

Characterization, Verification and Control for Large Quantum Systems

by

Christopher E. Granade

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics (Quantum Information)

Waterloo, Ontario, Canada, 2014

© Christopher E. Granade 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Quantum information processing offers potential improvements to a wide range of computing endeavors, including cryptography, chemistry simulations and machine learning. The development of practical quantum information processing devices is impeded, however, by challenges arising from the apparent exponential dimension of the space one must consider in characterizing quantum systems, verifying their correct operation, and in designing useful control sequences. In this work, we address each in turn by providing useful algorithms that can be readily applied in experimental practice.

In order to characterize the dynamics of quantum systems, we apply statistical methods based on Bayes' rule, thus enabling the use of strong prior information and parameter reduction. We first discuss an analytically-tractable special case, and then employ a numerical algorithm, sequential Monte Carlo, that uses simulation as a resource for characterization. We discuss several examples of SMC and show its application in nitrogen vacancy centers and neutron interferometry. We then discuss how characterization techniques such as SMC can be used to verify quantum systems by using credible region estimation, model selection, state-space modeling and hyperparameterization. Together, these techniques allow us to reason about the validity of assumptions used in analyzing quantum devices, and to bound the credible range of quantum dynamics.

Next, we discuss the use of optimal control theory to design robust control for quantum systems. We show extensions to existing OCT algorithms that allow for including models of classical electronics as well as quantum dynamics, enabling higher-fidelity control to be designed for cutting-edge experimental devices. Moreover, we show how control can be implemented in parallel across node-based architectures, providing a valuable tool for implementing proposed fault-tolerant protocols.

We close by showing how these algorithms can be augmented using quantum simulation resources to enable addressing characterization and control design challenges in even large quantum devices. In particular, we will introduce a novel genetic algorithm for quantum control design, MOQCA, that utilizes quantum coprocessors to design robust control sequences. Importantly, MOQCA is also memetic, in that improvement is performed between genetic steps. We then extend sequential Monte Carlo with quantum simulation resources to enable characterizing and verifying the dynamics of large quantum devices. By using novel insights in epistemic information locality, we are able to learn dynamics using strictly smaller simulators, leading to an algorithm we call quantum bootstrapping. We demonstrate by using a numerical example of learning the dynamics of a 50-qubit device using an 8-qubit simulator.

Acknowledgements

I would like to thank my supervisor, D. G. Cory, for his kind support throughout my graduate career, for insight and advice that always pointed me towards the interesting questions, and for the patience to let me see research through. I would also like to thank my parents and brother for helping to shoulder the burden during the hard time, and for sharing in the joy during the good. Much love to Rai, a friend that has stuck with me through all the changes that come with these years of study.

Special thanks as well to my colleagues that have been there for interesting research, clarifying conversations, and most importantly of all, friendship: Troy Borneman, Ben Criger, Yuval Sanders, Sarah Kaiser, Ian Hincks, Dan Puzzuoli, Christopher Wood, Holger Haas, and Razieh Annabestani. I am grateful for enlightening arguments with Christopher Ferrie, as well as for his constant support, and for helping to see so many projects through when I lost sight of the ending. I am humbled and honored by the constant friendship and patient advice given by Nathan Wiebe, who has helped me through every hardship along the way, and who has trusted in me when I needed it most. I am especially grateful and humbled by Robin Blume-Kohout and Scott Aaronson, who opened my eyes to the wonderful world of quantum information and who gave me a chance to shine.

Last but not least, thank you to Sarah Kavassalis, Blake Stacey, John Armstrong, and the rest of the online community. You have always been with me throughout my degree, and have helped me find my place in the world by challenging me with new perspectives, through thoughtful conversation and with understanding.

Dedication

To those that have put up with more than their fair share, and to those that find their own way through it all.

