at the output with a 50:50 chance. A difference in the phases can lead to the change in $P(x_m)$; however, observing $\boldsymbol{x}_m$ alone is insufficient to determine the probabilities because the probabilities would change for each x_m due to the adaptive procedure and the backaction on the quantum state, which is unique for each $\boldsymbol{x}_m$. Therefore, evaluating the actions is not possible with the available information, and we need to wait until the end of the task to determine the quality of ϱ.

One approach that is available for learning an entire policy is the direct policy search [30]. In this approach, the policy is parameterized and an optimization algorithm is used to search for the parameters. However, this approach can be impractical if the policy space is large because the optimization algorithm may require an impractically long time to find a successful solution. For this reason, we do not use a non-Markovian policy, as the size of the policy of would scale exponentially with N (Eq. 4.6). Instead, we adopt the Markovian feedback based on logarithmic-search heuristic in Eq. 3.44, which leads to the estimate being

$$\tilde{\phi} = \sum_{m=1}^{N} (-1)^{x_m} \Delta_m. \tag{4.12}$$

The number of parameters in this policy scales linearly with the size of the search space with N.

For each value of N, we search for a vector of policy parameters $\boldsymbol{\Delta} = (\Delta_1, \Delta_2, \ldots, \Delta_N)$ (Eq. 3.45) that minimizes the imprecision using a DE algorithm, which is a black-box heuristic optimization algorithm [141]. This type of algorithm uses only the fitness function and in this case the sharpness function (Eq. 3.42) to search for a successful policy. The fitness is computed from multiple trials of the same policy, which enables us to optimize independently from the AQEM dynamics [142]. As such, the simulation of an AQEM scheme used in this

work can be replaced with an experimental setup where the dynamics include noise and loss is not fully described. Once $\boldsymbol{\Delta}$ is optimized, the policy can be used to estimate an unknown $\phi \in [0, 2\pi)$ without further training as long as the dynamic of the experimental setup, e.g., the noise model or other system parameters, does not change.

4.3 Differential evolution

In this section, we describe a version of DE that appear in the original paper [143], which we build upon to devise the noise-resistant policy-search algorithm in Chapter 5 and list the known advantages and disadvantages [144]. We include the review here to aid reader's understanding of the results in Sec. 4.4, which are obtained from the algorithm in Chapter 5 without including phase noise.

Differential evolution is a global optimization algorithm that iteratively searches for a feasible solution in a continuous search space [143]. The search process evolves a population of solution candidates using a heuristic inspired by biological evolution. What distinguishes DE from other evolutionary algorithms are the rules for generating and selecting candidates for the subsequent generation of solution candidates [145, 32].

The optimization process begins with initializing a population of candidate solutions. A candidate solution at generation G is represented by a vector

$$\boldsymbol{V}_i(G) = \left(V_i(G)^{(j)}, V_i(G)^{(2)}, \ldots, V_i(G)^{(N)}\right), \tag{4.13}$$

each component corresponding to a parameter in the solution. A population of N_p of these candidates are initialized randomly in the search space of dimension N, and each of the candidates is evaluated for its fitness using a fitness function $S(\boldsymbol{V_i}(G))$, which is related to the quality of the solution.

The algorithm then enters the iterative optimization procedure. For each of the candi-

dates, an offspring,

$$\boldsymbol{D}_i(G) = \left(D_i(G)^{(1)}, D_i(G)^{(2)}, \ldots, D_i(G)^{(N)}\right), \tag{4.14}$$

is then created by combining the parent with randomly selected members of the population. The rule for creating the offspring is

$$D_i(G)^{(j)} = \begin{cases} V_{i,1}(G)^{(j)} + \mathcal{F} \cdot (V_{i,2}(G)^{(j)} - V_{i,3}(G)^{(j)}) & \text{if rand} \leq C_r \\ V_i(G)^{(j)} & \text{otherwise,} \end{cases} \tag{4.15}$$

where $\mathcal{F}$ is a scaling parameter and C_r the crossover rate. The vectors $\boldsymbol{V}_{i,1}(G), \boldsymbol{V}_{i,2}(G), \boldsymbol{V}_{i,3}(G)$ are randomly selected from the population. The fitness of the offspring is then computed and compared to that of the parent. The one with the highest fitness value is selected for the next generation.

The iterative steps are repeated until the termination condition is reached, either by reaching a predetermined number of generations or the fitness of a candidate reaches a threshold, then the candidate with the highest fitness is selected as the solution. The solution delivered by DE is not guaranteed to be optimal, although the convergence of the candidate solutions to a global solution enables a feasible solution to be generated [143, 144]. The convergence, and thus the quality of the final solution, depends on the parameters of the algorithms, which are C_r, $\mathcal{F}$, N_p, and the total number of generations. These parameters are adjusted through trial and error such that a successful solution is obtained within reasonable computational time.

Differential evolution is of interest to us because of two reasons. The first reason is that the optimization only uses the fitness and not its derivative. By ignoring the internal working of the plant, DE can be applied to quantum control problems where the dynamic is intractable or not fully known. Moreover, DE has been found to deliver feasible solutions for high-dimensional search problems, even being able to find a solution in 100-dimension

search space for certain test problems [146].

One concern of using DE is when the fitness function is noisy [147]. Noise in the optimization problem negatively affects any evolutionary algorithm [148]; however, DE has been shown to be more strongly affected in comparison to other population-based heuristic algorithms. Therefore, DE has to be tested in a noisy problem and modification made to make the algorithm noise resistant [149]. We address this issue in Sec. 5.2. For the next section, we turn to the result of applying the algorithm to QEAPE without including phase noise.

4.4 Results and discussion

In this section, we demonstrate the ability of a policy-search algorithm based on DE (Chapter 5) in finding a successful policy for QEAPE without phase noise. We show that the quantum-enhanced imprecision is a result of the quantum correlation between particles that exists in an entangled state and not simply a result of a clever control that DE is able to find. These results are used later as benchmarks for when phase noise is included.

We apply the policy-search algorithm to simulations of QEAPE that utilize the logarithmic-search update rule (Eq. 3.44). In addition to using N-particle sine states (Eq. 3.36), we also simulate QEAPE schemes that uses an N-particle product states,

$$|\psi\rangle_N = |1,0\rangle^{\otimes N}\,, \tag{4.16}$$

as the input states in order to compare the power-law scaling of the imprecisions (Eq. 3.9). We set the DE parameters to $\mathcal{F} = 0.1, C_r = 0.6$ and $N_\mathrm{p} = 50$. The total number of generations before the algorithm accepts a policy is 100.

The QEAPE utilizes $N = \{4, 5, \ldots, 100\}$, and the linear regression of the log-log plot is computed to obtain the power-law scaling (Fig. 4.4). The scaling observed from the non-entangled state is $V_\mathrm{H} \propto N^{-1.071}$, which closely adheres to the SQL. A slight improvement can be attributed to the limit of $N = 100$, which is far from $N \to \infty$ assumed for SQL.

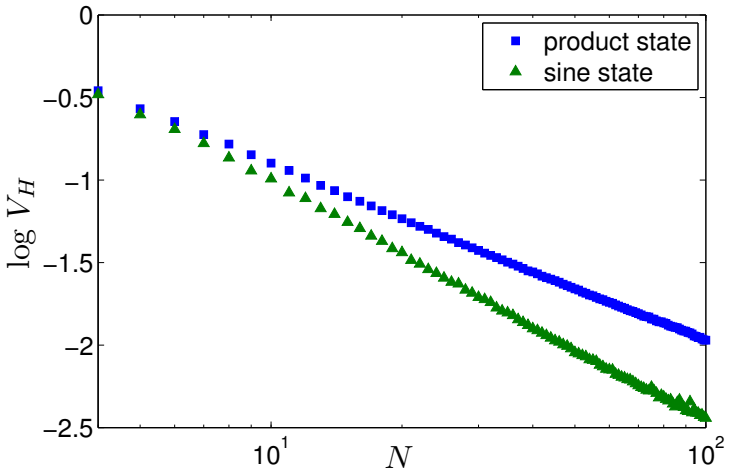


Figure 4.4: Logarithm of Holevo variances generated using a non-entangled state (blue ■) and an entangled state (green ▲). [Reproduced from Ref. [2], Fig. 3]

We find that using the sine state leads to a scaling $V_{\mathrm{H}} \propto N^{-1.421}$, which exceeds the SQL. Hence, entanglement between the particles is a crucial part that leads to AQEM being able to attain quantum-enhanced precision.

4.5 Conclusion

In this chapter, we formalize the description of an AQEM scheme that divides its resource equally over M measurements and model the procedure as a decision-making process. Using this framework, we are able to derive the imprecision and the CRLB that take into account the feedback procedure. We also analyze the characteristics of the decision-making process based on MDP and POMDP framework and decide on the appropriate technique for devising control policies. We use a policy-search algorithm to find policies that minimize the imprecision and attain power-law scaling up to $N = 100$ to an adaptive interferometric phase estimation scheme and is able to show that the quantum-enhanced precision is attained when entanglement is involved. Otherwise, the optimal scaling approaches the SQL.

The results we have obtained are used as benchmarks in the next part of the thesis where phase noise is included in the simulation of the interferometer. Phase noise emulates the

imperfections that are present in a physical interferometer, which increases the noise in the objective function and makes it difficult for the search algorithm to find a successful policy. In the next chapter, we present how the policy-search algorithm is devised to be resistant to phase noise.

Chapter 5

Scalable and Noise-Resistant Policy-Search Algorithm

In this chapter, we describe the policy-search algorithm devised to be noise-resistant and capable of delivering successful policies for a high number of particles. We show that the algorithm can deliver policies for QEAPE that surpasses the SQL for up to 100 particles with low-level normal noise and loss. In principle, the algorithm can deliver successful policies up to any number of particles as long as the CPU time is available. However, for the purpose of demonstrating the algorithm's efficacy, we do not push beyond $N = 100$ where the rounding error in the numerical interferes with the simulation. Furthermore, we explain the implementation of the algorithm on a high-performance computing cluster, where we use parallelized resources to reduce the runtime.

5.1 Introduction

The policy-search algorithms that have previously been designed for adaptive phase estimation fail to deliver successful policies consistently beyond 90 particles without noise [35] and 45 particles with noise [34]. Beyond these points, the imprecision shows a stagnation, which is a result of the optimization algorithm not given sufficient computational resource to find

a successful policy. Therefore, a new search algorithm needs to be devised for QEAPE.

As the representation of the policy, the fitness function, and the training set are decided based on the task of QEAPE, the focus is on designing the optimization algorithm for the problem. Previous works show that the optimization landscape based on the sharpness function is nonconvex [90], and therefore a global optimization strategy is selected. An optimization algorithm searches for a solution in a search space defined by the parameters of the policy, and since the size of the policy grows with the number of particles, the algorithm must be able to generate successful policies for high-dimensional search space. A global optimization algorithm that has been shown to be able to deliver successful policy for a search space as high as 100 is differential evolution (DE) [143, 144].

However, using DE alone is not sufficient to deliver successful policies consistently for more than 93 particles within a reasonable time. Another heuristic is used to reduce the time cost by identifying the neighbourhood where the solution that breaks SQL exists in high-dimensional problems. The centre of the neighbourhood is identified by constructing an N-dimensional vector where the first $N-1$ dimensions in the search space come from the optimized policy of the $(N-1)$-dimensional problem. The N^{th} position is chosen to the be same as the $(N-1)^{\text{th}}$ dimension, and the search algorithm then randomly tries out the policies, as defined by DE, around this point. This heuristic has allowed the number of iterations required to find a successful policy to go down to 100 iterations where otherwise the number of iterations would have scaled with N [34].

Although DE has an advantage in terms of high-dimensional problems, DE has a known disadvantage when the optimization problem is noisy [149]. To obtain successful policies for the noisy, high-dimensional phase estimation problem, we create a variant of DE that is robust against phase noise by using the mean sharpness instead of the sharpness function. And to push up to 100 particles, we include accept-reject criterion that switch from accepting policies after 100 iterations to accepting policies only after V_{H} falls within an acceptable distance away from a projected imprecision in the logarithmic plot (Appendix A).

5.2 Noise-resistant global optimization heuristics

Differential evolution is able to find feasible solutions in high-dimensional search spaces for a set of test problems [146] and for adaptive phase estimation [35]. However, when DE is employed for the problem of noisy phase estimation for N up to 100, we observe that DE does not perform as well as PSO and, in fact, fails to deliver better than SQL scaling. To devise a noise-resistant global optimization algorithm for our scheme, we use the mean value $\bar{S}$ instead of S to determine the performance of a policy. This strategy is one of the many strategies proposed in the literature to create noise-resistant DE and other evolutionary algorithms [149, 150] and is found to work best for our problem.

The principle behind the use of the mean fitness value is as follows: if noise is added to the fitness function, the process of averaging recovers the true fitness value. The optimization using this value is, therefore, a close approximation to the noiseless optimization. The major drawback of this approach is that computing the fitness function multiple times makes the procedure computationally expensive. Therefore, determining the smallest sample size of $\{S\}$ necessary to recover S is crucial. To this end, we employ the heuristic applied to PSO in the previous work [34]. The method updates $\bar{S}$ by computing one new sample of S every iteration until a better offspring is generated.

The sample size for computing S is then determined by the probability for DE to generate an offspring that is better than the parent. This probability decreases as the candidates converge on the optimal value. As a result, the sample size grows automatically as the optimization progresses. The computational resources are allocated towards candidates that are close to optimal, which is a favourable strategy as the differences between fitness values are dominated by noise in this case. Large sample size enables accurate estimations of the candidates' fitness values.

In the case of adaptive phase estimation, the phase noise does not add an additive noise in S. Rather, the phase noise increases the imprecision and in turn increases S. As such, the mean value computed from J samples does not converge on the fitness value of the noiseless

case but rather making the estimate of S more precise.s For an approximate of low noise, computing the mean sharpness is approximately the same as computing S using JK number of samples.

5.3 Improving scalability

In this section, we explain the accept-reject criteria and how this technique leads to DE delivering successful policies up to $N = 100$. Although DE can generate successful policies for $N > 45$, which is the limitation observed when PSO is used [35], the variances also display stagnation when $N > 90$. To generate policies from a search space that scales up to 100 dimensions, we implement a set of heuristics and criteria to the noise-resistant DE to ensure that only successful policies are accepted.

The stagnation is a manifestation of the algorithm not being able to converge to a successful solution in the time limit imposed. Previously the algorithm accepted a policy after a fixed number of iterations regardless of whether the population converges. However, as the dimension of the search space increases, so does the time for the population to converge. Eventually, the algorithm fails to deliver a policy that passes the test. We change the criterion for accepting a policy from a fixed number of iterations to only if V_{H} is within a distance corresponding to a confidence interval of 0.98 from the inverse power-law line. Thus, we guarantee that the policy from our algorithm always delivers a power-law scaling better than SQL.

The acceptable error δ_y for $N > 93$ is calculated from the statistics of V_{H} (Eq. 3.41). The Holevo variance V_{H} are collected from $N = \{4, 5, \ldots, 93\}$, in which we accept the policies after a fixed number of iterations. A linear equation is determined from $\{y_i\} = \{\log V_{\mathrm{H}}(N)\}$ and $\{x_i\} = \{\log N\}$, and is used to predict the next data point y'. The acceptable error from this predicted value is calculated using the previously stored data and the best value

of $V_{\rm H}$ at iteration G from a statistical formula, namely [151, 152]

$$\delta_y = t^*_{v-2}\sqrt{\frac{\sum_{i=1}^{v}(y'_i - y_i)^2}{v-2}\left(\frac{1}{v} + \frac{(x'-\bar{x})^2}{\sum_{i=1}^{v}(x_i-\bar{x})^2}\right)}, \tag{5.1}$$

where v is the number of data points, $x' = \log N$ for which the error is calculated, and $\bar{x}$ is the average of $\{x_i\}$. The value t^*_{v-2} is the quantile of the Student's t distribution, which is a unimodal symmetric distribution, for $v-2$ data points, We approximate using a normal distribution, as a Student's t distribution becomes a normal distribution for large v. The policy is accepted if $|\log V_{\rm H}(N) - y'| \leq \delta_y$, or the optimization continues.

The noise-resistant DE variant, including the accept-reject criteria, works as follows.

Algorithm 1 Noise-resistant DE

Step 1 Initialize the population of size $N_{\rm p}$ randomly.
Step 2 Evaluate the fitness function for each candidate *twice*, and store the mean fitness value and the sample size.
Step 3 Generate a donor $\boldsymbol{D}_i(G)$ for each of candidate $\boldsymbol{V}_i(G)$, where G is the iterative time step, from three other candidates $\{\boldsymbol{V}_{i,1}(G), \boldsymbol{V}_{i,2}(G), \boldsymbol{V}_{i,3}(G)\}$ chosen randomly. For element j of the donor $D_i(G)^{(j)}$,

$$D_i(G)^{(j)} = \begin{cases} V^{(j)}_{i,1}(G) + \mathcal{F}\cdot(V^{(j)}_{i,2}(G) - V^{(j)}_{i,3}(G)) & \text{if } r \leq C_r, \\ V_i(G)^{(j)} & \text{else,} \end{cases} \tag{5.2}$$

where $\mathcal{F}$ is the mutation rate, $C_r \in [0,1]$ is the crossover rate, and $r \in [0,1]$ is a random number.
Step 4 Evaluate the mean fitness value for each of the new candidates from two samples.

Step 5 Compare new and the old candidate using the mean fitness value

$$\boldsymbol{V}_i(G+1) = \begin{cases} \boldsymbol{D}_i(G) & \text{if } \bar{S}(\boldsymbol{D}_i(G)) > \bar{S}(\boldsymbol{V}_i(G)), \\ \boldsymbol{V}_i(G) & \text{else,} \end{cases} \tag{5.3}$$

Step 6 Evaluate the fitness function once, and update the mean fitness value and the sample size.
Step 7 Repeat step 3 to 6 until the criterion to terminate the algorithm is met.
Step 8 Compute the fitness value of the entire population for 10 times before selecting the candidate with the highest mean fitness value as the solution.

5.4 Selecting loss-tolerant policies

In the process of developing and testing the policy-search algorithm, we discover that DE generates two types of policies, only one of which is tolerant to photon loss. Therefore, we implement a criterion to automatically select a loss-tolerant policy based on the sampled bias of the estimates. Because $\theta^{(k)} = \tilde{\phi}^{(k)} - \phi_0^{(k)}$ is a periodic variable, we uses the mean direction $\bar{\theta}$ [154] as the measure of the bias and selects only the policies that deliver $\bar{\theta} \approx 0$, which have been observed to be loss-tolerant. Otherwise, the policy is rejected and the optimization is restarted from the beginning.

Once photon loss $\eta > 0.05$ is included in the simulation of the QEAPE, we observe two responses from DE-designed policies. In one case, the policies are able to deliver imprecision scalings that exceed SQL, whereas in the other case, the imprecisions fail to decrease with N. In examining these policies, we discover that DE-designed policies can be classified by the trend of Δ_m with m. The loss-tolerant policies are the ones whose $\lim_{m\to N} \Delta_m = 0$ asymptotically from 2π or $\lim_{m\to N} \Delta_m = 2\pi$ asymptotically from 0, the latter being equivalent to the phase adjustment vector (Eq. 3.45) having negative elements in $[0, 2\pi)$. In both of these cases, the control parameter Φ_m is adjusted in an increasingly smaller increment as the measurement progresses, and the distribution of $\tilde{\phi}$ governed by these policies is unimodal. The policies that are sensitive to photon loss, on the other hand, are the ones whose $\lim_{m\to N} \Delta_m = \pi$ with Δ_m oscillating around π. As such, the distribution of $\tilde{\phi}$ is bimodal with the distance between two modes being π. We aim to reject the loss-sensitive policies to ensure that the QEAPE procedure designed using our policy-search algorithm to be able to tolerate photon loss.

A property that can be used to differentiate between loss-tolerant and loss-sensitive policies is the bias of $\tilde{\phi}$, which has been observed in the visualization of $\tilde{\phi}$ distribution. Because the phase shifts are wrapped in the domain $[0, 2\pi)$, the bias of $\tilde{\phi}$ is measured using mean

direction [154], which is a mean of a circular random variable,

$$\bar{\theta} = \arctan\left(\sum_{k=1}^{K} \sin\left(\theta^{(k)}\right), \sum_{k=1}^{K} \cos\left(\theta^{(k)}\right)\right). \tag{5.4}$$

Loss-tolerant policies have been observed to deliver unbiased estimates, i.e., $\bar{\theta} \approx 0$. Loss-sensitive policies, however, deliver estimates with bias of either 0 or π depending on whether N is odd or even.

We devise the criterion to select loss-tolerant policies by computing the bias for $N = 4$ and $N = 5$. If $\bar{\theta}$ falls within $0 \pm \sqrt{V_{\mathrm{H}}}$ for both N, the algorithm accepts the policy and continues to search for a policy for $N = 6$ and beyond. If not, the policy-search algorithm resets N to $N = 4$ and restarts the optimization process. Because the level of error we tolerate for $\bar{\theta}$ depends on V_{H}, which increases with the level of phase noise, the criterion fails to eliminate loss-sensitive policies in a highly noisy environment.

In general, the most precise estimator is not necessarily the same as an unbiased one, and DE being able to find one that can deliver both is a consequence of the generalized-logarithm-search structure as the sharpness, which we use as the fitness, is independent of bias. We do not anticipate this observation to hold true for other policy structures or that the estimator's bias is fundamentally connected to the policy's ability to tolerate photon loss.

5.5 Implementing algorithm on a high-performance computing cluster

The computational complexity of the algorithm is polynomial [35], but it has a high degree, and therefore it is important to identify the performance critical parts of the implementation. Two strategies have been used to decrease runtime of the algorithm. Firstly, we distribute the solution candidates on a high-performance computing cluster so that the process of com-

puting the fitness values is parallelized. Secondly, we identify that over 90% of the execution time is spent on generating random numbers one by one, and so we explore strategies for distributing the random number generation. Once generated, the numbers are stored in tables that each candidate can access independently.

In order to distribute the candidates over the cluster, we utilize a Message Passing Interface (MPI) library [153], which enables us to parallelize the calculation of the fitness function onto multiple CPUs in the same cluster. The advantage is that the MPI library specifies the communication between the CPUs and the memories during compilation of the code, and so the same code can be implemented on any type of clusters that support MPI. To minimize the runtime, we implement the code such that there is one candidate per CPUs as the time used to communicate between CPUs is then insignificant compared to the time used to compute the fitness function.

A large part of time cost of the fitness comes from random number generation, which is primarily used in estimating the Holevo variance as the computation involves simulations of the adaptive measurement procedure. Generating random numbers as they are needed is not efficient on contemporary hardware. The operations can be vectorized to use the single-instruction multiple-data architectures of the central and the graphics processing units. Abstracting the random number generation routines and introducing a buffer, we are able to vectorize the respective operations. We study two approaches: one relies on the CPU, using the Intel Vector Statistical Library (VSL), the other on graphics processing units. Eventually, the VSL-based vectorized solution proves to be more scalable. In both cases, the random numbers are stored in tables to be used by candidates.

5.6 Results and discussions

By applying the method of adding noise-resistance to DE, we are able the obtain policies that deliver the scaling of $V_H \propto N^{-1.421}$ when the width of the normal distribution is 0.2

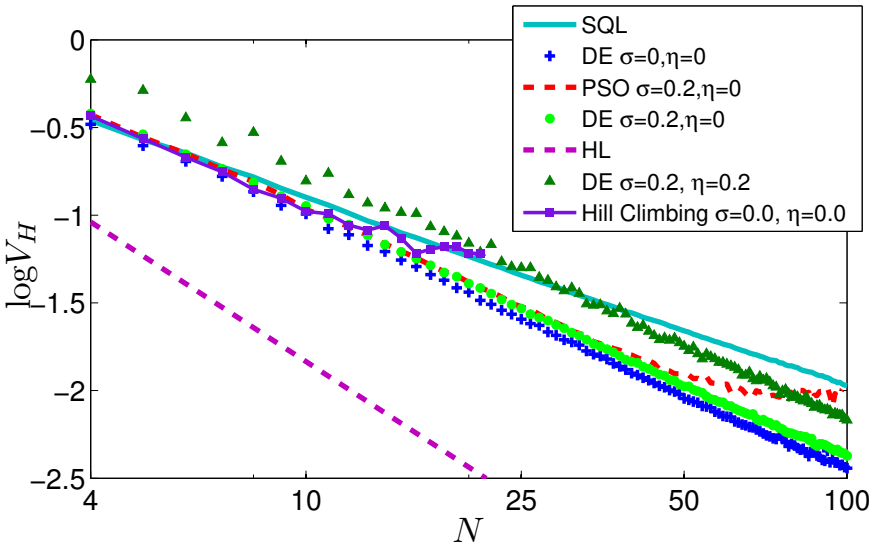


Figure 5.1: The logarithm of Holevo variance from adaptive interferometric-phase estimation. The interferometer includes small phase noise of width σ and loss rate η. Three algorithms are used to generate the feedback policy DE, PSO, and stochastic hill-climbing. This image is a rescaled version of Figure 1 in Ref. [1]. [Reproduced from Ref. [3], Fig. 3]

rad, and the probability of losing a photon is 0.2 are included. This result shows a scaling exceeding N^{-1} expected from SQL, which is given for the ideal interferometer as a benchmark in Fig. 5.1. The SQL data is generated using a non-entangled N-particle state (Eq. 4.16). The HL shown in Fig. 5.1 is an extrapolation using the intercept from the SQL data and is included for the purpose of providing a possible benchmark for the scheme.

Although both the SQL and the HL are reported in the literature for the mean-squared error $\Delta\widetilde{\phi}$ [8], we use the same benchmark for Holevo variance V_{H}. This follows from the approximation of V_{H} at low error $\left|\phi_k - \widetilde{\phi}_k\right| \ll 1$. Under this condition, the Sharpness in Eq. (3.42) is approximated by a series expansion, and through this approximation, V_{H} is found to be the mean-squared error.

The accept-reject criterion applied to $N > 93$ enables the scheme to show the attain the power-law up to $N = 100$ (Figure 5.1). The limitation at 100 photons is due to the computational time and the rounding error in the generation of the large multiparticle entangled state. The time required to find a policy under the accept-reject criteria from 94 to 100 photons are between 1.5 to 3 hours per data point.

Policies that are found using stochastic hill climbing break down at 20 photons even for ideal phase estimation. The noise-resistant PSO shows the breakdown at 45 photons, consistent with the previous result [35]. We did not apply the accept-reject criteria to PSO as the computational time would have exceeded the time used by DE at the same number of N and hence not considered worth the investment.

5.7 Conclusion

In this chapter, we describe the policy-search algorithm adaptive phase estimation that can generate successful policies for when noise and loss are included. We attain quantum-enhanced precision using a noise-resistant variant of DE and accept-reject criteria, which enables the algorithm to generate successful policies up to 100 particles. In principle, the algorithm can generate policies up to any number of particles and the runtime will become a limiting factor once N reaches a certain number.

The methods we employed do not require explicit knowledge of the system's dynamics, although the convexity of the fitness functions, the dimension of the problems and the presence of noise have to be taken into account in order to generate a feasible policy. We minimize the runtime of the algorithms by distributing the candidates on a high-performance cluster and vectorizing the random number generation and employing GPUs and VSL. This technique mostly affects the simulation of the quantum system as the simulations are the most time- and resource-consuming components of the current algorithms. With this algorithm, we now have an alternative, data-driven approach for policy generation that can be compared to the model-based Bayesian feedback.

Chapter 6

Comparing AQEM Policies

In this chapter, we commence with the comparison of model-based and data-driven control. We use the problem of quantum-enhanced adaptive phase estimation (Sec. 3.4) as the test problem and devise criteria for comparing the policies designed using Bayesian inference (Sec. 3.4.2), which requires a model of the quantum dynamics, and using the DE algorithm, which is an example of a model-free control design procedure (Chapter 5). We develop a test of the QEAPE's robustness against phase noise of unknown properties and assess the computational resources used to design and implement the quantum control policies. We also compare the asymptotic power-law scalings for when the dynamic model used in the Bayesian feedback matches with quantum dynamics in the simulation.

6.1 Introduction

AQEM performance critically depends on the choice of policy ϱ [13], which can be obtained by optimizing a known mathematical model [123, 124, 125] or by a trial-and-error approach [90, 35, 3]. Whereas policies from these methods are resistant to known noise models [123], whether they are robust against unknown noise is yet an unstudied but critical property of a QEM scheme as noise can destroy the entanglement advantage and restore the SQL [155, 156]. Our aim is to test the robustness of AQEM policies in the presence of noise with unknown

properties.

Our test focuses on quantum-enhanced adaptive phase estimation (QEAPE), whose policies have been devised using Bayesian techniques [39, 40] and by searching the policy space using an evolutionary algorithm (EA) [90, 35, 3]. The Bayesian technique computes feedback based on a trusted, noiseless quantum model, whereas EAs [32] devise policies for feedback based on the fitness of policies obtained through trial and error. This evolution is, by design, ignorant of the quantum-dynamical nature but employs heuristics to shrink the search space. Here both methods are applied to QEAPE including phase noise, which could arise from path-length fluctuation in the interferometer [157, 158].

Typically, noise is assumed to be normal as a result of the central-limit theorem [159]. The periodicity of the phase makes the normal distribution problematic unless the noise is small compared to 2π radians, which we assume here; technically, we would use the wrapped-up normal distribution [57]. As our aim is to test robustness for unknown noise, we consider three other noise distributions for our test: random telegraph [58], skew-normal [59] and log-normal [60] noise. The random-telegraph noise simulates a discrete noise process. Skew-normal and log-normal distributions represent asymmetric noise, which serve as distinct generalizations of the normal distribution. Both distributions are used to simulate noise in detectors and electronics [160, 161, 162].

We first test the robustness of the policy-search algorithm by including the phase noise models with low variance and skewness during the training stage. The purpose for this part of our work is two folds. We want to confirm that the policy-search approach can indeed adapt the policy to deliver quantum-enhanced precision as the data-driven control promises. We also use this preliminary study to determine the contribution of skewness to the imprecision scaling so to decide on the parameters of the robustness test between the Bayesian feedback and the EA-designed policy.

For AQEM, we seek an efficient procedure that beats the SQL, and we choose ϱ that requires the least resource to run. We assess the policy-generating procedure according to

the complexity of its time cost [61], which is evaluated by the scaling in the number of operations with the number of particles N. Here we consider two policy-design procedures, namely, a policy-search method based on differential evolution (DE) [143], which is a variant of evolutionary algorithms, and a method based on Bayesian inference, resulting in one policy designed by each method. To determine which policy is superior, we compare the complexity in space and time cost [62]. Thus, we are able to assess and compare the costs for generating policies and determine the best policy.

Through our analysis, we find that both Bayesian-feedback and DE-designed policies are robust in the face of unknown noise. Specifically, the Bayesian method yields imprecision that approaches HL and outperforms DE-designed policies for most noise models. This performance superiority is due to the Bayesian method effectively memorizing the measurement history through the agent's complete knowledge of the quantum state. Storing the entire model in the agent yields better imprecision scaling but incurs higher space and operational time costs compared to the DE-derived policy.

6.2 Background

In this section, we describe the noise models chosen for the robustness test and the basics of the regression analysis that we use to determine the asymptotic imprecision scaling. Previously, only the normal-distribution noise has been considered when devising the noise-resistant DE (Chapter 5). Here we included other noise models, including a discrete distribution and skewed distributions to the test. The asymptotic imprecision scaling has only been determined by assuming that the imprecision is represented by a power-law line, which is no longer true for low N when the noise level is increased. Therefore, the regression procedure must include a method to determine where the power-law line starts as well as the method for computing the scaling.

6.2.1 Noisy phase estimation

In this section, we explain the choices of the phase-noise models for the robustness test. This noise is simulated by turning ϕ into a random variable that has a unimodal probability distribution with the peak at ϕ_0 (Alg. 2). The mode ϕ_0 is assumed to be the unknown parameter to be estimated. For the test, ϕ is sampled from one of the following distributions: a normal, a three-stage random telegraph, a skew-normal, or a log-normal distribution. We review the relationship between the noise parameters and the variance and skewness as quantified by the second and third central moments [163] as we use both in selecting the parameters for the robustness test and the variance in particular to define the noise level.

Algorithm 2 Adaptive phase estimation

Input: N, ρ_1, ϱ_N, ϕ_0
Output: $\widetilde{\phi}$
  Initialization: $\Phi_1 \leftarrow 0$, $m \leftarrow 1$
  for $m \leq N$ **do**
    $\phi \leftarrow \text{RandomNumber}(\phi_0)$ //Add noise to the phase shift.
    x_m, $\rho_{m+1} \leftarrow \text{Measure}(\rho_m, \phi, \Phi_m)$ //Update the state ρ_m depending on $\phi - \Phi_m$. //Compute probabilities of $x_m \in \{0, 1\}$ from //ρ_{m+1}, and choose x_m randomly based on //these probabilities.
    if $x_m = 0$ **then**
      $\Phi_{m+1} \leftarrow \Phi_m - \Delta_m$ //Action chosen if $x_m = 0$.
    else
      $\Phi_{m+1} \leftarrow \Phi_m + \Delta_m$ //Action chosen if $x_m = 1$.
    end if
    $m \leftarrow m + 1$
  end for
  return Φ_{N+1}

Normal-distribution noise

Normal-distribution noise is important for testing robustness of the search algorithm because the normal distribution is especially prominent due to the central limit theorem, which states that the average of a random variable has a normal distribution [159]. Due to the prevalence of the normal distribution, assuming normal-distribution noise model is common [164]. The

normal distribution,

$$p(\phi) = \frac{\mathrm{e}^{\frac{(\phi-\mu)^2}{\sigma^2}}}{\sqrt{2\pi}\sigma}, \tag{6.1}$$

is parametrized by the mean μ and standard deviation σ. As this distribution is symmetric, skewness γ is identically zero and thus the mode is at μ, and the variance is $V = \sigma^2$. In our simulations, we set $\mu \equiv \phi_0$ so the only free parameter is σ, which is bounded above by $\sigma < \pi$ as otherwise the width would exceed the domain of ϕ. We restrict $\sigma \leq 2$ as higher values of σ would be uninteresting as they would correspond to almost completely noisy measurements.

Random-telegraph noise

Random-telegraph noise [58] is a noise that is sampled from a discrete distribution that, for each time step, randomly switches between two values, one being the correct and the other an erroneous value. Whereas this noise is most relevant to digital electronics, as it simulates a bit-flip error, this noise simulates other forms of digitized noise, such as salt-and-pepper noise in image processing [165].

We modify two-stage random-telegraph noise to have three stages,

$$p(\phi) = \begin{cases} 1 - p_\mathrm{s}, & \phi = \phi_0, \\ \frac{p_\mathrm{s}}{2}, & \phi = \phi_0 \pm \delta. \end{cases} \tag{6.2}$$

The probability of switching to an erroneous value is p_s, and δ is the distance between the true and erroneous values leading to

$$V = p_\mathrm{s}\delta^2, \ \gamma \equiv 0 \tag{6.3}$$

with the last relation following from the symmetry of the distribution.

Unimodality of the distribution implies that $p_\mathrm{s} < 2/3$. Furthermore, we restrict $\delta < \pi$ so that the distance between the two side peaks are less than 2π. To comply with both

constraints and being able to raise the noise level to at least $V = 3$ so the result is comparable to other distributions, we fix $p_s = 1/2$ for the test and vary only δ.

Skew-normal-distribution noise

The skew-normal distribution [59] is modified from a normal distribution by multiplying with a function whose skewness parameter is α, and so skew-normal noise is a class of noise that includes normal-distribution noise as a limiting case. Although this distribution is not widely used as a noise model, this probability distribution arises in simulations of noise for filters and detectors [160, 162].

The skew-normal distribution is

$$p(\phi) = \frac{e^{-\frac{(\phi-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}\left[1 + \text{erf}\left(\frac{\alpha}{\sqrt{2}\sigma}(\phi-\mu)\right)\right] \tag{6.4}$$

for erf() the error function [166]. Skewness of the distribution is

$$\gamma = \frac{4-\pi}{2}\frac{2\beta}{\pi - 2\beta},\ \beta = \frac{\alpha^2}{1+\alpha^2}, \tag{6.5}$$

and the variance is

$$V = \sigma^2\left(1 - \frac{2\beta}{\pi}\right) \tag{6.6}$$

The mode, however, does not have a closed form although it remains close to μ as α/σ increases. For the simulation, we assume the mode is μ.

Log-normal-distribution noise

Log-normal distribution [60] is a heavy-tailed skewed distribution that provides another approach to generalizing the normal distribution and is employed in the study of networks [167, 168] and electronics [161]. In this case, the logarithm of the random variable is

said to have a normal distribution, leading to the distribution

$$p(\phi) = \frac{\mathrm{e}^{-\frac{(\log\phi-\mu')^2}{2\sigma'^2}}}{\sqrt{2\pi}\sigma'\phi} \tag{6.7}$$

with mode and variance

$$\phi_0 = \mathrm{e}^{\mu'-\sigma'^2},\ V = \left(\mathrm{e}^{\sigma'^2}-1\right)\mathrm{e}^{2\mu'+\sigma'^2}, \tag{6.8}$$

respectively, and skewness

$$\gamma = \left(\mathrm{e}^{\sigma'^2}+2\right)\sqrt{\mathrm{e}^{\sigma'^2}-1}.$$

As this distribution is defined for $\phi \in (0,\infty)$, we first generate a random number within the compact phase domain given μ' and σ' and then apply the shift

$$\phi \mapsto \phi + \phi_0 - \mathrm{e}^{\mu'-\sigma'^2} \tag{6.9}$$

so that the mode of the distribution is centred at ϕ_0 (Eq. 6.8).

6.2.2 Regression analysis

The imprecision $\Delta\tilde{\phi}$ and N are asymptotically power-law related (Eq. 3.9). However, when the system is noisy, this relationship fails for low N, with the actual bound on N depending on the noise model. We employ regression analysis to select the subset of V_H at high N that scales as $N^{-\wp}$ and estimate the corresponding $\wp$ by building piecewise functions and selecting the best candidate to represent the data. In this subsection, we explain our regression-analysis procedure for fitting a model given a set of data.