Table of Contents

List of Tables	x
List of Figures	xi
List of Algorithms	xviii
List of Source Code Listings	xix
Glossary	xxi
1 Introduction	1
1.1 Challenges for Large Quantum Systems	1
1.2 Small Quantum Devices and Co-Processors	3
1.3 Outline and Main Results	4
1.4 Introduction to Classical Statistics	6
1.4.1 Distributions, Expectations and Variances	6
1.4.2 Conditional Distributions, Marginalization and Bayes' Rule	8

1.5	Introduction to Quantum Mechanics	9
1.5.1	States, Measurements, Processes and Dynamics	9
1.5.2	The Stabilizer Formalism	12
1.5.3	Superoperators and Supergenerators	17
1.5.4	Error Correction and Stabilizer Codes	31
2	Characterization with Classical Resources	35
2.1	Bayesian Inference for Quantum Applications	36
2.1.1	Prior Information	37
2.1.2	Decision Theory	39
2.1.3	Sampling Periodic Distributions	42
2.2	Sequential Monte Carlo	46
2.2.1	Overview of Algorithm	46
2.2.2	Resampling	51
2.2.3	Adaptive Experiment Design	53
2.2.4	Hyperparameters and State-Space Methods	55
2.3	Examples of SMC	57
2.3.1	Robust Hamiltonian Learning	57
2.3.2	Randomized Benchmarking	66
2.3.3	Nitrogen Vacancy Centers	77
3	Verification of Quantum Systems	85
3.1	Region Estimation	85
3.1.1	Confidence and Credible Regions	86
3.1.2	Region Estimation with SMC	87
3.1.3	Hyperparameter Region Estimation	89
3.2	Model Selection	91

3.2.1	AIC, BIC and Bayes Factors	91
3.2.2	Bayes Factors with SMC	92
3.2.3	Applications to State-Space Models	93
4	Honest Approximation	100
4.1	Stabilizer Formalism as an Efficiently Simulatable Subtheory	102
4.2	Implementation of Honesty Approximation	102
4.3	Application to Error Correction	103
5	Control Design and Optimization	108
5.1	GRAPE and Conjugate-Gradient Methods	109
5.2	Optimal Control with Realistic Circuit Models	110
5.2.1	Distortions	110
5.2.2	Modeling Nonlinear Circuits	113
5.2.3	Robustness to Distributions	117
5.2.4	Heuristics for Pulse Desiderata	118
5.3	Composite Control Design for Node-Based Architectures	119
5.3.1	Wide Quantum Channels	121
6	Quantum Coprocessors and Semiquantum Algorithms	126
6.1	Control Design with Quantum Coprocessors	127
6.1.1	In-Place and Parallel Control Optimization Models	127
6.1.2	Robustness and Multi-Objective Optimization	129
6.1.3	Multi-Objective Memetic Algorithms	131
6.1.4	Single-Qubit Example of MOQCA	134
6.1.5	Future Improvements	137
6.2	Hamiltonian Learning with Quantum Resources	141

6.2.1	Design of Quantum Hamiltonian Learning Experiments	142
6.2.2	Ideal Performance and Scaling with Dimension	145
6.2.3	Robustness of QHL	148
6.3	Quantum Hamiltonian Learning with Truncated Simulation	159
6.3.1	Epistemic Information Locality	159
6.3.2	Learning Commuting Hamiltonians	161
6.3.3	Scanning and Global SMC Clouds	163
6.4	Quantum Bootstrapping	166
6.4.1	Control Characterization and Tuneup	167
6.4.2	Numerical Examples	168
7	Conclusions: Infrastructure for Large Quantum Devices	170
References		172
APPENDICES		206
A Likelihood-Free SMC		207
A.1	Weak and Strong Simulation	207
A.2	Robustness to Finite Sampling	210
B Bounds for Quantum Hamiltonian Learning		212
B.1	Sampling Error	212
B.2	Compressed Hamiltonian Learning with Non-commuting Hamiltonians .	213
C Source Code		216
D QInfer: Implementation of SMC		233
D.1	Design Considerations	233
D.2	Performance Testing	235

List of Tables

2.1	Results of using SMC and least-squares fitting to estimate the fidelity of $U = X$, simulated using the superconducting qubit gate set. (Left) Bad prior from Figure 2.19, (right) accurate prior from Figure 2.20.	76
4.1	Statistics for $\bar{\Lambda}_{\text{gadget}}$ using $N = 10^6$ randomly sampled pure states.	107
5.1	Values of parameters as defined by (5.9) for designing example pulses.	116
6.1	Parameters used in single-qubit MOQCA demonstration.	135
6.2	Error in compressed QHL for 50-qubit Ising model with an 8-qubit simulator, for each of several choices of observable size a . Reported values are of the 2-norm distance $\ \hat{x} - x\ _2$ after 500 experiments / scan.	165