Fitting the model

Regression analysis aims to determine the mathematical relationship between the dependent (V_{H} here) and independent variables (here N) [169]. The process of building this mathematical model begins with selecting a function $f(N)$ based on the knowledge of the mechanism and observations of the trends [170]. The function is only a best guess as the discerned trend could be subjective.

After a function is selected, the function is then fitted to the data by finding the function parameters that minimize the error between the predicted V_{H} (denoted as $f(N)$) and the data V_{H} [171]. The method we employ is the least-squares estimation [172], used in the linear regression to calculate the parameters by constraining the gradients to zero and solving the resulting system of linear equations. We choose this method as we fit linear and piecewise linear equations to log-log plot of V_{H} and N.

Consistency of the fit

As the fitted function is only an educated guess, the fitting result must be examined for inconsistencies with respect to the model's assumptions [171]. An alternative function can then be proposed, fitted, and compared to the previous function in order to find one that best represents the data. Deciding on the best model from the set is done using statistical criteria that either estimate the goodness of the fit to the data or between two models fitted to the same data [173]. The model that is consistently shown to fit well according to each of the criteria is then selected to represent the data. Below are common linear-regression criteria used in this work.

Adjusted R-squared. The coefficient of determination

$$R^2 = 1 - \frac{\sum\limits_{N} \left(V_{\mathrm{H}}^{(N)} - f(N) \right)^2}{\sum\limits_{N} \left(V_{\mathrm{H}}^{(N)} - \bar{V}_{\mathrm{H}} \right)}, \tag{6.10}$$

is a standard measure of the goodness of fit [174]. However, the R^2 does not take into account the number of model parameters b and the number of data v, and so using this measure alone for model selection can lead to a model that over fits, i.e., a model that fits existing data very well but is unable to predict future data with the same precision. Adjusted R^2

$$\overline{R^2} = R^2 - \frac{b}{v-b-1}(1-R^2), \tag{6.11}$$

is created with a trade-off between high R^2 and the simplicity of the model, quantified by b. Both measures have a maximum value of 1.

Corrected Akaike Information Criterion. The Akaike information criterion (AIC) quantifies information lost due to the discrepancy between the model function and the true function $g(N)$ [175],

$$\mathrm{AIC} := b \ln \left(\frac{\sum_N \left(V_{\mathrm{H}}^{(N)} - f(N) \right)^2}{v} \right) + 2b. \tag{6.12}$$

Although the AIC already takes b and v into account, AIC has been known to overfit when v is small, and so corrected AIC,

$$\mathrm{AIC_c} := \mathrm{AIC} + \frac{2(b+2)(b+3)}{v-b-3}, \tag{6.13}$$

is introduce to increase penalty for models with large b to avoid overfitting.

***F*-test** assesses a full model (maximum b), as the null hypothesis, against a reduced model (reduction from the full model) as the alternative hypothesis [174, 170]. The F-value is computed from

$$F' := \frac{\sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{r}})^2 - \sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{f}})^2}{\sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{f}})^2} \frac{(v - b_{\mathrm{f}} - 1)}{(b_{\mathrm{f}} + 1 - b_{\mathrm{r}})}, \tag{6.14}$$

where the subscript f indicates the full model and r the reduced model. The usual assumption is that

$$\sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{f}})^2 < \sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{r}})^2, \tag{6.15}$$

so the F value is always positive.

In the F-test, the F-value is compared to the critical value computed from an F-distribution, which is a single-sided distribution. If the F-value is smaller than the critical value, then the reduced model is considered to be just as good as the full model in explaining the data, and the reduced model is selected to represent the data.

Mallows's C_p,

$$C_p := \frac{\sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{r}})^2}{\sum_N (V_{\mathrm{H}}^{(N)} - f(N)_{\mathrm{f}})^2 / v} - v + 2b + 2, \tag{6.16}$$

estimates the mean-square prediction error [170, 173] and is used compare a reduced model to the full model. If the model has all the necessary parameters, C_p is expected to take the value $C_p \approx b + 1$, and so a model that deliver this value is selected in the model selection process. This condition, however, may not be reached, and a model that minimized C_p is also considered.

6.3 Approach

In this section, we explore the skewness's contribution to the imprecision scaling and, based on the results, devise a test to determine whether quantum-enhanced precision is feasible in the presence of unknown noise. We then assess whether power-law scaling of phase imprecision vs particle number N is valid asymptotically and establish a method to determine this power $\wp$. Finally, we define the resource for generating and implementing the control policies in terms of the scaling of the space and time cost with N.

6.3.1 Skewness and imprecision scaling

In this section, we explain the preliminary test for the robustness of DE-designed policy and determine the contribution of the skewness to the imprecision scaling. We run the policy-search algorithm on simulations of QEAPE including phase noise on a computer cluster of 48 CPUs and collect the Holevo variance of accepted policies.

For random telegraph noise and skew-normal noise, we generate the policies for $N = \{4, 5, \ldots, 100\}$, giving the wall time of 48 hours. For the log-normal noise, we generate data up to $N = 30$ as the rejection-sampling method for generating random numbers are time-consuming, taking 5 hours to generate data for $N = \{4, 5, \ldots, 30\}$. As such, the power-law scaling $\wp$ of V_{H} from the log-normal noise should not be directly compared with the data from random telegraph noise and skew-normal noise but taken as suggestions for whether the DE algorithm is able to generate successful policies.

For each noise model, we collect data for three parameter sets. For asymmetric distributions, we vary only the parameters associated with skewness in order to determine how the asymmetry affects $\Delta\widetilde{\phi}$ and $\wp$. We also compute the variance of the distributions to show the correlation with the scaling $\wp$.

The power-law scaling $\wp$ is computed by linearly fitting $\log V_{\mathrm{H}}$ vs $\log N$. The data from the ideal interferometer is used to generate the SQL and HL from the intercept of the linear fit. The SQL and HL computed in this fashion are used to provide theoretical benchmarks for the adaptive scheme given an ideal interferometer.

6.3.2 Robustness test

The robustness of QEAPE policies is determined by testing the policies in the presence of noise whose model is not recognized by the policies and the method that designs the policies, although DE-designed policies are learned in a training stage that includes the noise. Here we define the test for QEAPE, including phase noise from Sec. 6.2.1. We specify the domain of N for simulating the phase estimation schemes to obtain V_{H} in noisy conditions. The

noise parameters are variance V and skewness γ (Sec. 6.2.1), but here we fix γ for the asymptotic distributions, and we obtain the robustness-test threshold in terms of V, which is the maximum for each noise model such that the SQL is violated.

Varying N

To ascertain the asymptotic value for $\wp$, we simulate QEAPE for

$$N \in \{4, 5, \ldots, 100\}, \tag{6.17}$$

as V_{H} computed from this domain is sufficient to show power-law relationship at high N. Furthermore, increasing N further requires changing double-precision arithmetic to quadruple-precision arithmetic to generate and manipulate the sine state without rounding error. Consequently, this increase in precision leads to a fifteen-fold increase in run-time at $N = 100$, which is a large expense for generating a single data point. Therefore, we do not attempt to verify the robustness beyond this 100 particles.

Skewness

We fix skewness γ to a single value for all runs and only vary V because V is the dominant term in our noise models and γ has a small effect [4]. We fix the skewness for the asymmetric distribution to

$$\gamma = 0.8509, \tag{6.18}$$

which is sufficiently large to distinguish between the various noise models; otherwise all noise looks Gaussian. This value of γ (6.18) corresponds to $\alpha = 5$ for the skew-normal distribution where we are able to observe its effect on $\wp$ when compared to symmetric noise distributions. This same level of skewness corresponds to $\sigma' = 0.2715$ in the log-normal distribution.

Robustness threshold

Our ϱ is robust if the SQL-breaking condition $\wp > 1/2$ is satisfied for all four noise models in Sec. 6.2.1. As discussed in Sec. 6.3.2, we fix γ, and we ignore higher cumulants; thus, ϱ robustness threshold is in terms of V, i.e., the maximum V such that $\wp > 1/2$ holds for all four noise models. This optimization problem is hard so we adopt a simpler characterization procedure instead to get insight into the robustness threshold. Our approach is to run the simulations for $V \in \{1, 2, 3\}$ for symmetric noise and $V \in \{1, 3, 5, 7\}$ for asymmetric noise, and we do not push beyond $V = 7$ to keep below an imprecision width of 2π. We use these data to determine whether QEAPE policies pass the robustness test.

6.3.3 Determining asymptotic power-law scaling

To ascertain the robustness of QEAPE policies, the asymptotic $\wp$ is estimated from a subset of V_{H} at sufficiently high N, and determining this subset is done by fitting piecewise linear equations to a log-log plot of V_{H} vs N. In this subsection, we introduce five piecewise functions that are constructed from observations regarding the trend of V_{H} vs N. We then explain the method of finding the breakpoints between segments in the piecewise function and fitting the functions to the data. Using the criteria in Sec. 6.2.2, we create a majority-vote method for selecting the function that best represents the data and thus $\wp$ from the last segment of the fit is used to estimate the asymptotic scaling.

Piecewise models

The trend in $\log V_{\mathrm{H}}$ vs $\log N$ differs under noisy conditions, and here we describe the trends we have observed that lead to piecewise linear functions. We construct five such functions, containing 1 to 3 segments that are then fitted to $\log V_{\mathrm{H}}$ vs $\log N$.

When the interferometer is noiseless, the relationship appears to be a power-law captured in a linear equation, although the accept-reject criterion in the policy-search algorithm can lead to a different $\wp$ for $N > 93$. Once the noise level becomes high, typically $V > 1$, the

relationship does not appear to be linear for low N. Therefore, we include segmented models that fit linear interpolation to the data in the first segment.

Combining these observations, we construct five piecewise-linear models that can potentially represent $\log V_{\mathrm{H}}$ as a piecewise function of $\log N$. The models have 1 to 3 segments, each segment connected at the breakpoints determined by the fitting methods. Three of the models are one, two, and three linear models, whereas the other two are a two-segment model, where the first segment (low N) is a linear interpolation, and a three-segment model where the first segment is a linear interpolation and the second and third segments are linear.

Fitting method

The method for fitting the linear equations that we use is the least-squares method [172]. However, because the functions in Sec. 6.3.3 are segmented, we include a step to optimize the breakpoints depending on the specific function.

The full model for the regression analysis is the three-segment linear function, which is fitted using a linear-square method and the segments determined by a heuristic global optimization algorithm [176]. The two-segment linear function is also fitted using the same least-square method although a brute-force search is used to find the breakpoint starting from $N = 4$.

The method for finding the breakpoints for models with interpolation is different as the linear interpolation leads to a small residual. Thus, optimizing using the least-squares method can lead to a single segment of linear interpolation. For this reason, we first find the stop point for the first segment by fixing the latter segments to a single linear line and search for a breakpoint that results in a large decrease in sum square error. As for the single-segment linear model, we use a standard library to fit the data.

Fitting figures of merit and model selection

After the functions are fitted to $V_{\mathrm{H}}(N)$, the criteria $\overline{R^2}$, $\mathrm{AIC_c}$, the F-value, and Mallows's C_p (Sec. 6.2.2) are calculated for each of the function and the fits are visually inspected. These criteria are used to select the function that best fits the data. Here we explain how the best function is chosen.

After the functions are fitted and the criteria are calculated, each fit is visually presented and inspected to ascertain that the segmentation fits the pattern. If correction is unnecessary, the functions are then ranked for each of the criteria. Note that we do not perform the full F-test as we discovered that reduced models typically fail the test even though there is no discernible difference when compared to the full model. However, the F-value can still be used to quantify the difference between using the full and reduced model, so we use the F-value to rank the functions instead of conducting a pass-fail test.

After the functions are ranked, the function that is voted as best by most of the criteria is chosen to represent the data. In the case where the full model, the three-segment linear function, is voted according to the criteria, the value of $\wp$ from the last segment and the subset of N where this value is computed is compared to the next alternative function to determine whether the function overfits the data. If $\wp$ from the two functions differs more than 0.001, then the full model is chosen; otherwise, the alternative function is chosen. The limit we use here is specified based on the precision used in this paper and can be changed based on the desired precision of $\wp$.

6.3.4 Resource complexity

To compare and select between policies and methods of generating policies, we determine the complexities of designing and of implementing ϱ using the loop-analysis method [61] to determine both of these complexities. We begin this subsection by explaining the time complexity for designing policies. We then explain implementation complexity by describing the controller's actions and concomitant resources, which are quantified by the space and

time required for executing ϱ with N particles.

Design complexity

When an optimization algorithm is used to design ϱ, there is a time cost associated with the use of the algorithm. The scaling of the upper bound of this cost is called the design complexity. We assume that the calculation is performed on a simulation of the AQEM task, as is common practice in policy design in quantum control [86], and, therefore, the time cost includes the cost of simulating the AQEM task. We assume that only a single processor is used for the purpose of comparing policy-design methods, although this cost can be reduced by parallelizing the optimization on multiple CPUs. For policies that are devised through analytical optimization, such as Bayesian feedback, the cost is zero as no design algorithm is used.

Time cost of training a policy. Here we calculate the upper bound in the time cost for generating a policy using the noise-resistant DE algorithm (Alg. 1), which has been previously shown to be $O(N^6)$ through loop analysis [35]. Here we show that when the time cost of the recursively training N-particles policy using $(N-1)$-particle policy as the initial guess, the time complexity is $O(N^7)$. This calculation makes certain assumptions about the implementation of the algorithm and how the relevant DE parameters are set. Therefore, we start with explaining these assumptions before listing the loops in the DE algorithm. We then compute the time cost for training an N-particle QEAPE and report the term with the highest order in N as the time complexity.

The assumptions we make in this calculation are in regards to the accept-reject criterion and the implementation of the algorithm. We ignore the accept-reject criterion and the criterion for selecting a loss-tolerant policy as the criteria make the time cost random. Instead, we fix the number of generation, as is the case for $N = \{4, 5, \ldots, 93\}$ without checking for loss-tolerance, in order to estimate the how the time cost scales with N. We acknowledge

here that the number of generations used in our training is low, due to the use of phase adjustment vector $\mathbf{\Delta}$ (Eq. 3.45) for $N-1$ to initialize the search at N. In terms of the implementation, we assume that the algorithm is executed on a single CPU, which would give us the maximum time required to find a successful policy for QEAPE.

The time cost of the policy-search algorithm is calculated based on the loop analysis of DE, which is the optimization algorithm used. Differential evolution improves the policy candidates iteratively, and thus the outermost loop in the algorithm is of the number of generations Υ before a policy is accepted. For every generation, the candidates computes the fitness values of the policies, generate offspring, and select the members who survive into the next generation. Therefore, for each generation there is a loop over the size of the population N_{p}. Let the time use to execute a fitness function be τ_f, the time to generate an offspring be τ_o, and the time to select a member for the next generation be τ_s. The time cost for a noise-resistant DE τ_1 is

$$\tau_1 = \sum_{t=1}^{\Upsilon} \sum_{l=1}^{N_{\mathrm{p}}} (\tau_f + \tau_f + \tau_o + \tau_s) \tag{6.19}$$

The second τ_f accounts for the second call to the fitness value in the noise-resistant DE (Alg. 1).

As the number of generations Υ and N_{p} are parameters of the DE algorithm, we have the freedom to choose these parameters as long as DE can still deliver a successful policy. In previous work, it has been shown that the $\Upsilon \in O(N)$ enables a solution at $N > 45$ to be found [35]. Therefore, we assume $\Upsilon = \alpha' N$ for this calculation. In the implementation we fix Υ to $\Upsilon = \alpha' N_{\max}$, which we also found to work. We also fix the size of the population $N_{\max}/2$, so that in the implementation we can ensure that every CPUs computes for one candidate. The choice of the fixed size follows a finding that the population should scales as $N_{\mathrm{p}} \in O(N)$ to account for the increase in the dimension of the search space [177].

The fitness of a policy candidate is quantified by the sampled sharpness (Eq. 3.42), and

the program simulates $K = 10N^2$ instances of QEAPE (Sec. 3.4.1). For each instance of adaptive phase estimation, N particles sequentially pass through the interferometer and measured. Measuring the m^{th} particle results in an update of the input state, which is stored as a vector of size $(N-m)$. Updating each of these elements is assumed to use up a constant time ζ_1. Therefore, an instance of QEAPE results in a time cost that is

$$\sum_{m=1}^{N} \zeta_1 (N-m) = \frac{\zeta_1}{2} \left(N^2 - N\right) \tag{6.20}$$

Hence, computing fitness for a policy candidate costs

$$\tau_f = \frac{10\zeta_1}{2} \left(N^4 - N^3\right) \tag{6.21}$$

What remains are calculating the time cost for generating an offspring and for selecting the candidates for the next generation. Randomizing the candidates for generating the offspring has a constant time cost ζ_2. Generating an offspring, on the other hand, involves a loop over the N elements, and so the time cost is $\zeta_3 N$ for a candidate policy. Deciding whether to keep the candidate or replacing it with its offspring is a comparison that uses up constant time ζ_4.

Adding up the time costs for the noise-resistant DE (1) gives us

$$\tau_1(N) = \Upsilon \left(N_{\text{p}} \zeta_1 (N^4 - N^3) + N_{\text{p}} (\zeta_2 + \zeta_3 N + \frac{\zeta_1}{2}(N^4 - N^3)) + N_{\text{p}} \zeta_4 \right) \tag{6.22}$$

$$= \frac{3}{2}\alpha' \zeta_1 \left(N^6 - N^5\right) + \frac{\alpha' \zeta_3}{2} N^3 + \frac{\zeta_2 + \zeta_4}{2} \alpha' N^2. \tag{6.23}$$

The highest order of N in the time cost is N^6, and so the time complexity for generating a policy scales as $O(N^6)$.

The calculation so far assumes that the policy for $N-1$ particles is already available. In practice, the $N-1$ policy has to be trained before the N-particles policy can be trained. Thus, there is another recursive training process starting from N_0, where the policy is initialized

using a uniform distribution, to a policy of size N. This recursive training can be viewed as another loop over the training process whose time complexity is $O(N^6)$. Using the Gauss summation formula and assuming $N_0 = 1$, the complexity of finding a successful policy for is therefore $O(N^7)$. This complexity does not change when noise is included as the addition of noise does not change the number of loops but add to the overhead of the time cost. On the other, an addition of the accept-reject criterion, which is used after $N = 93$ to make the algorithm scalable, would change the time cost and hence the training complexity. The process of selecting a loss-tolerant policy is also excluded from this analysis.

Implementation complexities

In this subsection, we explain the resource complexity required to implement an AQEM policy, quantified by the scaling of space and time costs with the number of particles N [62]. We begin by explaining the connection between an AQEM policy and an algorithm by viewing the controller as a computer, allowing us to use the method of algorithm analysis to calculate the complexities [61]. We then define the space and time cost for implementing policies and how these costs are calculated.

Controller. The controller holds ϱ (Eq. 3.31) and implements ϱ to execute decisions based on feedback from detections of outgoing particles. The controller is essentially an agent who receives input from detectors and transmits a control signal to an actuator that shifts the interferometric phase. Thus, ϱ is represented as a computer algorithm expressed as a computer program. Computer memory is required to store ϱ and time to execute ϱ. The candidate ϱ can be designed by various means including Bayesian feedback and policy-search method (Sec. 3.4.2). Space and time costs are discussed below.

Space complexity. We determine the upper bound for space cost, which is the worst-case amount of memory used by an algorithm reported as a big O function of the size of the problem [62]. As ϱ is executed by an algorithm, this worst-case, or maximum-size, memory

corresponds to how much space is required to hold the critical information required to execute the feedback. For ϱ, the computer's space cost for memory depends on the type of ϱ.

In the case of the DE-designed policy, the size of the policy is the number of parameters in Eq. 3.44, which is N [35]. This linear scaling of the policy size, with respect to the number N particles, is due to the generalized-logarithmic-search heuristic leading to the size N phase-adjustment vector (Eq. 3.45).

In Bayesian feedback procedure, the feedback is determined by a model of the quantum state, which is updated according to the measurement outcome x_m. To avoid assuming the value of ϕ, the amplitudes $\psi_{\boldsymbol{x}_m}(\phi)$ of the quantum state is decomposed to

$$\psi_n(\boldsymbol{x}_m, \phi) = \sum_{z=-m/2}^{m/2} \psi_{n,m,z}(\boldsymbol{x}_m)\mathrm{e}^{\mathrm{i}z\phi}, \tag{6.24}$$

and so updating ψ_n is done by updating $\psi_{n,m,z}$, which are later used to compute Φ_m. The value of $\psi_{n,m,z}$ are stored in an array of size $(N+1-m)\times(m+1)$, whose maximum size is $(N+1)^2$ because $m \in \{1, 2, \ldots, N\}$. Therefore, the space complexity of the Bayesian feedback is $O(N^2)$, whereas the complexity of the DE-designed policy is $O(N)$.

Time complexity. Time complexity is determined from the upper bound of the time cost that is used to implement a single AQEM shot, i.e., the cost for using all N particles once. This cost is calculated by assuming the time a particle takes to pass through an interferometer is constant, mimicking the physical implementation of the control procedure. We then use loop analysis, which counts the number of loops that perform operations, which we assume all take the same constant time [61]. For a shot of AQEM task using N particles, Φ_m is computed N times, corresponding to each particle passing through the interferometer and being detected. For each particle, nested loops for computing Φ_{m+1} exist according to ϱ. The complexity is reported as the scaling of this time cost with N.

We determine the implementation cost for a DE-designed policy by recognizing that the update of Φ_m according to Eq. (3.44) is constant in time. We denote the time for a particle to

pass through the interferometer as β_1 and the time used to compute Φ_{m+1} is β_2. Therefore, the time cost for the execution of the adaptive phase estimation is

$$\tau_2^{(L)}(N) = \sum_{m=1} N(\beta_1 + \beta_2) \tag{6.25}$$

$$= (\beta_1 + \beta_2)N, \tag{6.26}$$

and the time complexity is $O(N)$.

Bayesian feedback, on the other hand, has nested loops for updating the quantum state and calculating Φ_{m+1} for each particle that passes through. The update of the quantum state involves two nested loops for $n \in \{0, 1, \ldots, N-m\}$ and for $z \in \{1, 2, \ldots, m+1\}$. Each step within these nested loops involves a constant number of computations and therefore incurs a constant time cost β_1'. The time cost for updating the state is

$$\sum_{n=0}^{N-m} \sum_{z=1}^{m+1} \beta_1' = \beta_1'(m+1)(N-m-1). \tag{6.27}$$

The next step of the computation is to generate three candidates for Φ_{m+1} and selecting the one with maximum estimated sharpness (Eq. 3.42). If $m < N$, the candidates are computed from the updated quantum state in a series of three two-nested loops. The loop analysis shows that the time costs for these loops are:

loop a

$$\sum_{n=1}^{N-m+1} \sum_{z=1}^{m} \beta_2' = \beta_2' m(N-m+1), \tag{6.28}$$

loop b

$$\sum_{n=0}^{N-m} \sum_{z=1}^{m+1} \beta_3' = \beta_3'(m+1)(N-m+1), \tag{6.29}$$

loop c

$$\sum_{n=0}^{N-m} \sum_{z=1}^{m+1} \beta_4' = \beta_4'(m+1)(N-m+1). \tag{6.30}$$

If $m = N$, then only a two-nested loops are called and the time cost is

$$\sum_{z=1}^{N-1} \beta_5' = \beta_5'(N-1). \tag{6.31}$$

This computation gives us the estimate of ϕ.

Summing all these time costs for an instance of phase estimation leads to

$$\begin{aligned}\tau_2^{(B)} =& \sum_{m=1}^{N-1}[\beta_1 + \beta_1'(m+1)(N-m-1) + \beta_2' m(N-m+1) + \beta_3'(m+1)(N-m+1) \\ & + \beta_4'(m+1)(N-m+1)] + \beta_5'(N-1) + \beta_1 \qquad (6.32) \\ =& \frac{N^3}{6}(\beta_1' + \beta_3' + \beta_4' + \beta_5') + N^2\left(\frac{\beta_2'}{2} + \beta_3' + \beta_4'\right) - \frac{N}{6}(7\beta_1' + 4\beta_2' + \beta_3' + \beta_4' + 6\beta_5' + 6\beta_1) \\ & + (\beta_1' - \beta_3' - \beta_4' - \beta_5'). \qquad (6.33)\end{aligned}$$

As the term with the highest order is N^3, the time complexity for using Bayesian feedback is $O(N^3)$.

6.4 Results

In this section, we report results for the robustness test and the resources calculation of QEAPE. We start by presenting the results from the preliminary study of robustness of DE-designed policies and the effect of skewness in Sec. 6.4.1. Then we present and compare $V_{\mathrm{H}}(N)$ (Eq. 3.41) obtained by a policy-search method and Bayesian feedback (Sec. 3.4.2). Our analysis considers all four types of noise discussed in Sec. 6.2.1. We report values of $\wp$ (Eq. 3.9) from the regression procedure discussed in Sec. 6.3.3 and resource complexities discussed in §6.3.4 for both the DE-designed policies and the Bayesian feedback.

Table 6.1: The power-law scaling $\wp$ computed from the linear fit of $\log V_{\mathrm{H}}$ vs. $\log N$ [Reproduced with modification from Ref. [4] ©2017 IEEE]

Noise model	**Parameters**	$V(\phi)$	γ	$2\wp$	R^2
SQL				1	
HL				2	
ideal				1.4397	0.9991
telegraph	$P_{\mathrm{s}} = 0.5, \delta = 0.2$	0.02	0	1.4147	0.9994
	$P_{\mathrm{s}} = 0.5, \delta = 0.5$	0.125	0	1.3495	0.9994
	$P_{\mathrm{s}} = 0.5, \delta = 1.0$	0.5	0	1.2703	0.9995
skew-normal	$\sigma = 0.5, \alpha = 0$	0.25	0	1.3411	0.9995
	$\sigma = 0.5, \alpha = 2.5$	0.112797	0.575781	1.3815	0.9994
	$\sigma = 0.5, \alpha = 10$	0.0924208	0.955557	1.3915	0.9993
log-normal	$\mu = 0.2, \sigma = 0.2$	0.063367	0.614295	1.4785	0.9944
	$\mu = 0.2, \sigma = 0.5$	0.544062	1.75019	1.3062	0.9985
	$\mu = 0.2, \sigma = 1.0$	6.96798	6.18488	1.1363	0.9976

6.4.1 Preliminary results

In this subsection, we present the Holevo variance obtained from applying the policy-search algorithm to adaptive phase estimation in the presence of one of the three noise models (Sec. 6.2.1) and $\wp$ computed from these data.

Fig. 6.1b shows the distributions for random telegraph noise plotted for three parameter sets that are used to generate V_{H} in Fig. 6.1a. The scaling $\wp$ computed from these data exceed SQL (Table 6.1) with the tendency towards the SQL as δ, and therefore the variance, increases.

A skew-normal distribution narrows as skewness, determined by α, increases (Fig. 6.1d). At $\alpha = 0$, the distribution takes the form of a normal distribution with a width of σ. Without changing σ, we increase α and found that the values of $\wp$ increase, correlating to the decrease in variance rather than the increase in skewness (Fig. 6.1c).

The data from the log-normal distributions (Fig. 6.1f) shows the same trend between $\wp$ and the variance of the distribution (Fig. 6.1e). Fixing the parameter μ, the variance of log-normal distribution increases with skewness determined by σ. As a result, $\wp$ computed from the distribution with the largest σ in the three sets also is the closest to $\wp$ of the SQL.

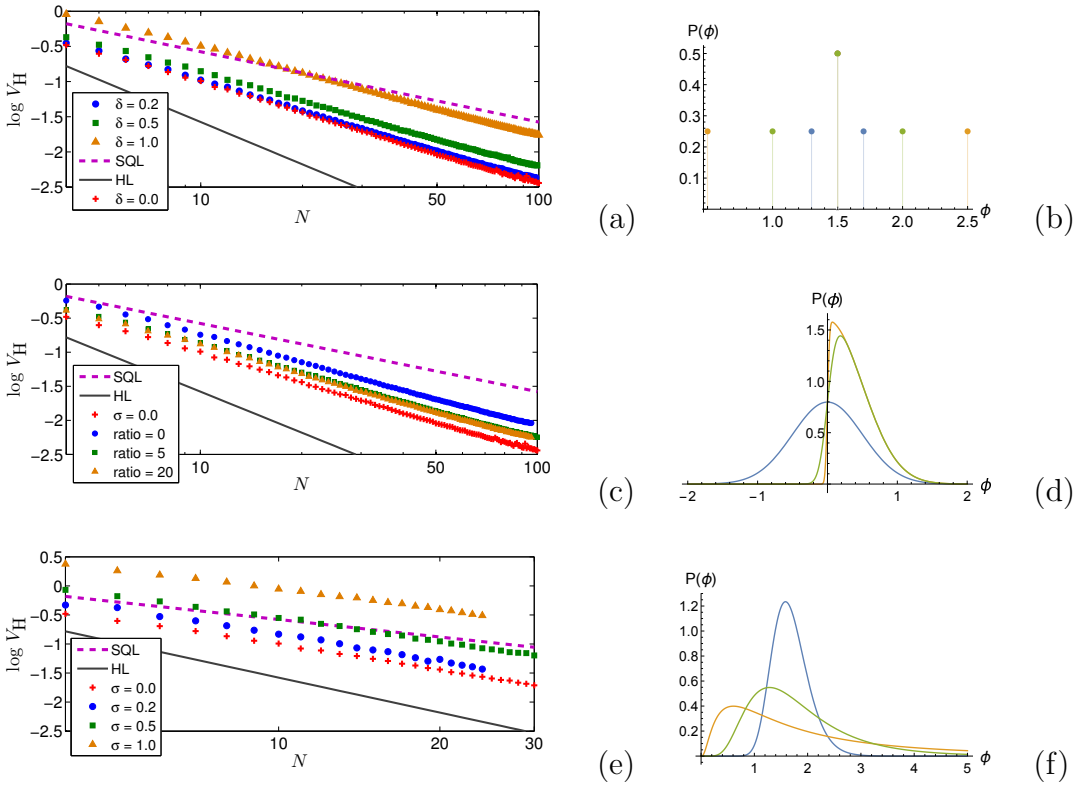


Figure 6.1: Logarithmic plots of V_{H} versus number of particles in the presence of phase noise. The red crosses are the results from an ideal interferometer. The intercept from these data is used to calculate the SQL (purple dash line) and HL (solid black line). (a) V_{H} in the presence of random telegraph noise with $P_{\mathrm{s}} = 0.5$. Three values of δ are used in the simulation: 0.2 rad (blue circles), 0.5 (green squares), and 1.0 (orange triangle). (b) The random telegraph distributions for $P_{\mathrm{s}} = 0.5$ and three values of δ: 0.2 (blue), 0.5 (green) and 1.0 (orange). (c) V_{H} in the presence of skew-normal noise with $\sigma = 0.5$. Three values of α/σ are used in the simulation: 0 (blue circles), 5 (green squares), and 20 (orange triangle). (d) The skew-normal distributions for $\sigma = 0.5$ and three values of α/σ: 0 (blue), 5 (green) and 20 (orange). (e) V_{H} in the presence of log-normal noise with $\mu = 0.2$. Three values of σ are used in the simulation: 0.2 (blue circles), 0.5 (green squares), and 1.0 (orange triangle). (f) The log-normal distributions for $\mu = 0.2$ and three values of σ: 0.2 (blue), 0.5 (green) and 1.0 (orange). [Reproduced from Ref. [4] ©2017 IEEE]

The power-law scaling of the Holevo variance in the presence of the three noise models all exceed the SQL for the selected noise parameters (Table 6.1), including the one for log-normal noise with variance of 6.96798, which is a large variance for a variable in $[0, 2\pi)$. The data in Table 6.1 also suggest that $\wp$ decreases with the increase in noise variance and is not strongly affected by the skewness at low-level variance. The algorithm is, therefore, speculated to not be robust for strong noise beyond the variance of $V = 6.96798$. Also, the inverse correlation between variance and $\wp$ is only shown for the phase noise of the same functional form. A conclusive relationship between variance and $\wp$ cannot yet be established due to limited data.

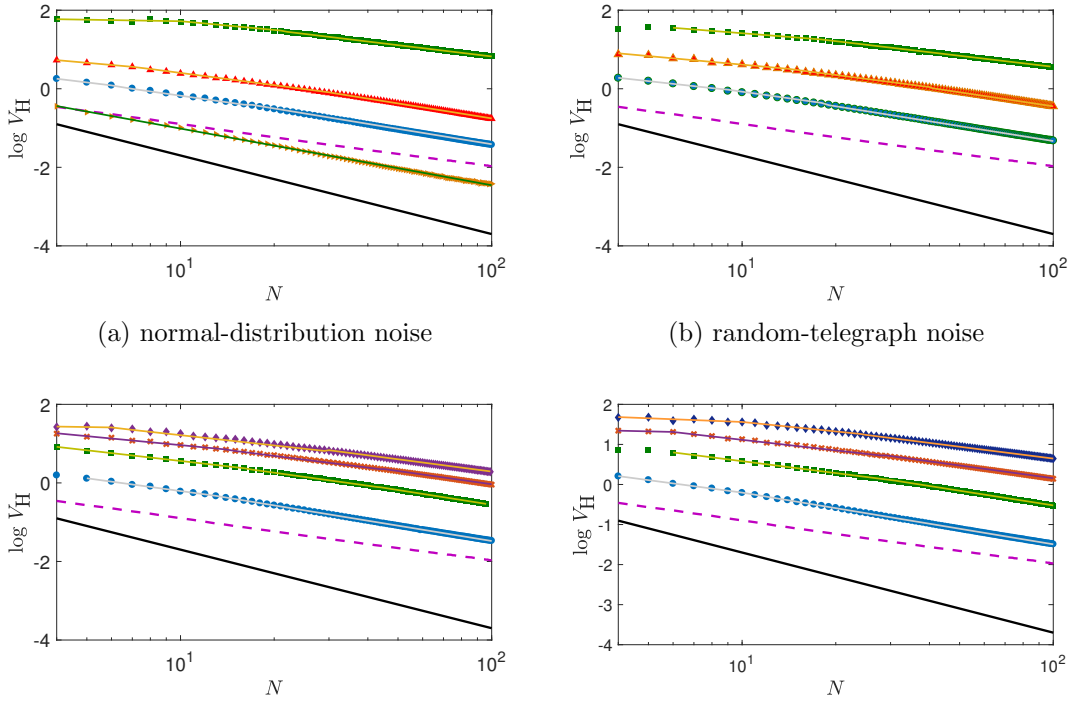
6.4.2 Variance vs number of particles

In this subsection, we present results for V_{H} as a function of number N of particles. Specifically, we present plots of V_{H} vs N from 4 to 100 particles, which is enough to determine the imprecision scaling as discussed in Sec. 6.3.4. Both cases of using DE-designed policies and of using Bayesian feedback are presented as log-log plots and compared to the SQL, with these plots obtained by computing from simulations of noiseless phase estimation using a product state $|0, 1\rangle^{\otimes N}$ following the notation of Eq. 3.35. The HL is generated based on the intercept of the SQL data using the scaling of $1/N^2$ to provide a benchmark.

Policy search

Here we present the log-log plots of V_{H} vs N from QEAPE using DE-designed policies, as shown in Fig. 6.2. Subfigures 6.2(a–d) present V_{H} for inclusion of normal-distribution noise, random-telegraph noise, skew-normal noise, log-normal noise, respectively.

Figure 6.2(a) also includes V_{H} from noiseless interferometry. This locus appears as a straight line in the plot, indicating a power-law relationship between V_{H} and N. As the noise variance V increases, this power-law relationship breaks into two parts, clearly visible in the V_{H} vs N plot from $V = 3$. This trend also appears at $V = 2$ as the model selection



(a) normal-distribution noise

(b) random-telegraph noise

(c) skew-normal-distribution noise

(d) log-normal-distribution noise

Figure 6.2: Logarithmic plots of Holevo variance from simulations of QEAPE. The policies are designed using a policy-search method implemented in the specified noise condition, namely, (a) normal-distribution noise, (b) random-telegraph noise, (c), skew-normal-distribution noise, and (d) log-normal-distribution noise. The plot for the normal-distribution noise also includes the data from the noiseless simulation (brown side-facing triangle) and its linear fit (green solid). The blue circles are data when $V = 1$, the red triangles when $V = 2$, the green squares when $V = 3$, the brown plus when $V = 4$, the brown crosses when $V = 5$, and the purple diamonds when $V = 7$. The lines shown are the piecewise-linear fits of the data whose scaling is reported. The solid black line is the HL and the dashed purple line in the SQL is generated from noiseless QEAPE. [Reproduced from Ref. [5], Fig. 1.]

procedure Sec. 6.3.3 selects the two-segment model for this data set. The observation that the power-law relationship fails when noise is included is also evident in Figs. 6.2(b–d). In these cases, the plots are fit to two- or three-segment linear equations as V_{H} appears to have a bump at low N as V increases.

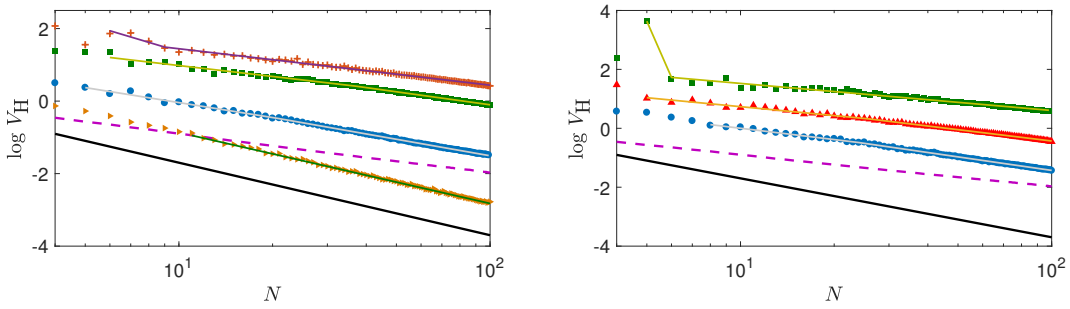
The increase in phase noise V also results in an increase in the intercepts of V_{H} power-law lines; however, the rate of change appears to depend on the noise model. The difference is shown in Fig. 6.2(a–b), both include symmetric noise distributions but with different spacing of the intercepts. The same observation holds for Fig. 6.2(c–d), which are from asymmetric distributions. Comparing the four plots shows that the intercepts appear to increase slower for asymmetric distributions than the symmetric distributions, being close to 1 for $V = 3$ in the former and $V = 2$ for the latter.