List of Figures

1.1	Overview of using a parallel array of small quantum coprocessors.	4
1.2	Examples of Wood diagrams for vectorization and Roth's Lemma.	19
1.3	States on the Bloch sphere corresponding to vectors $\underline{r} = (0.7, 0, 0)$ (green) and $\underline{r} = (-1, 0, 1)/\sqrt{2}$ (orange).	20
1.4	Wood diagram for a superoperator representing preparation and measurement of an ancilla.	23
1.5	Circuits for producing Choi states $J(\Lambda)/d$.	24
2.1	Bayes risk and envelope for Larmor precession model, assuming normal prior $\omega \sim N(\mu, \sigma^2)$, with $\mu = 0.4$. (left) $\sigma^2 = 10^{-3}$, (right) $\sigma = 5 \times 10^{-5}$.	44
2.2	Flow of data and hypotheses in the SMC algorithm.	47
2.3	Flow of data through SMC algorithm for a periodic likelihood.	48
2.4	Two different representations of the multinormal distribution $N(0, \mathbb{I})$, using particle density (left) and particle weights (right).	50
2.5	Failure of Liu-West mode ($a^2 + h^2 = 1$) for bimodal posterior, contrasted with $a = 1, h > 0$ modification.	53
2.6	Decision process for adaptive experiment design.	54

2.7	Spectral density function used to generate realizations for state-space tracking example.	56
2.8	Tracking of a stochastic process, using a diffusive update in SMC. (Left) True and SMC-estimated trajectories. (Right) Error in SMC-estimated trajectory.	57
2.9	SMC performance for Larmor precession w/ known T_2 , vs. n_{mp} .	59
2.10	Comparison of likelihood calls per particle for different Larmor precession model experiment design strategies	60
2.11	Performance of SMC with the unknown- T_2 Larmor model.	62
2.12	Gates implementing correlated model of (2.34).	63
2.13	Mean quadratic loss and 99% confidence interval for the correlated precession model of (2.34) over 600 trials, versus the number of experiments N performed.	64
2.14	Experimental demonstration of referenced Rabi model (2.36).	65
2.15	Overview of randomized benchmarking.	67
2.16	Sketch of Magesan et al derivation of zeroth-order model for randomized benchmarking.	67
2.17	Optimal m for zeroth-order randomized benchmarking.	72
2.18	Bayes risk of SMC-accelerated randomized benchmarking, compared to BCRB, covariance and LSF Bayes risk.	74
2.19	Comparison of prior, SMC and LSF estimates for inaccurate randomized benchmarking prior.	75
2.20	Comparison of prior, SMC and LSF estimates for accurate randomized benchmarking prior.	75
2.21	Nitrogen vacancy in a diamond lattice.	78
2.22	Schematic of the energy levels of a nitrogen vacancy center, showing pumping laser and state-dependent spontaneous emissions between the ground state $ g\rangle$ and excited $ e\rangle$ manifolds. The zero-field $\Delta_{\text{zfs}}S_z^2$ and Zeeman $\gamma_e B_z S_z$ splittings are illustrated in the ground-state manifold.	80
2.23	Estimated Rabi power curves as a function of normalized DAC output amplitude and phase, along with a 68% credible interval on each estimate.	83

2.24	Simulation of Rabi and Ramsey experiments for hyperfine estimation in nitrogen vacancy centers.	84
3.1	Credibility approximated by a covariance ellipse of the SMC posterior for the known- T_2 Larmor precession model, compared to the credibility for a true normal distribution.	88
3.2	Excess covariance of the hyperregion estimator for the Gaussian-Larmor hyperparameter model (2.29), using 2,000 SMC particles, and the prior $(\mu, \sigma^2) \sim \mathcal{N}([0.5, 0.0025], [0.001^2, 0.0025^2])$	90
3.3	Schematic of a three-blade neutron interferometer with a controllable phase flag and a sample inserted in one arm. Detectors are placed at the ends of the two interferometric paths, and are called the ``O-beam'' and the ``H-beam'' by convention.	93
3.4	Example data collected from a three-blade neutron interferometer, shown with an SMC fit to the static model.	94
3.5	History of the estimator $\hat{\phi}$ of the phase difference between two arms of a three-blade interferometer as a function of the number of data points considered. (Left) No stochasticity assumed, (Right) $\phi = \phi(t)$. Shaded bands indicate the standard deviation of the posterior over ϕ	95
3.6	Logarithm of the Bayes factor between the static and stochastic models for the three-blade neutron interferometer. Negative values correspond to rejection of the static model in favor of the stochastic model.	96
3.7	Example of reference trajectories as estimated by parameter-space maximum-likelihood estimator and the state-space models of (3.12) and (3.13).	98
3.8	Total likelihoods for nitrogen-hyperfine coupling experiments, using each of several different referencing models and the same physical model.	99
4.1	Circuit layout for extracting behavior of error-correction round $\bar{\Lambda}_{\text{gadget}}$	104
4.2	Fowler schedule for applying CNOT gates to measure XXXX syndromes in a surface or lattice code.	104
4.3	Syndrome measurement with and without explicit wait locations.	106
5.1	Example of an exponential convolution kernel $g_*[\underline{p}] = e^{-t/t_c}/t_c * \underline{p} = \underline{q}$ applied as a distortion operator, where $t_c = 500$ ps.	112