Bayesian feedback

Log-log plots of V_{H} as a function of N, shown in Fig. 6.3, are computed from simulations of QEAPE controlled by Bayesian feedback. Figures 6.3(a–d) present V_{H} in the presence of normal-distribution noise, random-telegraph noise, skew-normal noise, and log-normal noise respectively.

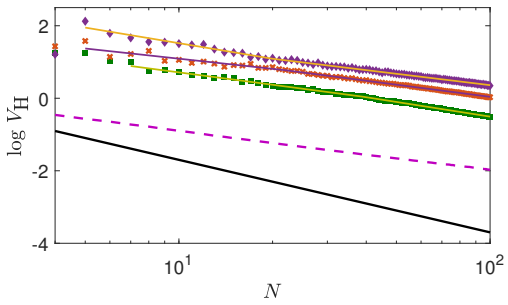
Similar to Fig. 6.2, the trend of V_{H} vs N in Fig. 6.3 shows that the power-law relationship also breaks into parts. Instead of a bump, V_{H} from Bayesian feedback exhibits noise for low N. For this reason, the model-selection procedure Sec. 6.3.3 favours models with linear interpolation in the first segment. The subsequent segment appears straight in the log-log plots, although some, such as $V = 7$ in Fig. 6.3(c), shows a break into two linear segments.

The intercepts of the V_{H} vs N plots increase with the increase of V, and the observation of the changes are similar to when DE-designed policies are used (Sec. 6.4.2). The asymmetric noise show a slow increase in intercept when compared to symmetric noise, and the rate of change depends on the noise model.

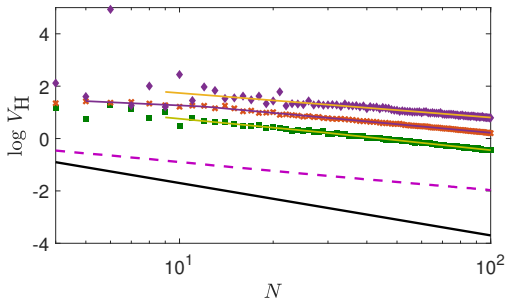


(a) normal-distribution noise

(b) random-telegraph noise



(c) skew-normal-distribution noise



(d) log-normal-distribution noise

Figure 6.3: Logarithmic plots of Holevo variance from simulations of QEAPE using Bayesian feedback method. The simulation includes one of the four noise models, namely, (a) normal-distribution noise, (b) random-telegraph noise (c), skew-normal-distribution noise, and (d) log-normal-distribution noise. The plot for the normal-distribution noise also includes the data from the noiseless simulation (brown side-facing triangle) and its linear fit (green solid). The blue circles are data when $V = 1$, the red triangles when $V = 2$, the green squares when $V = 3$, the brown plus when $V = 4$, the brown crosses when $V = 5$, and the purple diamonds when $V = 7$. The lines shown are the piecewise-linear fit of the data whose scaling is reported. The solid black line is the HL and the dashed purple line in the SQL generated from noiseless QEAPE. [Reproduced from Ref. [5], Fig. 2.]

Table 6.2: Power-law scaling from QEAPE under noisy conditions using DE-designed policies $\wp_{\mathrm{S}}$ and Bayesian feedback $\wp_{\mathrm{B}}$. [Reproduced from Ref. [5], Table. 1.]

	V	γ	$2\wp_{\mathrm{S}}$	$\overline{R^2}_{\mathrm{S}}$	$2\wp_{\mathrm{B}}$	$\overline{R^2}_{\mathrm{B}}$
SQL			1		1	
HL			2		2	
No noise			1.459	0.9998	1.957	0.9993
Normal	1	0	1.302	0.9999	1.512	0.9985
	2	0	1.267	0.9999	–	–
	3	0	0.954	0.9992	1.190	0.9997
	4	0	–	–	1.004	0.9948
Random telegraph	1	0	1.266	0.9999	1.526	0.9991
	2	0	1.186	0.9997	1.277	0.9967
	3	0	0.935	0.9993	0.919	0.9892
Skew-normal	1	0.8509	1.296	0.9999	–	–
	3	0.8509	1.246	0.9999	1.343	0.9987
	5	0.8509	1.118	0.9998	1.116	0.9927
	7	0.8509	1.039	0.9996	1.041	0.9964
Log-normal	1	0.8509	1.290	0.9999	–	–
	3	0.8509	1.217	0.9998	1.258	0.9919
	5	0.8509	1.058	0.9997	1.086	0.9961
	7	0.8509	0.981	0.9994	0.9209	0.7965

6.4.3 Power-law scaling

In this subsection, we present values of $\wp$, summarized in Table. 6.2, that are estimated by fitting V_{H} plots in Sec. 6.4.2. These $\wp$'s are from the last segment of the selected piecewise linear models (Sec. 6.3.3), which changes with the increase in V. We also include $\overline{R^2}$ (Eq. 6.11) to show the goodness of fit.

The power-law scaling for DE-designed policy, namely, $\wp_{\mathrm{S}}$, shows a decrease as the noise level V increases, starting from the noiseless phase estimation at $2\wp_{\mathrm{S}} = 1.459$. The DE-designed policies fail to deliver $\wp_{\mathrm{S}} > 1/2$ when $V = 3$ for the symmetric noise distributions. This limit increases with asymmetric noise models to $V = 7$ in log-normal noise. Skew-normal noise only shows a scaling at approaches the SQL but does not breach it at all.

Similar trends are observed for Bayesian feedback. The scaling $\wp_{\mathrm{B}}$ from noiseless interferometer closely approximates the HL at $2\wp_{\mathrm{B}} = 1.957$ and approaches SQL when $V = 4$

Table 6.3: Upper bound in policy space and time cost of the policy from DE algorithm (DE) and Bayesian feedback (BF). [Reproduced from Ref. [5], Table. 2.]

Complexity	DE	BF
Design time	$O(N^7)$	–
Policy space	$O(N)$	$O(N^2)$
Implementation time	$O(N)$	$O(N^3)$

for normal-distribution noise. This limit drops to $V = 3$ when random-telegraph noise is included. This limit also appears at $V = 7$ for log-normal noise, whereas the same noise level just leads to $\wp_{\mathrm{B}}$ approaching SQL when skew-normal noise is present. This trend, aside from the case of normal-distributed noise, is the same as the trend for the DE-designed policies.

The goodness-of-fit for these fits are reported in term of $\overline{R^2}$, where $\overline{R^2} = 1$ indicates a perfect fit. The values of the goodness is $\overline{R^2}_{\mathrm{S}} > 0.999$ for V_{H} delivered by DE-designed policies and $\overline{R^2}_{\mathrm{B}} > 0.99$ for Bayesian feedback except for when a log-normal noise of $V = 7$ is present. Overall, the fits that are chosen using the method in Sec. 6.3.3 provide good fits to the data, and the DE-designed policies always deliver fits with $\overline{R^2}_{\mathrm{S}} > \overline{R^2}_{\mathrm{B}}$.

6.4.4 Bounds on time and space costs

The results from calculating space and time complexities for both designing and implementing DE-designed policies and Bayesian feedback is shown in Table 6.3. Here we compare these results.

The time complexity for generating policies, named here design time, is of a high polynomial degree when the DE algorithm is used. Bayesian feedback, which is designed through an analytical process, incurs no time cost for the design. When the implementation time is compared, the time complexity goes from $O(N)$ for the DE-designed policy to $O(N^3)$ for Bayesian feedback as shown in Table 6.3. Space complexity for the DE-designed policy is also linear with respect to the number of particles whereas Bayesian feedback requires a memory of $O(N^2)$, also shown in Table 6.3.

6.5 Discussion

In this section, we discuss policy robustness and why QEAPE policies are so strongly noise-resistant. We explain the difference between the scaling parameter $\wp$ attained by DE-designed policies and Bayesian feedback. Our analysis enables a user to choose policy-design methods that minimize resource complexity for design and for implementation as appropriate.

6.5.1 Robustness of QEAPE policies

In this subsection, we discuss the robustness of QEAPE policies and the robustness threshold based on $\wp$ in Table 6.2. Both DE-designed policies and Bayesian feedback are able to deliver $\wp > 1/2$ for all four noise models until $V = 3$ at which point scaling in the presence of random-telegraph noise fails to exceed the SQL. For noise models that are asymmetric, quantum-enhanced precision is observed up to $V = 7$ at skewness $\gamma = 0.8509$. This high level of robustness is surprising, especially so for Bayesian feedback, as the interferometer dynamic no longer agrees with the noiseless assumption.

One possible reason behind this high robustness threshold is the property of the sine state (Eq. 3.36) being strongly robust against loss [128]. The structure of the sine state may also contribute to robustness against phase noise as well. Adaptive-phase policies discovered by evolutionary algorithms could also play a major role in the robustness of the QEAPE scheme.

The effect of the feedback policy is highlighted by the threshold for normal-distribution noise, where the threshold for DE-designed policies is at $V = 3$ as opposed to Bayesian feedback at $V = 4$. This result, however, does not imply that robustness against the noise of unknown distribution is improved as the thresholds for all other noise models are the same for both the DE-designed policies and Bayesian feedback.

6.5.2 Space cost and power-law scalings

Table 6.2 shows that the power-law scaling delivered by Bayesian feedback is consistently superior to those delivered by the DE-designed policies before the robustness threshold is reached. Here we use the space complexity of the policies in Table 6.3 to explain the reason behind this difference.

Table 6.3 shows that Bayesian feedback has a space cost that scales a polynomial degree higher than for the DE-designed policies, which indicates that Bayesian feedback utilizes more information and hence is more complex than the DE-designed policies. By using a trusted quantum-state model, Bayesian feedback effectively uses the history of measurement outcomes $x_1 x_2 \cdots x_m$ to determine Φ_{m+1} instead of the current outcome x_m, which is the approach used by DE-designed policies in (Eq. 3.44). As such, DE-designed policies are restricted by the generalized-logarithm-search strategy (Sec. 3.4.2) and so cannot deliver a value of $\wp_{\mathrm{S}}$ that approaches the HL. Improvement of DE-designed policies is done by changing the update rule, i.e., the structure of the policy, so that ϱ uses a part of the measurement history.

6.5.3 Choosing a QEAPE policy

In this subsection, we explain how the space and implementation time complexity (Sec. 6.3.4) is used to decide between competing policies and method of generating the policies. In particular, we discuss choosing between the DE-designed policies and Bayesian feedback.

The consideration of the space and time complexity of the policies comes after ascertaining that the candidate policies are able to deliver the target performance. In the case of QEAPE, the target is to attain $\wp > 1/2$, which both the DE-designed policies and Bayesian feedback are able to deliver. Both methods also have the same robustness threshold against phase noise of unknown distribution. Based on these comparisons, both policies appears equally suitable.

When comparing space and implementation complexity (Sec. 6.4.4), DE-designed policies

show an advantage as scaling of both costs are bound to linear scaling with respect to particle number N, whereas Bayesian feedback is quadratic in space complexity and cubic in time complexity. For this reason, we favour DE-designed policies for robust QEAPE.

We do not include the design complexity in choosing a QEAPE policy as the training cost is a one-time cost. Once the training is completed and a successful policy is found, the policy can be utilized with no further training as long as the quantum dynamic remains unchanged. The design cost can also be reduced by distributing the training on several CPUs (Sec. 5.5). The situation where design time may be of interest is when learning occurs in a physical setup where parallelizing is not possible and one shot of the experiment is expensive. In this case, there could be an upper bound to the number of experiments and hence time that can be invested in training ϱ. In that case, design complexity may become relevant.

6.6 Conclusion

We have tested QEAPE policy robustness based both on Bayesian feedback and on direct policy search, and we compared performance with respect to resource complexities for designing and for implementing policies. We find that both the DE-designed and the Bayesian-feedback policies are robust against phase noise up to the noise level characterized by $V = 3$. Although imprecision scaling provided by Bayesian feedback is superior to the policies designed by direct policy search, direct policy search consumes far less time and space resource than does Bayesian feedback for policy implementation.

Our methods for robustness testing and for comparing policies based on resource complexity could have applications to AQEM and quantum-control applications in general. Robustness is a desirable property of any QEM schemes and the test presented here can be adapted to quantify the robustness of non-adaptive procedures and investigate the role of the input state to the robustness of QEM generally. Quantifying the resource used by control policies can be used to show the efficacy of the policies not only in AQEM but in other

quantum control tasks, and the comparison of the resource complexity can be used to select a policy that is most efficient in accomplishing a task.

Chapter 7

Conclusion

In this chapter, we summarize the findings from the three research objectives and discuss their implication to our main research problem. We then discuss future directions of this research including both the fundamental questions and the possible applications.

7.1 Summary

Our main research problem is determining the advantages and disadvantages of policies generated using an evolutionary algorithm, which is one method of generating a control policy that does not use a model of the quantum dynamics. In order to do so, we first formulated a general AQEM scheme to a decision-making process and analyzed the characteristics of a particular measurement scheme, namely the quantum-enhanced adaptive phase estimation. Secondly, we devised a noise-resistant policy-search algorithm using DE that is capable of generating successful policies up to 100 particles, pushing beyond the previous limitations. Lastly, we compared the performances, robustness, and complexity of resources used by the policies generated using this DE algorithm to the ones from Bayesian feedback. Below, we discuss the implications of our findings point by point.

7.1.1 AQEM as a decision-making process

In this part of the research, we cast a discrete-time AQEM procedure in a decision-making process and found that the AQEM can be described as a POMDP with ϕ being the missing parameter that would turn the POMDP into a MDP model. From this result, we proceeded to determine a suitable method for generating the feedback policy and its role in the imprecision of obtained from an AQEM procedure.

Since the assumption about the quantum dynamics cannot be validated, we carry out the analysis of the POMDP without trusting the dynamic model. This assumption means that the POMDP is without defined transition probabilities and necessitates the use of a policy-generation technique that samples control trajectories from the plant. Each trajectory is randomly selected due to the randomness of the quantum measurement outcomes with the probability determined in part by the policy. As such, the sequence of measurement outcomes alone does not tell us if a certain trajectory is the correct one, i.e., the one leading to $\tilde{\phi} \approx \phi_0$ for $\phi_0 \in [0, 2\pi)$, and an optimization method that periodically evaluates an entire policy instead of the actions was used.

Using this formalism, we also gained insight into the physical explanation for AQEM imprecision. The role of a policy is two folds. One is that the policy discretizes the domain of the estimate $\tilde{\phi} \in [0, 2\pi)$ into at most 2^N possible values. The policy also controls the probability distribution for obtaining each of these 2^N value of $\tilde{\phi}$, which means the policy controls the probability distribution of $\tilde{\phi}$ and hence the imprecision. The Cramér-Rao lower bound for this imprecision can be derived, but because of the bound's dependence on the policy, we were not able to obtain the bound as a function of N. A policy-search algorithm based on DE was used to find the policies that minimize the imprecision and reveal the lower bound for the product states, which is shown to be the SQL, and the sine states, which exceeds the SQL.

From this part of the work, we establish a connection between the discrete-time AQEM procedure and classical techniques that have been used to solve decision-making problems.

The technique we were able to select for the adaptive phase estimation was due to the constraint on the information available in the measurement outcomes. Because we are working with only one closed-loop quantum control procedure, it is unclear if this restriction is fundamental to quantum control task and whether there can be a workaround that enables more information to be gleaned from the plant without compromising the performance of the procedure.

7.1.2 Scalable and noise-resistant policy-search algorithm

In the second part of the thesis, we devise a noise-resistant policy-search algorithm that can be used for a hundred-particle QEAPE. Although policy-search algorithms have been used previously to generate policies, they were not able to deliver successful policies for a high number of particles or not tested in the presence of phase noise. We overcame both limitations by using DE to enable successful search in a high-dimensional policy space and including the accept-reject criteria for the high number of particles so that a sufficient number of iterations is permitted to obtain a successful policy, and use mean policy performance to average out the added noise in the fitness function. We have shown that this algorithm is able to deliver imprecision that consistently scales better than SQL up to 100 particles, which is the limit to obtaining an accurate simulation of the QEAPE using double-precision floating-point numbers.

Although the algorithm was designed with QEAPE in mind, the policy-search algorithm does not use any knowledge of the quantum dynamics and only the performance of the policy. Therefore, the algorithm can be adapted to other quantum control problems that can be efficiently solved using black-box optimization. The code is also written into modules and so can be customized to match specific challenges of a quantum control problem. We describe the features of this code, its structure and location, and how it can be used for other problems in Appendix A.

7.1.3 Comparing AQEM policies

In the last part of this thesis, we explored the advantages and disadvantages of model-based and model-free approaches for generating policies using QEAPE as a test problem. The noise-resistant policy-search algorithm we devised in the second part of the thesis was used as an example of a model-free method, whereas the Bayesian feedback is used as an example of a model-based method. We devise three criteria for the comparison. The first is the performance when the quantum dynamic matches the model used for the model-based policy generation. The second is the robustness test, which is when the dynamics in the simulation does not match the model assumed by the policy-generation method. Thirdly, we assessed the complexity of the computational resources used in designing and implementing the policy.

By using these criteria, we were able to show that both methods are able to deliver imprecision scalings that exceed the SQL with the Bayesian feedback being superior to the DE-derived policy when the dynamic matches the model. Both methods are also robust in the presence of phase noise with unknown properties to a similar level of noise quantified by the variance of the phase-noise distribution. The time complexity of the design cost is in high polynomial for the policy-search algorithm whereas the Bayesian feedback does not incur a design cost as the policy is calculated analytically, although this cost is a one-time cost and so is not relevant in the operation of the control system. However, both the time and space complexities for implementing DE-designed policy best the ones for the Bayesian feedback due to the controller not having to compute the quantum model during the task.

Comparing performances of control policies is a common practice in control engineering, but the method for determining robustness and resources in designing and implementing the control policies were never introduced. These criteria, however, are important in quantum control as the quantum enhancement is fragile in the presence of noise and loss and so some degree of robustness should be observed in order for the policy to be practical. The time and space complexities can be used as a way of estimating scalability of the control policy

with the number of particles. Furthermore, they can be used to determine a reason for the performance difference between two types of policies.

In a practical situation, the criteria we have devised can be used to select a policy that best suits the constraints of an AQEM procedure. For example, if ϕ is a result of a transient phenomenon where the speed of the estimation procedure is of great importance, the DE-derived policy is favoured as the policy leads to a fast estimation of ϕ that is still able to exceed the SQL. However, if the value of ϕ is static and maximum precision is required, then the Bayesian feedback should be chosen. We do not consider the design complexity in choosing the policy in this work as this is an upfront cost as the policy is trained beforehand.

7.2 Research outlook

As the research in this thesis combines many disciplines, the work can be expanded in different ways. We focus here on two directions only, one being in quantum control and another in QEM.

1. One question that has arisen during the analysis of the AQEM decision-making process is whether a technique that evaluates the action rather than the policy can ever be used to generate closed-loop quantum-control policies. That is because an outcome from a quantum measurement is informationally incomplete and thus is always insufficient to determine the state of the quantum plant without a model or a model-inference method. It might be that by not assuming or constructing a model of the quantum dynamics, only black-box optimization algorithms can be applied to closed-loop quantum control problems. A conclusion cannot be drawn at this point as we have only examined one example of this class of control problems and thus further examples are needed before progress can be made.

2. For this thesis, we have made comparisons between a model-free and a model-based method to control design, which are two extreme ends of the spectrum. Control design

techniques that combined both data and model do exist and may be of interest to quantum control as the data-driven part of the algorithm allows for the control policy to adapt to unknown quantum dynamics while the model-based part can provide insight to the dynamics and reduces the number of samples needed to train the policy and thus the design time cost.

3. This thesis has been focused on quantum-enhanced single-parameter estimation, but the research direction in QEM has now been expanded into multiple-parameter estimation, where a quantum resource is used to enhanced precisions of simultaneous estimation of more than one parameter. This expansion increases the difficulty of devising measurement procedures as the dynamics become complex, which is the circumstance in which data-driven control is expected to be advantageous. Applying a data-driven control-design algorithm to such a problem may become a practical way of devising successful policies.

4. A quantum control task that stems directly from the QEM is the generation of large entangled states, which is a key challenge in implementing QEM. Although states that deliver quantum-enhanced precision can be found, the quantum channel in which to generate them is often not easy to determine. The discrete-time AQEM also has another challenge which is splitting the entangled state into bundles. It may be that by using a data-driven approach is necessary to design these procedures due to the complexity of the quantum circuits involved.

Bibliography

[1] Pantita Palittapongarnpim, Peter Wittek, and Barry C. Sanders. Controlling adaptive quantum-phase estimation with scalable reinforcement learning. In *Proc. of ESANN-16, 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 327–332, Bruges, Belgium, 2016. i6doc.

[2] Pantita Palittpongarnpim, Peter Wittek, and Barry C. Sanders. Single-shot adaptive measurement for quantum-enhanced metrology. In *Proc. SPIE Quantum Communications and Quantum Imaging XIV*, volume 9980, pages 99800H1–99800H11, Bellingham, WA, 2016. SPIE.

[3] Pantita Palittapongarnpim, Peter Wittek, Ehsan Zahedinejad, Shakib Vedaie, and Barry C. Sanders. Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics. *Neurocomputing*, 268:116–126, December 2017.

[4] Pantita Palittapongarnpim, Peter Wittek, and Barry C. Sanders. Robustness of learning-assisted adaptive quantum-enhanced metrology in the presence of noise. In *Proc. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 294–299, Banff, AB, October 2017. IEEE Systems, Man, and Cybernetics Society.

[5] Pantita Palittapongarnpim and Barry C. Sanders. Robustness of quantum-enhanced adaptive phase estimation. Manuscript submitted to *Physical Review A*, November 2018.

[6] Michael A. Nielsen and Isaac L. Chuang. Quantum information theory. In *Quantum computation and quantum information*, chapter 12, pages 528–593. Cambridge University Press, Cambridge, 10 Anv. edition, December 2010.

[7] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81(3):1301–1350, September 2009.

[8] Vittorio Giovannetti, Seth. Lloyd, and Lorenzo Maccone. Advances in quantum metrology. *Nat. Photon.*, 5(4):222–229, March 2011.

[9] Alfio Borzì, Gabriele Ciaramella, and Martin Sprengel. Introduction. In *Formulation and Numerical Solution of Quantum Xontrol Problems*, Computational science and engineering, chapter 1, pages 1–6. The Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[10] Moshe Shapiro and Paul Brumer. *Quantum Control of Molecular Processes*. Wiley, Weinheim, Germany, 2nd rev. and enlarged edition, January 2012.

[11] Matthew Grace, Constantin Brif, Herschel Rabitz, Ian A. Walmsley, Robert L. Kosut, and Daniel A. Lidar. Optimal control of quantum gates and suppression of decoherence in a system of interacting two-level particles. *J. Phys. B*, 40(9):S103, April 2007.

[12] Howard M. Wiseman. Adaptive phase measurements of optical modes: Going beyond the marginal Q distribution. *Phys. Rev. Lett.*, 75(25):4587–4590, December 1995.

[13] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. Data-driven predictive control for autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):259–286, May 2018.

[14] James R. Leigh. Synthesis of automatic feedback control loops: a more quantitative view. In *Control Theory*, chapter 3, pages 19–28. Institution of Engineering and Technology, London, UK, 2 edition, 2004.

[15] Jan Werschnik and Eberhard K. U. Gross. Quantum optimal control theory. *J. Phys. B*, 40(18):R175 – R211, September 2007.

[16] Alfio Borzì, Gabriele Ciaramella, and Martin Sprengel. *Formulation and Numerical Solution of Quantum Control Problems*. Computational science and engineering. The Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[17] James R. Leigh. An introduction to robust control design using H_∞ methods. In *Control Theory*, chapter 16, pages 201–223. Institution of Engineering and Technology, London, UK, 2 edition, 2004.

[18] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Inf. Sci.*, 235:3–35, June 2013.

[19] Narciso F. Macia and George J. Thaler. *Modeling and Control of Dynamic Systems*. Thomson Delmar Learning, Clifton Park, NY, 2005.

[20] Constantin Brif, Raj Chakrabarti, and Herschel Rabitz. Control of quantum phenomena: past, present and future. *New J. Phys.*, 12(7):075008, July 2010.

[21] Zhongsheng Hou, Huijun Gao, and Frank L. Lewis. Data-driven control and learning systems. *IEEE Trans. Ind. Electron.*, 64(5):4070–4075, May 2017.

[22] Géza Tóth and Iagoba Apellaniz. Quantum metrology from a quantum information science perspective. *J. Phys. A: Math. Theor.*, 47(42):424006, October 2014.

[23] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Inf.*, 2:15023, January 2016.

[24] Michael G. Safonov and Tung-Ching Tsao. The unfalsified control concept and learning. In *Proc. 1994 33rd IEEE Conference on Decision and Control*, volume 3, pages 2819–2824, Lake Buena Vista, FL, December 1994. IEEE.

[25] King-Sun Fu. Learning control systems–review and outlook. *IEEE Trans. Autom. Control*, 15(2):210–221, April 1970.

[26] Zhongsheng Hou and Shangtai Jin. Introduction. In *Model Free Adaptive Control: Theory and Applications*, chapter 1, pages 1–18. CRC Press, Boca Raton, FL, September 2014.

[27] Thomas Degris, Patrick M. Pilarski, and Richard S. Sutton. Model-free reinforcement learning with continuous action in practice. In *Proc. 2012 American Control Conference (ACC)*, pages 2177–2182, Montreal, Canada, June 2012. IEEE.

[28] Giorgio Battistelli, João Pedro Hespanha, Edoardo Mosca, and Pietro Tesi. Model-free adaptive switching control of time-varying plants. *IEEE Trans. Autom. Control*, 58(5):1208–1220, May 2013.

[29] David E. Moriarty, Alan C. Schultz, and John J. Grefenstette. Evolutionary algorithms for reinforcement learning. *J. Artif. Intell. Res.*, 11:241–276, September 1999.

[30] Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. *A Survey on Policy Search for Robotics*, volume 2 of *Foundations and Trends*. now, Delft, Netherlands, September 2013.

[31] Thomas Rückstieß, Frank Sehnke, Tom Schaul, Daan Wierstra, Yi Sun, and Jürgen Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn*, 1(1):14–24, March 2010.

[32] Agoston E. Eiben and James E. Smith. Popular evolutionary algorithm variants. In *Introduction to Evolutionary Computing*, Natural Computing, chapter 6, pages 99–116. Springer, Berlin, Germany, 2 edition, 2015.

[33] Peter J. Fleming and Robin C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Contr. Eng. Pract.*, 10(11):1223–1241, November 2002.

[34] Alexander Hentschel and Barry C. Sanders. Efficient algorithm for optimizing adaptive quantum metrology processes. *Phys. Rev. Lett.*, 107(23):233601, November 2011.

[35] Neil B. Lovett, Cécile Crosnier, Martí Perarnau-Llobet, and Barry C. Sanders. Differential evolution for many-particle adaptive quantum metrology. *Phys. Rev. Lett.*, 110(22):220501, May 2013.

[36] Ehsan Zahedinejad, Joydip Ghosh, and Barry C. Sanders. High-fidelity single-shot toffoli gate via quantum control. *Phys. Rev. Lett.*, 114(20):200502, May 2015.

[37] Ehsan Zahedinejad, Joydip Ghosh, and Barry C. Sanders. Designing high-fidelity single-shot three-qubit gates: A machine-learning approach. *Phys. Rev. Applied*, 6(5):054005, November 2016.

[38] Ehsan Zahedinejad, Sophie Schirmer, and Barry C. Sanders. Evolutionary algorithms for hard quantum control. *Phys. Rev. A*, 90:032310, September 2014.

[39] Dominic W. Berry and Howard M. Wiseman. Optimal states and almost optimal adaptive measurements for quantum interferometry. *Phys. Rev. Lett.*, 85(24):5098–5101, December 2000.

[40] D. W. Berry, H. M. Wiseman, and J. K. Breslin. Optimal input states and feedback for interferometric phase estimation. *Phys. Rev. A*, 63(5):053804, May 2001.

[41] James N. Hollenhorst. Quantum limits on resonant-mass gravitational-radiation detectors. *Phys. Rev. D*, 19(5):1669–1679, March 1979.

[42] Carlton M. Caves, Kip S. Thorne, Ronald W. P. Drever, Vernon D. Sandberg, and Mark Zimmermann. On the measurement of a weak classical force coupled to a quantum-mechanical oscillator. i. issues of principle. *Rev. Mod. Phys.*, 52(2):341–392, April 1980.

[43] Carlton M. Caves. Quantum-mechanical noise in an interferometer. *Phys. Rev. D*, 23(8):1693–1708, April 1981.

[44] John J . Bollinger, Wayne M. Itano, David J. Wineland, and Daniel J. Heinzen. Optimal frequency measurements with maximally correlated states. *Phys. Rev. A*, 54(6):R4649–R4652, December 1996.

[45] Johannes Borregaard and Anders S. Sørensen. Near-Heisenberg-limited atomic clocks in the presence of decoherence. *Phys. Rev. Lett.*, 111(9):090801, August 2013.

[46] S. Danilin, A. V. Lebedev, A. Vepsäläinen, G. B. Lesovik, G. Blatter, and G. S. Paraoanu. Quantum-enhanced magnetometry by phase estimation algorithms with a single artificial atom. *npj Quantum Inf.*, 4:29, June 2018.

[47] Bernard Yurke, Samuel L. McCall, and John R. Klauder. SU(2) and SU(1,1) interferometers. *Phys. Rev. A*, 33(6):4033–4054, June 1986.

[48] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum-enhanced measurements: Beating the standard quantum limit. *Science*, 306(5700):1330–1336, November 2004.

[49] Roy S. Bondurant and Jeffrey H. Shapiro. Squeezed states in phase-sensing interferometers. *Phys. Rev. D*, 30(12):2548–2556, December 1984.

[50] B. Depraetere, M. Liu, G. Pinte, I. Grondman, and R. Babuska. Comparison of model-free and model-based methods for time optimal hit control of a badminton robot. *Mechatronics*, 24(8):1021–1030, December 2014.

[51] Eugene A. Feinberg and Adam Shwartz. *Handbook of Markov Decision Processes: Methods and Applications*. International Series in Operations Research & Management Science. Kluwer Academic, Norwell, MA, 2002.

[52] Marco Wiering and Martijn van Otterlo. Reinforcement learning and markov decision processes. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, volume 12 of *Adaptation, Learning, And Optimization*, chapter 1, pages 3–44. Springer, Berlin, Germany, 2012.

[53] Steven M. Kay. Cramer-Rao lower bound. In A. V. Oppenheimer, editor, *Fundamentals of Statistical Signal Processing: Estimation Theory*, chapter 3. Prentice-Hall, Upper Saddle River, NJ, 1993.

[54] Carl W. Helstrom. Quantum detection and estimation theory. *J. Stat. Phys.*, 1(2):231–252, June 1969.

[55] Kurt Jacobs. Metrology. In *Qauntum Measurement Theory and its Applications*, chapter 6, pages 303–322. Cambridge University Press, Cambridge, UK, 2014.

[56] Paul H. Lewis and Chang Yang. Performance criteria and some effects of feedback. In *Basic Control Systems Engineering*, chapter 7, pages 157–206. Prentice Hall, Upper Saddle River, NJ, 1997.

[57] Ernst Breitenberger. Analogues of the normal distribution on the circle and the sphere. *Biometrika*, 50(1/2):81–88, 1963.

[58] David S. Newman. On the probability distribution of a filtered random telegraph signal. *Ann. Math. Stat.*, 39(3):890–896, 1968.

[59] Adelchi Azzalini and Antonella Capitanio. The skew-normal distribution: probability. In D. R. Cox, A. Agresti, B Hambly, S. Holmes, and X.-L. Meng, editors, *The*

Skew-Normal and Related Families, Institute of Mathematical Statistics Monograph, chapter 2, pages 24 – 56. Cambridge University Press, Cambridge, UK, 2014.

[60] Eckhard Limpert, Werner A. Stahel, and Markus Abbt. Log-normal distributions across the sciences: Keys and clues. *BioScience*, 51(5):341–352, May 2001.

[61] Dana Vrajitoru and William Knight. Deterministic analysis of algorithms. In *Practical Analysis of Algorithms*, chapter 5, pages 169–293. Springer, Cham, Switzerland, 2014.

[62] Dana Vrajitoru and William Knight. Introduction. In *Practical Analysis of Algorithms*, chapter 1, pages 1–7. Springer, Cham, Switzerland, 2014.

[63] Armin Rahmani, Takuya Kitagawa, Eugene Demler, and Claudio Chamon. Cooling through optimal control of quantum evolution. *Phys. Rev. A*, 87(4):043607, April 2013.

[64] Xiaoting Wang, Sai Vinjanampathy, Frederick W. Strauch, and Kurt Jacobs. Ultra-efficient cooling of resonators: Beating sideband cooling with quantum control. *Phys. Rev. Lett.*, 107(17):177204, October 2011.

[65] Pradnya A. Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *Proc. 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 261–265, Jalgaon, India, December 2016. IEEE.

[66] Paul H. Lewis and Chang Yang. Control systems engineering. In *Basic Control Systems Engineering*, chapter 1, pages 1–12. Prentice Hall, Upper Saddle River, NJ, 1997.

[67] Seth Lloyd. Coherent quantum feedback. *Phys. Rev. A*, 62(2):022108, July 2000.

[68] Clive Emary and John Gough. Coherent feedback control in quantum transport. *Phys. Rev. B*, 90(20):205436, November 2014.

[69] Narciso F. Macia and George J. Thaler. Introduction. In *Modeling & Control of Dynamic Systems*, chapter 1, pages 1–16. Thomson Delmar Learning, Clifton Park, NY, 2005.

[70] Jerzy Zabczyk. Introduction. In *Mathematical Control Theory: An Introduction*, Modern Birkäuser Classics, chapter 0, pages 1–9. Birkhäuser, Boston, 1995.

[71] Jing Zhang, Yu xi Liu, Re-Bing Wu, Kurt Jacobs, and Franco Nori. Quantum feedback: Theory, experiments, and applications. *Phys. Rep.*, 679:1–60, March 2017.

[72] Howard M. Wiseman and Gerard J. Milburn. *Quantum measurement theory*, chapter 1, pages 1–50. Cambridge University Press, Cambridge, MA, 2009.

[73] Salman Habib, Kurt Jacobs, and Hideo Mabuchi. Quantum feedback control. *Los Alamos Science*, 27:126–135, 2002.

[74] Clemens Schäfermeier, Hugo Kerdoncuff, Ulrich B. Hoff, Hao Fu, Alexander Huck, Jan Bilek, Glen I. Harris, Warwick P. Bowen, Tobias Gehring, and Ulrik L. Andersen. Quantum enhanced feedback cooling of a mechanical oscillator using nonclassical light. *Nat. Commun.*, 7:13628, November 2016.

[75] Mazyar Mirrahimi and Ramon Van Handel. Stabilizing feedback controls for quantum systems. *SIAM J. Control Optim.*, 46(2):445–467, April 2007.

[76] Howard M. Wiseman, Dominic W. Berry, Stephen D. Bartlett, Brendon L. Higgins, and Geoff J. Pryde. Adaptive measurements in the optical quantum information laboratory. *IEEE J. Sel. Top. Quantum Electron.*, 15(6):1661–1672, November 2009.

[77] James R. Leigh. Mathematical modelling. In *Control Theory*, chapter 6, pages 61–81. Institution of Engineering and Technology, London, UK, 2 edition, 2004.

[78] Alfio Borzì, Gabriele Ciaramella, and Martin Sprengel. Quantum mechanics and the schrödinger equation. In *Formulation and Numerical Solution of Quantum Control*

Problems, Computational science and engineering, chapter 2, pages 7–72. The Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[79] Siyun Wang, Jodie M. Simkoff, Michael Baldea, Leo H. Chiang, Ivan Castillo, Rahul Bindlish, and David B. Stanley. Data-driven plant-model mismatch quantification in input-constrained linear MPC. In *Proc. 11th IFAC Symposium on Dynamics and Control of Process Systems Including Biosystems 2016*, volume 49, pages 25–30, Amsterdam, Netherlands, 2016. Elsevier.

[80] Shuang Cong. Control methods of closed quantum system. In *Control of Quantum Systems : Theory and Methods*, chapter 3, pages 39–72. Wiley, Singapore, 2014.

[81] Christiane P. Koch. Controlling open quantum systems: tools, achievements, and limitations. *J. Phys. Condens. Matter*, 28(21):213001, May 2016.

[82] Howard M. Wiseman and Gerard J. Milburn. Quantum theory of optical feedback via homodyne detection. *Phys. Rev. Lett.*, 70(5):548–551, February 1993.

[83] Andrew C. Doherty, Salman Habib, Kurt Jacobs, Hideo Mabuchi, and Sze M. Tan. Quantum feedback control and classical control theory. *Phys. Rev. A*, 62:012105, June 2000.

[84] Matthew R. James and Hendra I. Nurdin. A tutorial introduction to quantum feedback control. In *Proc. 2015 IEEE Conference on Control Applications (CCA)*, pages 1–12, Sydney, NSW, September 2015. IEEE CCA.

[85] Shuang Cong. Numerical optimization methods. In *Control of Quantum Systems : Theory and Methods*, chapter 6, pages 221–310. Wiley, Singapore, 2014.

[86] 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.*, 172(2):296–305, February 2005.

[87] Simone Formentin, Klaske van Heusden, and Alireza Karimi. Model-based and data-driven model-reference control: A comparative analysis. In *Proc. 2013 European Control Conference (ECC)*, pages 1410–1415, Zurich, Switzerland, July 2013. IEEE.

[88] Alexey A. Melnikov, Hendrik Poulsen Nautrup, Mario Krenn, Vedran Dunjko, Markus Tiersch, Anton Zeilinger, and Hans J. Briegel. Active learning machine learns to create new quantum experiments. In *Proc. National Academy of Sciences*, volume 115, pages 1221–1226, Washington, DC, February 2018. National Academy of Sciences.

[89] Richard S. Sutton and Andrew G. Barto. Introduction. In *Reinforcement Learning: An Introduction*, Adaptive Computation and Machine Learning, chapter 1, pages 3–24. MIT, Massachusetts, 2 edition, 2017.

[90] Aleaxander Hentschel and Barry C. Sanders. Machine learning for precise quantum measurement. *Phys. Rev. Lett.*, 104(6):063603, February 2010.