5.2	Parallel one-port <i>RLC</i> circuit with non-linear inductance and resistance and matching capacitance.	114
5.3	Example of a 200 ns, 7 V square pulse along x after being transmitted through a nonlinear resonator with $\alpha_I = 0.0006/\text{A}^2$ and $\alpha_R = 0$	115
5.4	Example pulse implementing $\pi/4)_x$ for a nonlinear resonator.	116
5.5	Example of a robust pulse for the nonlinear resonator.	118
5.6	Geometry for layered lattice or sublattice codes, using node-based geometries.	120
5.7	Example of two nodes in an e/n node-based architecture.	122
5.8	State manifold for electron/nuclear example.	123
5.9	Walsh-Hadamard functions on 8 dimensions, ranked in increasing ``sequency'' order.	124
5.10	Suppression of cross couplings in a wide quantum channel.	125
6.1	Use of parallel quantum coprocessors to evaluate pulse fidelities.	128
6.2	Projection onto the plane of the Pareto optimal solutions to the trivial objective $\varrho(r) = r$ under the constraint $\ r\ \leq 1$	130
6.3	Two-point crossover operator between two distinct pulse individuals, with randomly chosen cut points $t_{\text{cx},1}$ and $t_{\text{cx},2}$. All pulse steps between the cut points are swapped between the two individuals.	134
6.4	Example pulse generated MOCQA with 200 generations, a population of 140 pulses each no more than 100 ns in length, and using the MOQCA parameters in Table 6.1.	136
6.5	Number of non-dominated individuals found by the MOQCA algorithm as a function of generations.	137
6.6	Estimated infidelities $1 - \hat{\Phi}$ for each hypothesis about H_{int} as a function of the number of MOQCA generations.	138
6.7	Histogram of 2-norm infidelities across the Pareto front after 60 MOQCA generations.	139
6.8	Projection of final population and Pareto front infidelities after 60 MOQCA generations onto 10 planes through infidelity space (\mathbb{R}^5). The lower-left corner of each projection is the ideal target pulse.	140

6.9	Flow chart for quantum Hamiltonian learning with interactive likelihood evaluation and particle guess heuristic.	143
6.10	Experiment and simulator design for non-interactive and interactive quantum likelihood evaluation.	144
6.11	The median quadratic loss plotted as a function of the number of (left) non-interactive and (right) interactive QLE experiments, learning Ising models on the line graph for each of several numbers of qubits n . The shaded areas show a 50% confidence interval for the quadratic loss. 10 000, 10 000 and 20 000 particles were used in the $n = 4$, $n = 8$ and $n = 12$ cases respectively.	146
6.12	The median quadratic loss plotted as a function of the number of interactive QLE experiments, learning Ising models on the complete graph for each of several numbers of qubits n . The shaded areas show a 50% confidence interval for the quadratic loss.	147
6.13	Scaling for four-qubit (red) and six-qubit (blue) spin chain models, where each model admits a single parameter that is much less certain than the others. The black lines illustrate that the exponential rate at which errors decay is dramatically different once the uncertain parameter ``catches up,''' such that the scaling switches to the $n = 4$ and $n = 6$ rates.	148
6.14	The median decay exponent for the quadratic loss as a function of the number of parameters in the Ising model d , and for each of the complete and linear interaction graphs.	149
6.15	The median quadratic loss for a 9 qubit Ising model on the line for the IQLE and QLE cases with 10% ALE tolerance for estimating likelihoods, compared to the case of IQLE with perfect likelihood estimation. 10 000 particles were used for the learning algorithm in all cases.	150
6.16	Median value of the error decay rate γ computed for IQLE experiments in which the interaction graph is a line, and where n ranges from 1 to 12 with ALE tolerance $\epsilon = 0.4/n$. For each experiment, learning was performed using 20 000 SMC particles. For comparison, the infinite-sampling limit $\epsilon = 0$ is also plotted.	151
6.17	The median quadratic loss, plotting for 200 IQLE experiments with a line-graph Ising Hamiltonian on four qubits, and with varying levels of depolarizing noise p_{depol} .	153