[91] Michael T. Rosenstein and Andrew G. Barto. Robot weightlifting by direct policy search. In *Proc. 17th Int. Joint Conf. Artificial Intelligence*, volume 2 of *IJCAI'01*, pages 839–844, San Francisco, CA, 2001. Morgan Kaufman.

[92] Richard S. Sutton and Andrew G. Barto. The reinforcement learning problem. In *Reinforcement Learning: An Introduction*, Adaptive Computation and Machine Learning, chapter 3, pages 43–78. MIT, Massachusetts, 2 edition, 2017.

[93] Matthijs T.J. Spaan. Partially observable markov decision processes. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, volume 12 of *Adaptation, Learning, And Optimization*, chapter 12, pages 387–414. Springer, Berlin, 2012.

[94] Craig Boutilier and David Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Proc. 13th National Conf. Artificial intelligence (AAAI'96)*, volume 2, Palo Alto, CA, 1996. AAAI.

[95] Marco Wiering and Martijn van Otterlo. Evolutionary computation for reinforcement learning. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement learning: State-of-the-Art*, volume 12 of *Adaptation, Learning, And Optimization*, chapter 10, pages 325–355. Springer, Berlin, Germany, 2012.

[96] Tobias Glasmachers and Jürgen Schmidhuber. Optimal direct policy search. In Jürgen Schmidhuber, Kristinn R. Thórisson, and Moshe Looks, editors, *Artificial General Intelligence*, volume 6830 of *Lecture Notes in Computer Science*, pages 52–61, Berlin, Germany, 2011. Springer.

[97] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *Int. J. Rob. Res.*, 32(11):1238–1274, September 2013.

[98] Agoston E. Eiben and James E. Smith. Evolutionary computing: the origins. In *Introduction to Evolutionary Computing*, Natural Computing, chapter 2, pages 13–24. Springer, Berlin, Germany, 2 edition, 2015.

[99] Wojciech Wasilewski, Kristan. Jensen, Hanna Krauter, Jelmar J. Renema, Mikhail V. Balabas, and Eugene S. Polzik. Quantum noise limited and entanglement-assisted magnetometry. *Phys. Rev. Lett.*, 104(13):133601, March 2010.

[100] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96(1):010401, January 2006.

[101] Samuel L. Braunstein. Squeezing as an irreducible resource. *Phys. Rev. A*, 71(5):055801, May 2005.

[102] William K. Wootters. Quantum entanglement as a quantifiable resource. *Philos. Trans. Royal Soc. A*, 356(1743):1717–1731, August 1998.

[103] Malvin C. Teich and Bahaa E. A. Saleh. Squeezed state of light. *Quantum Opt.*, 1(2):153–191, 1989.

[104] Géza Tóth, Christian Knapp, Otfried Gühne, and Hans J. Briegel. Spin squeezing and entanglement. *Phys. Rev. A*, 79:042334, April 2009.

[105] Naeimeh Behbood, Ferran Martin Ciurana, Giorgio Colangelo, M. Napolitano, Géza Tóth, Robert J. Sewell, and Morgan W. Mitchell. Generation of macroscopic singlet states in a cold atomic ensemble. *Phys. Rev. Lett.*, 113:093601, August 2014.

[106] Robert McConnell, Hao Zhang, Jiazhong Hu, Senka Cuk, and Vladan Vuletic. Entanglement with negative wigner function of almost 3,000 atoms heralded by one photon. *Nature*, 519:439–442, March 2015.

[107] Nicolai Friis, Oliver Marty, Christine Maier, Cornelius Hempel, Milan Holzäpfel, Petar Jurcevic, Martin B. Plenio, Marcus Huber, Christian Roos, Rainer Blatt, and Ben Lanyon. Observation of entangled states of a fully controlled 20-qubit system. *Phys. Rev. X*, 8:021012, April 2018.

[108] Klaas Landsman. Quantum physics on a general hilbert space. In *Foundations of Quantum Theory: From Classical Concepts to Operator Algebras*, chapter 4, pages 103–123. Springer, Cham, Switzerland, 2017.

[109] Lorenzo Maccone. Intuitive reason for the usefulness of entanglement in quantum metrology. *Phys. Rev. A*, 88(4):042109, October 2013.

[110] T. J. Volkoff. Optimal and near-optimal probe states for quantum metrology of number-conserving two-mode bosonic hamiltonians. *Phys. Rev. A*, 94(4):042327, October 2016.

[111] Lorenzo Maccone and Giovanni De Cillis. Robust strategies for lossy quantum interferometry. *Phys. Rev. A*, 79(2):023812, February 2009.

[112] Brendon L. Higgins, Dominic W. Berry, Stephen D. Bartlett, Howard M. Wiseman, and Geoff J. Pryde. Entanglement-free Heisenberg-limited phase estimation. *Nature*, 450(7168):393–396, September 2007.

[113] Eric M. Kessler, Igor Lovchinsky, Alex O. Sushkov, and Mikhail D. Lukin. Quantum error correction for metrology. *Phys. Rev. Lett.*, 112:150802, April 2014.

[114] Alexander S. Holevo. Quantum evolutions and channels. In *Quantum Systems, Channels, Information: A Mathematical Introduction*, volume 16 of *De Gruyter Studies in Mathematical Physics*, chapter Quantum evolutions and channels, pages 103–131. Walter de Gruyter, Berlin, Germany, December 2012.

[115] William Gordon Ritter. Quantum channels and representation theory. *J. Math. Phys.*, 46(8):082103, August 2005.

[116] Masahito Hayashi. Mathematical formulation of quantum systems. In *Quantum Information: An Introduction*, chapter 1, pages 9–25. Springer, Berlin, Germany, 2006.

[117] Howard M. Wiseman and Gerard J. Milburn. *Quantum Measurement and Control*. Cambridge University Press, Cambridge, MA, 2009.

[118] Thomas A. Schonhoff and Arthur A. Giordano. Fundamental of estimation theory. In *Detection and Estimation Theory and its Applications*, chapter 10, pages 267–316. Pearson Prentice Hall, Upper Saddle River, NJ, 2006.

[119] B. Roy Frieden. What is fisher information. In *Physics from Fisher Information*, chapter 1, pages 22–50. Cambridge University Press, Cambridge, UK, 1998.

[120] Janek Kołodyński and Rafal Demkowicz-Dobrzański. Efficient tools for quantum metrology with uncorrelated noise. *New J. Phys.*, 15(7):073043, July 2013.

[121] Marcin Zwierz, Carlos A. Pérez-Delgado, and Pieter Kok. Ultimate limits to quantum metrology and the meaning of the Heisenberg limit. *Phys. Rev. A*, 85(4):042112, April 2012.

[122] Michael A. Armen, John K. Au, John K. Stockton, Andrew C. Doherty, and Hideo Mabuchi. Adaptive homodyne measurement of optical phase. *Phys. Rev. Lett.*, 89(13):133602, September 2002.

[123] Shibdas Roy, Ian R. Petersen, and Elanor H. Huntington. Robust adaptive quantum phase estimation. *New J. Phys.*, 17(6):063020, June 2015.

[124] H. M. Wiseman and R.B. Killip. Adaptive single-shot phase measurements: A semi-classical approach. *Phys. Rev. A*, 56(1):944–957, July 1997.

[125] H. M. Wiseman and R.B. Killip. Adaptive single-shot phase measurements: The full quantum theory. *Phys. Rev. A*, 57(3):2169–2185, March 1998.

[126] John A. Armstrong. Theory of interferometric analysis of laser phase noise. *J. Opt. Soc. Am.*, 56(8):1024–1031, August 1966.

[127] Parameswar Hariharan and Barry C. Sanders. Quantum phenomena in optical interferometry. In Emil Wolf, editor, *Progress in Optics*, volume XXXVI, chapter 2, pages 49–128. Elsevier, Amsterdam, Netherlands, 1996.

[128] Alexander Hentschel and Barry C. Sanders. Ordered measurements of permutationally-symmetric qubit strings. *J. Phys. A: Math. Theor.*, 44(11):115301, February 2011.

[129] Gil S. Summy and David T. Pegg. Phase optimized quantum states of light. *Opt. Commun.*, 77(1):75–79, June 1990.

[130] Michael I. Mishchenko. Wigner d-functions. In *Electromagnetic Scattering by Particles and Particle Groups: An Introduction*, chapter Appendix F, pages 385–389. Cambridge University Press, 2014.

[131] Barry C. Sanders and Gerard J. Milburn. Optimal quantum measurements for phase estimation. *Phys. Rev. Lett.*, 75(16):2944–2947, October 1995.

[132] K. F. Riley, M. P. Hobson, and S. J. Bence. Probability. In *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, chapter 30, pages 1119–1220. Cambridge University Press, Cambridge, UK, 3 edition, 2006.

[133] Zdeněk Hradil, R. Myška, Jan Peřina, Michael Zawisky, Yuji Hasegawa, and Helmut Rauch. Quantum phase in interferometry. *Phys. Rev. Lett.*, 76:4295–4298, June 1996.

[134] Samuel L. Braunstein. Some limits to precision phase measurement. *Phys. Rev. A*, 49:69–75, January 1994.

[135] Wolfgang von der Linden, Volker Dose, and Udo von Toussaint. Bayesian parameter estimation. In *Bayesian Probability Theory: Applications in the Physical Sciences*, chapter 14. Cambridge University Press, Cambridge, MA, 1 edition, June 2014.

[136] Howard M. Wiseman, Stefano Mancini, and Jin Wang. Bayesian feedback versus Markovian feedback in a two-level atom. *Phys. Rev. A*, 66:013807, July 2002.

[137] Rafal Demkowicz-Dobrzański, Marcin Jarzyna, and Janek Kołodyński. Quantum limits in optical interferometry. In Emil Wolf, editor, *Progress in Optics*, volume 60, chapter 4, pages 345 – 435. Elsevier, Amsterdam, 2015.

[138] Eugene A. Feinberg and Adam Shwartz. Introduction. In Eugene A. Feinberg and Adam Shwartz, editors, *Handbook of Markov Decision Processes: Methods and Applications*, International Series in Operations Research & Management Science, chapter 1, pages 1–17. Kluwer Academic, Norwell, MA, 2002.

[139] David J. Olive. Point estimation II. In *Statistical Theory and Inference*, pages 157–182. Springer, Cham, 2014.

[140] Pablo Zegers. Fisher information properties. *Entropy*, 17(7):4918, July 2015.

[141] Peter Winker and Manfred Gilli. Applications of optimization heuristics to estimation and modelling problems. *Comput. Stat. Data Anal.*, 47(2):211–223, 2004.

[142] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Singapore, 2006.

[143] Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11(4):341–359, December 1997.

[144] Swagatam Das and Ponnuthurai N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.*, 15(1):4–31, February 2011.

[145] Kenneth V. Price, Storn Rainer M., and Jouni A. Lampinen. The motivation for differential evolution. In Th. Bäck, J.N. Kok, and H.P. Spaink, editors, *Differential Evolution: A Pratical Approach to Global Optimization*, Natural Computing, chapter 1, pages 1–36. Springer, Berlin, 2005.

[146] Jakob Vesterstrom and René Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Proc. Congress on Evolutionary Computation, 2004 (CEC2004)*, volume 2, pages 1980 – 1987, Piscataway, New Jersey, June 2004. IEEE.

[147] Thiemo Krink, Bogdan Filipic, and Gary B. Fogel. Noisy optimization problems – a particular challenge for differential evolution? In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 1, pages 332–339. IEEE, June 2004.

[148] Pratyusha Rakshit, Amit Konar, and Swagatam Das. Noisy evolutionary optimization algorithms – a comprehensive survey. *Swarm and Evolutionary Computation*, 33:18–45, April 2017.

[149] Swagatam Das, Amit Konar, and Uday K. Chakraborty. Improved differential evolution algorithms for handling noisy optimization problems. In *Proc. 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1691–1698, Piscataway, New Jersey, September 2005. IEEE.

[150] Luigi Barone, Anthony Di Pietro, and Lyndon While. Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In G. Greenwood, editor, *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1254–1267, Oregon, June 2004. IEEE.

[151] Xin Yan and Xiao Gang Su. Simple linear regression. In *Linear Regression Analysis : Theory and Computing*, chapter 2, pages 9–40. SG: World Scientific, Singapore, 2009.

[152] Nicholas H. Bingham and John M. Fry. Linear regression. In *Regression: Linear Models in Statistics*, Springer Undergraduate Mathematics, chapter 1, pages 1–32. Springer, London, 2010.

[153] Michael Jay Quinn. *Parallel programming in C with MPI and openMP*. McGraw-Hill, Dubuque, IA, 1st edition, 2003.

[154] Kanti V. Mardia and Peter E. Jupp. Summary statistics. In *Directional Statistics*, Wiley Series in Probability and Statistics, chapter 2, pages 13–24. Wiley-Blackwell, May 2008.

[155] Bruno M. Escher, Ruynet L. de Matos Filho, and Luiz Davidovich. General framework for estimating the ultimate precision limit in noisy quantum-enhanced metrology. *Nat. Phys.*, 7:406–411, March 2011.

[156] Rafal Demkowicz-Dobrzanski, Janek Kołodynski, and Madalin Guţă. The elusive Heisenberg limit in quantum-enhanced metrology. *Nat. Commun.*, 3:1063, September 2012.

[157] Norman Bobroff. Residual errors in laser interferometry from air turbulence and nonlinearity. *Appl. Opt.*, 26(13):2676–2682, Jul 1987.

[158] Laura C. Sinclair, Fabrizio R. Giorgetta, William C. Swann, E. Baumann, Ian Coddington, and Nathan R. Newbury. Optical phase noise from atmospheric fluctuations and its impact on optical time-frequency transfer. *Phys. Rev. A*, 89(2):023805, February 2014.

[159] Thomas A. Severini. Central limit theorems. In *Elements of Distribution Theory*, Cambridge Series in Statistical and Probabilistic Mathematics, chapter 12, pages 365–399. Cambridge University Press, Cambridge, UK, 2005.

[160] Mohsen Pourahmadi. Skew-normal ARMA models with nonlinear heteroscedastic predictors. *Commun. Stat. Theory Methods*, 36(9):1803–1819, June 2007.

[161] Wassim Jouini. Energy detection limits under log-normal approximated noise uncertainty. *IEEE Signal Process. Lett.*, 18(7):423–426, July 2011.

[162] Javad Rezaie and Jo Eidsvik. Kalman filter variants in the closed skew normal setting. *Comput. Stat. Data Anal*, 75:1 – 14, July 2014.

[163] Thomas A. Severini. Moments and cumulants. In *Elements of Distribution Theory*, Cambridge Series in Statistical and Probabilistic Mathematics, chapter 4, pages 94–131. Cambridge University Press, Cambridge, UK, 2005.

[164] Les Kirkup. Data distributions i. In *Data Analysis for Physical Scientists: Featuring Excel®*, chapter 3, pages 90–145. Cambridge University Press, Cambridge, UK, 2 edition, March 2012.

[165] Charles Boncelet. Image noise models. In Al Bovik, editor, *Handbook of Image and Video Processing*, pages 397–409. Elsevier, Amsterdam, Netherlands, 2005.

[166] K. F. Riley, M. P. Hobson, and S. J. Bence. Special functions. In *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, chapter 18, pages 577–647. Cambridge University Press, Cambridge, UK, 3 edition, 2006.

[167] Eszter A. Kish, Claes-Göran Granqvist, András Dér, and Laszlo B. Kish. Lognormal distribution of firing time and rate from a single neuron? *Cogn. Neurodyn.*, 9(4):459–462, August 2015.

[168] Shoichi Kai, Seiji Higaki, Masahide Imasaki, and Hiroshi Furukawa. 1/f noise, log-normal distribution, and cascade process in electrical networks. *Phys. Rev. A*, 35(1):374–379, January 1987.

[169] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. Introduction. In *Introduction to Linear Regression Analysis*, Wiley Series in Probability and Statistics, chapter 1, pages 1–11. Wiley, Hoboken, NJ, 5 edition, 2012.

[170] Samprit Chatterjee and Jeffrey S. Simonoff. Model building. In *Handbook of Regression Analysis*, chapter 2, pages 23–52. Wiley, Hoboken, NJ, 2013.

[171] Samprit Chatterjee and Ali S. Hadi. Introduction. In *Regression Analysis by Example*, Wiley Series in Probablity and Statistics, chapter 1, pages 1–24. Wiley, Hoboken, NJ, 5 edition, 2012.

[172] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. Simple linear regression. In *Introduction to Linear Regression Analysis*, Wiley Series in Probability and Statistics, chapter 2, pages 12–66. Wiley, Hoboken, 5 edition, 2012.

[173] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. Variable selection and model building. In *Introduction to Linear Regression Analysis*, Wiley Series in Probability and Statistics, chapter 10, pages 327–371. Wiley, Hoboken, NJ, 5 edition, 2012.

[174] Samprit Chatterjee and Jeffrey S. Simonoff. Multiple linear regression. In *Handbook of Regression Analysis*, chapter 1, pages 3–22. Wiley, Hoboken, NJ, 2013.

[175] Samprit Chatterjee and Ali S. Hadi. Variable selection procedures. In *Regression Analysis by Example*, Wiley Series in Probablity and Statistics, chapter 11, pages 299–334. Wiley, Hoboken, NJ, 5 edition, 2012.

[176] Charles Jekel. Fitting a piecewise linear function to data. `https://jekel.me/2017/Fit-a-piecewise-linear-function-to-data/`, April 2017.

[177] Alexander Hentschel. *Particle Swarm Optimization for Adaptive Quantum-Enhanced Phase Estimation*. phdthesis, University of Calgary, Alberta, Canada, May 2011.

Appendix A

Modularized Policy-Search Code

This appendix is transcribed from the software documentation on GitHub as retrieved on August 9th, 2018 from `http://panpalitta.github.io/phase_estimation/`. This reproduction excludes the detail on the functions of each class.

A.1 Software information

A.1.1 Policy-search algorithm for adaptive phase estimation

We implement a policy-search algorithm to the problem of adaptive phase estimation, which is an example of quantum control problems. The aim of this project is to create a library containing modules that streamlines the construction of an optimization algorithm for quantum control problems. Access to modules of optimization algorithms provides the building blocks that users can use to tweak the algorithm to their needs.

A.1.2 Features:

- Library in C++
- Support MPI

- Support VSL and GPU for random number generation
- Include modules for particle swarm optimization (PSO) and differential evolution (DE)
- Include uniform and clustered method of initializing solution candidates
- Include access to user specified accept-reject criteria
- Include preliminary support for multi-objective calculation
- Support Autotools

A.1.3 Links:

- Latest release: 1.0.1 `https://github.com/PanPalitta/phase_estimation/releases`
- Feedback and issue: `https://github.com/PanPalitta/phase_estimation/issues`

A.1.4 Copyright and license

A.1.5 Acknowledgement

This software has been developed by Pantita Palittapongarnpim and Peter Wittek with the financial support of NSERC and AITF.

The computational work was enabled by support from WestGrid and Calcul Quebec through Compute Canada.

A.2 Download and installation

A.2.1 Download

- Latest release: download `https://github.com/PanPalitta/phase_estimation/releases`
- Development version: Github `https://github.com/PanPalitta/phase_estimation`
- Feedback and issues: issue `https://github.com/PanPalitta/phase_estimation/issues`

A.2.2 Compilation and installation

The project contains the support for compilation using Autotools, and has been tested using GNU Compile Chain (GCC) and Intel Compilers. The Intel VSL library and CUDA are automatically detected. An MPI implementation is required to compile and run the code.

If you cloned the Git repository, first run autogen.sh in order to create missing files and generate the executable configure from `configure.ac`.

Follow the standard POSIX procedure:

```
$ ./configure [options]
$ make
$ make install
```

To use the Intel compilers, set the following environment variables:

```
export CC=/path/of/intel/compiler/icc
export CXX=/path/of/intel/compiler/icpc
export OMPI_CC=/path/of/intel/compiler/icc
export OMPI_CXX=/path/of/intel/compiler/icpc
```

In order to use icc and icpc compilers, you have to set these variables so the mpic++ will invoke icpc instead of the default compiler.

Options for configure

```
--prefix=PATH           Set directory prefix for installation
--with-mpi=MPIROOT      Use MPI root directory.
--with-mpi-compilers=DIR or --with-mpi-compilers=yes
use MPI compiler (mpicxx) found in directory DIR, or in your PATH if =yes
--with-mpi-libs="LIBS"  MPI libraries [default "-lmpi"]
--with-mpi-incdir=DIR   MPI include directory [default MPIROOT/include]
--with-mpi-libdir=DIR   MPI library directory [default MPIROOT/lib]
```

The above flags allow the identification of the correct MPI library the user wishes to use. The flags are especially useful if MPI is installed in a non-standard location, or when multiple MPI libraries are available.

```
--with-cuda=/path/to/cuda           Set path for CUDA
```

The configure script looks for CUDA in `/usr/local/cuda`. If your installation is not there, then specify the path with this parameter. If you do not want CUDA enabled, set the parameter to `--without-cuda`.

```
--with-vsl=PATH    prefix where Intel MKL/VSL is installed
```

Specify the path to the VSL installation with this parameter.

A.3 Setting and Usage

A.3.1 Usage

The program is designed to work on HPC clusters and it requires MPI to run. The basic use is as follows:

```
$ [mpirun -np NPROC] phase_estimation [config_file]
```

A.3.2 Input and configuration

Arguments:

```
config_file              Configuration file name
```

If it is run without a configuration file, some default values are taken for all parameters; the exact settings are identical to the one in the provided `default.cfg` file. The configuration file is a plain text file with the name of the parameter on the left, followed by an equation sign surrounded by a space on either side, and a value on the right-hand side. For example, the contents of `default.cfg` are as follows:

```
pop_size = 20
N_begin = 4
N_cut = 5
N_end = 10
iter = 100
iter_begin = 300
repeat = 10
output_filename = output.dat
time_filename = time.dat
```

If you supply a configuration file, but do not set a specific value to every possible option, the default values are again the ones described in `default.cfg`.

The meaning of the individual parameters:

- `pop_size`: population size.
- `N_begin`: the starting number of particles.
- `N_cut`: the number of particles where the program use cluster initialization around previous solution.

- `N_end`: the final number of particles.
- `iter`: number of iterations when cluster initialization is used.
- `iter_begin`: number of iteration when uniformly random initialization is used.
- `repeat`: number of time the candidates are compute before the best candidate is selected after the optimization
- `optimization`: choose the heuristic optimization algorithm: de (differential evolution) or pso (particle swarm optimization)
- `output_filename`: the name of the file to write the results to.
- `time_filename`: the name of the file to write the time taken to run the program for each number of variables.
- `random_seed`: fix a random seed. If it is not specified, the random number generator is initialized with the system time
- `data_end`: set where accept-reject criterion based on error from expected solution is used
- `prev_dev`: the boundary for the cluster initialization – for variables that are initialized from previous solution
- `new_dev`: the boundary for the cluster initialization – for new variable
- `t_goal`: parameter corresponding to the error which the algorithm will accept to solution

Output

The program output two files, one containing the policy and fitness value (`output.dat`) and the other containing the CPU time used in finding the policy (`time.dat`). The numbers are updated for every N and can be used to track the progress of the optimization.

Example of output:

output.dat

```
1 #N   Sharpness   Policy
2 4   0.854507    4.91056 5.44221 5.73905 5.87313
3 5   0.888902    4.92427 5.46181 5.76412 5.88404 5.95493
4 6   0.90593 4.9164  5.46353 5.75503 5.89696 5.95538 6.05514
5 7   0.920337    4.91471 5.48735 5.72474 5.8765  5.94925 6.06185 6.14271
6 8   0.932388    4.88864 5.45983 5.72587 5.88542 5.94729 6.06073 6.12173 6.17251
7 9   0.941697    4.87251 5.45756 5.74001 5.90138 5.95748 6.04226 6.13899 6.16975 6.0799
8 10  0.941703    4.86374 5.42822 5.74102 5.87777 5.96852 6.04537 6.13999 6.13744 6.0704
```

time.dat

```
1 #N   Time
2 4   1
3 5   0
4 6   1
5 7   1
6 8   0
7 9   0
8 10  0
```

A.4 Expanding the library

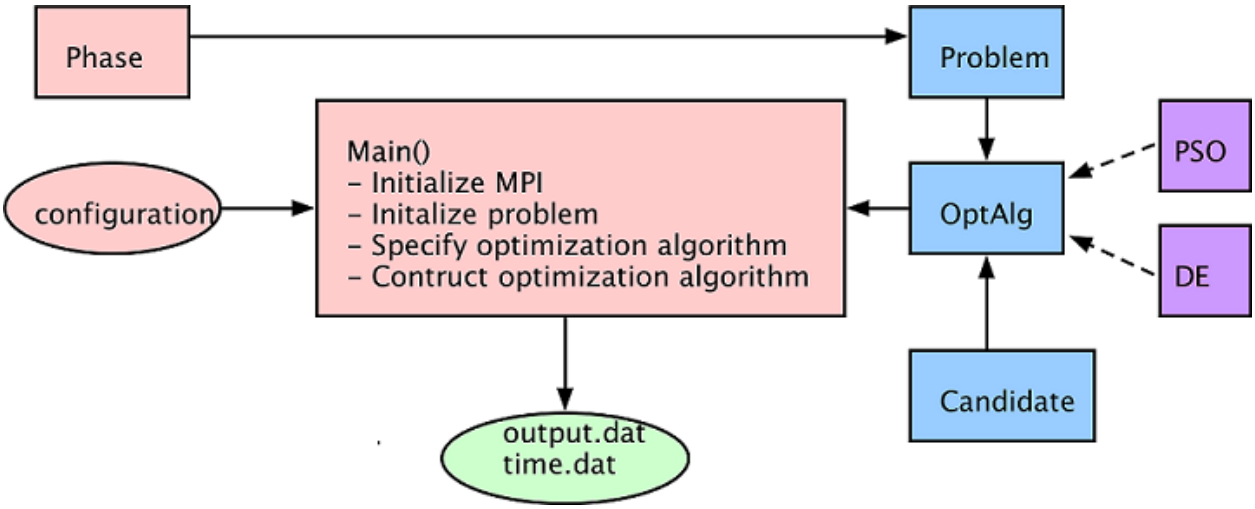
The intention of this project is to create a library that can be used for solving multiple quantum control problems. The code is therefore designed to ease the process of including new problems and algorithms and make the selection of problems and algorithm as error-free as possible. The following document is a guide on how users can write and include their

own problems and optimization algorithms to the existing library, and what is needed to customize and compile the code.

Readers are assumed to be familiar with population-based optimization algorithm, C++, object-oriented programming, class hierarchy and inheritance, and polymorphism.

A.4.1 Program structure

Diagram of program



User-specified components

The orange boxes correspond to the components in which the users specifies before the compiling the program. Phase class contains the modules for the adaptive phase estimation problem, which can be replaced with other problems. To select a problem of choice, replace `Phase()` by the constructor of the class in `main()` in the following line.

```
problem = new Phase(numvar, gaussian_rng, uniform_rng);
```

The Phase class is accessed through the Problem class. The pointer is given to the OptAlg class to be used for computing the fitness values and accept-reject criteria.

Optimization algorithms

The choice of optimization algorithm is specified in the configuration file along with other parameters shown as an orange oval in figure. Otherwise, it can also be coded in `main()` in the following line if necessary.

```
opt = new DE(problem, gaussian_rng, pop_size);
```

MPI

The MPI library is required for the program to run, as the program is designed to spread the solution candidate evenly on a group of processors. The processors communicates in the following situations.

- Constructing new set of candidates from existing population
- Finding the best candidate in the population or subset of the population
- Selecting the best candidate as solution

A.4.2 Add a new problem

A new problem should be written as a class derived from Problem class. There are five functions that user must include in the new problem.

- `fitness()` is a function intended to be a wrapper for changing conditions in which the fitness function is evaluated.
- `avg_fitness()` is the function for calculating the fitness value.
- `T_condition()` is a function for calculating additional conditions for when the optimization algorithm is set to accept solution after T iteration.
- `error_condition()` is a function for calculating additional conditions for when optimization algorithm is set to accept solution from error bound.

- `boundary()` is used to keep the solution candidate within the boundary of the search space.

This class does not use any MPI functionalities.

A.4.3 Add new algorithm

New algorithms can be added to the library of optimization algorithms by creating a derived class from the OptAlg class. The functions are designed based on swarm intelligence algorithms and evolutionary algorithms which shares the same backbone functions for initializing the population, for selecting the final solution candidate, and so forth. Aspects that are specific to the algorithm, such as how the new candidates are generated and selected, are declared as virtual function in OptAlg to allow the functions to be called from the derived class.

The functions, including virtual functions, are listed in OptAlg class document.

A.4.4 Constructing optimization algorithm

The library provides the module for users to contract the optimization algorithm in `main()`. The basic structure of the algorithm is given in `main.cpp`, which compile to the following structure.

- Initialize MPI and setting is compute to spread the number of candidates evenly on the processors
- Initialize problem and select the optimization algorithm
- Population is initialized using a user specified method
- The fitness values are computed and the population is prepared for the optimization
- The iterative optimization commences until the solution satisfied the specified criterion

- The program writes the fitness value, the solution, and the computational time as .dat files
- Program terminates

Most of these functionalities are in OptAlg class.

For adaptive phase estimation, the program runs many consecutive optimization problems with different number of variables N and the accept-reject criteria changes for different sets of N, which is possible by changing conditions given to the optimization algorithm in `main()`.

A.4.5 Changing the compilation setting

When a new problem and/or algorithm is included to the library, the following line in `src\Makefile.in`,

```
OBJS=main.o candidate.o phase_loss_opt.o io.o problem.o mpi_optalg.o
mpi_pso.o mpi_de.o candidate.o rng.o aux_functions.o,
```

should be updated to include the new class.

Appendix B

Copyright permissions

This appendix includes the relevant copyright permissions for published and submitted papers whose materials are included in this thesis. The publishers' policies granting permission for an author to include published materials in derivative works are given in the links below:

- Elsevier: `https://www.elsevier.com/about/policies/sharing`
- SPIE: `https://www.spiedigitallibrary.org/article-sharing-policies?SSO=1`
- IEEE: `https://ieeeauthorcenter.ieee.org/publishing-ethics/guidelines-and-policies/avoid-infringement-upon-ieee-copyright/`.

Copyright permission letters from co-authors of published and submitted papers are reproduced in the following pages.

From: Barry Sanders
Sent: January 22, 2019 3:41 PM
To: Pantita Palittapongarnpim
Subject: Re: Request permission to use copyrighted material in PhD thesis

Hi Pan, I approve of using all this copyrighted material material in your PhD thesis. Cheers, Barry

On Tue, 22 Jan 2019 at 15:40, Pantita Palittapongarnpim <[redacted]> wrote:

> Dear Dr. Barry C. Sanders,
>
> I am writing to you to ask for your permission to include materials from our co-written papers in my thesis, now called "Evolutionary Algorithm for Adaptive Quantum-Channel Control". The thesis will be added to the University of Calgary Theses Repository (http://theses.ucalgary.ca/) and the Library and Archives of Canada (http://collectionscanada.gc.ca/obj/s4/f2/frm-nl59-2-e.pdf).
>
> The materials I include in the thesis are from the papers listed below:
>
> 1. Figure 1, Figure 3, and excerpts of the text related to adaptive quantum-enhanced metrology and the algorithm used to generate successful policies for this project are taken from "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics" published in Neurocomputing vol. 268, pp.116-126 (2017), ISSN: 0925-2312, which is extended from "Controlling adaptive quantum phase estimation with scalable reinforcement learning" in the proceedings of 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 327--332 (2016), ISBN: 978-2-87587-026-1.
>
> 2. Figure 1, Figure 2, Figure 3, and excerpts of text from "Single-shot adaptive measurement for quantum-enhanced metrology" in the proceedings of SPIE Quantum Communications and Quantum Imaging XIV vol.9980, pp. 99800H (2016), ISSN: 1996-756X.
>
> 3. Figure 1, Figure 3, Table 1, and excerpts of text from "Robustness of learning-assisted adaptive quantum-enhanced metrology in the presence of noise" in the proceedings of 2017 IEEE International Conference on Systems, Man and Cybernetics (2017 SMC), pp. 294--299 (2017), ISBN: 978-1-5386-1645-1.

4. Figure 1, Figure 2, Table 1, Table 2, and excerpts of the text from "Robustness of Adaptive Quantum-Enhanced Phase Estimation" submitted to Physical Review A in 2018.

I hope to hear from you soon if you would allow me to use these materials in my thesis.

Sincerely,

Pantita Palittapongarnpim

PhD Candidate

Department of Physics and Astronomy

University of Calgary

AB, Canada

--

Barry C Sanders PhD DIC DSc FOSA FAPS FInstP FRSC http://iqst.ca/people/peoplepage.php?id=4
My free book *How to Build a Quantum Computer* http://iopscience.iop.org/book/978-0-7503-1536-4

From: Peter Wittek
Sent: January 22, 2019 6:18 AM
To: Pantita Palittapongarnpim
Subject: Re: Request permission to use copyrighted material in PhD thesis

Dear Pantita,

I happily consent to the inclusion of said material in your thesis.

Best regards,
Peter

On 21/01/2019 18:22, Pantita Palittapongarnpim wrote:
> Dear Dr. Peter Wittek,
>
>
>
> I am writing to you to ask for your permission to include materials from our co-written papers in my thesis, now called "Evolutionary Algorithm for Adaptive Quantum-Channel Control". The thesis will be added to the University of Calgary Theses Repository (http://theses.ucalgary.ca/) and the Library and Archives of Canada (http://collectionscanada.gc.ca/obj/s4/f2/frm-nl59-2-e.pdf).
>
>
>
> The materials I include in the thesis are from the papers listed below:
>
>
>
> 1. Figure 1, Figure 3, and excerpts of the text related to adaptive quantum-enhanced metrology and the algorithm used to generate successful policies for this project are taken from "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics" published in Neurocomputing vol. 268, pp.116-126 (2017), ISSN: 0925-2312, which is extended from "Controlling adaptive quantum phase estimation with scalable reinforcement learning" in the proceedings of 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 327--332 (2016), ISBN: 978-2-87587-026-1.
>
>
>
> 2. Figure 1, Figure 2, Figure 3, and excerpts of text from "Single-shot adaptive measurement for quantum-enhanced metrology" in the proceedings of SPIE Quantum Communications and Quantum Imaging XIV vol.9980, pp. 99800H (2016), ISSN: 1996-756X.
>
>
>
> 3. Figure 1, Figure 3, Table 1, and excerpts of text from "Robustness of learning-assisted adaptive quantum-enhanced metrology in the presence of noise" in the proceedings of 2017 IEEE International

Conference on Systems, Man and Cybernetics (2017 SMC), pp. 294--299 (2017), ISBN: 978-1-5386-1645-1.
>
>
>
> I hope to hear from you soon if you would allow me to use these materials in my thesis.
>
>
>
> Sincerely,
>
>
>
> Pantita Palittapongarnpim
>
>
>
> PhD Candidate
>
> Department of Physics and Astronomy
>
> University of Calgary
>
> AB, Canada
>
>
>

From: Seyed Shakib Vedaie
Sent: January 21, 2019 4:18 PM
To: Pantita Palittapongarnpim
Subject: Re: Request permission to use copyrighted material in PhD thesis

Dear Pantita,

I grant you permission to use the required materials in your thesis.

Best,
Shakib

On Mon, Jan 21, 2019, 4:11 PM Pantita Palittapongarnpim <[redacted]wrote:

> Dear Shakib Vedaie,
>
> I am writing to you to ask for your permission to include materials from our co-written papers in my thesis, now called "Evolutionary Algorithm for Adaptive Quantum-Channel Control". The thesis will be added to the University of Calgary Theses Repository (http://theses.ucalgary.ca/) and the Library and Archives of Canada (http://collectionscanada.gc.ca/obj/s4/f2/frm-nl59-2-e.pdf).
>
> The materials I include in the thesis are from the papers listed below:
>
> 1. Figure 1, Figure 3, and excerpts of the text related to adaptive quantum-enhanced metrology and the algorithm used to generate successful policies for this project are taken from "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics" published in Neurocomputing vol. 268, pp.116-126 (2017), ISSN: 0925-2312, which is extended from "Controlling adaptive quantum phase estimation with scalable reinforcement learning" in the proceedings of 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 327--332 (2016), ISBN: 978-2-87587-026-1.
>
> I hope to hear from you soon if you would allow me to use these materials in my thesis.
>
> Sincerely,

Pantita Palittapongarnpim

PhD Candidate

Department of Physics and Astronomy

University of Calgary

AB, Canada

From: ehsan zahedinejad
Sent: January 21, 2019 9:02 PM
To: Pantita Palittapongarnpim
Subject: Re: Request permission to use copyrighted material in PhD thesis

Dear Pantita,

I am happy to grant the permission.

Cheers,
Ehsan Zahedinejad.

On Mon, 21 Jan 2019 at 15:28, Pantita Palittapongarnpim <[redacted]> wrote:

> Dear Ehsan Zahedinejad,
>
> I am writing to you to ask for your permission to include materials from our co-written papers in my thesis, now called "Evolutionary Algorithm for Adaptive Quantum-Channel Control". The thesis will be added to the University of Calgary Theses Repository (http://theses.ucalgary.ca/) and the Library and Archives of Canada (http://collectionscanada.gc.ca/obj/s4/f2/frm-nl59-2-e.pdf).
>
> The materials I include in the thesis are from the papers listed below:
>
> 1. Figure 1, Figure 3, and excerpts of the text related to adaptive quantum-enhanced metrology and the algorithm used to generate successful policies for this project are taken from "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics" published in Neurocomputing vol. 268, pp.116-126 (2017), ISSN: 0925-2312, which is extended from "Controlling adaptive quantum phase estimation with scalable reinforcement learning" in the proceedings of 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2016), pp. 327--332 (2016), ISBN: 978-2-87587-026-1.
>
> I hope to hear from you soon if you would allow me to use these materials in my thesis.
>
> Sincerely,

Pantita Palittapongarnpim

PhD Candidate

Department of Physics and Astronomy

University of Calgary

AB, Canada

--
Ehsan Zahedinejad, **Machine Learning Researcher**
Bentall 5, Suite 458 | 550 Burrard St | Vancouver, BC | V6C 2B5
1QBit