6.18	The median value of the error decay rate γ found by fitting the quadratic loss of 200 random line-graph Ising Hamiltonians to $Ae^{-\gamma N}$, rescaled by the asymptotic behavior predicted by (6.9), and where $d = \dim \underline{x}$ is the number of model parameters. The dashed lines represent a 50% confidence interval for the data, and the crosses and circles correspond to $p_{\text{depol}} = 0.75/d$ and $p_{\text{depol}} = 0.5/d$ respectively.	154
6.19	Median value of the loss incurred by quantum Hamiltonian learning w/ IQLE in estimating the J -coupling between two qubits as a function of the number of measurements performed, for each of five physical models.	155
6.20	The performance of QHL for the case where the trusted simulator uses an Ising model on the line given that the true Hamiltonian is an Ising model on the complete graph with non-nearest neighbor interactions on the order of 10^{-4} and nearest neighbor interactions on the order of 0.5.	157
6.21	The logarithm of the posterior odds ratio of the true model to the reduced model, as analyzed by interactive QHL. The shaded regions include all trials.	158
6.22	Light cones for $A(t)$ for a single step of an r step protocol. The green region is the light cone after the evolution in the untrusted device, and the blue region is after inversion in the trusted device. The dashed lines show the spread of $A(t)$ due to inexact inversion in the trusted simulator.	160
6.23	Separation of $H_{\text{int}} = H_{\text{int} \cap A} + H_{\text{int} \setminus A}$ where $H_{\text{int} \cap A}$ are interactions with qubits in the support of A (red solid box) and $H_{\text{int} \setminus A}$ interacts with qubits that are swapped into the trusted simulator but are outside A (blue dashed box).	162
6.24	Scanning procedure for 7 qubits, a 4 qubit simulator and a 2 qubit observable. Blue (dashed) box is support of simulator, red (solid) box is support of A	164
6.25	Error in compressed QHL for $a = 4$ with varying N_{exp} per scan. Data consistent with $e^{-0.006N_{\text{exp}}}$ scaling.	166
6.26	Distribution of errors for each of the 49 Hamiltonian terms in the bootstrapped Hamiltonian for a 50 qubit Ising model using (left) 100 (right) 200 and (bottom left) 300 IQLE experiments per scan.	169
A.1	Inputs and outputs for weak and strong simulations of the same likelihood function.	208

A.2	Comparison of different extremes of likelihood-free sampling strategies.	209
A.3	Risk incurred by likelihood-free SMC for the noisy-coin photodector model versus the number of weak simulations m per particle.	211
A.4	Risk incurred by likelihood-free SMC for the noisy-coin photodetector model versus adaptive likelihood estimation tolerance ϵ	211
B.1	Schematic of repeated swaping between untrusted device and trusted simulator, showing regions of support W , X and Y for the simulator, observable and complement of the simulator, respectively. Boxes indicate Hamiltonian evolution under H or H_- for the untrusted and trusted devices.	214
D.1	Performance testing of serial and parallel CPU-only implementations versus GPU-only implementation of the Larmor precession model (2.11). The perfomance is measured as total elapsed time in seconds, normalized by the number of likelihood calls required.	236

List of Algorithms

1	Sequential Monte Carlo update algorithm.	51
2	Liu and West resampling algorithm.	52
3	Complete adaptive Bayesian experiment design algorithm, using sequential Monte Carlo approximations.	58
4	GRAPE: Conjugate Gradient optimization with Khaneja gradient calculation.	111
5	SPSA algorithm for local improvement.	132

List of Source Code Listings

1.1	Bell state description as a stabilizer state	15
1.2	Example of vectorization in QuTiP [1].	18
1.3	Source code for above figure, using QuTiP.	20
1.4	Calculation of state preparation superoperator using QuTiP.	22
1.5	Column-stacking partial trace superoperators in Mathematica.	23
1.6	Computation of different superoperator representations.	26
1.7	Explicit enumeration of $N(S)/S$ for the five-qubit perfect code.	33
2.1	Calculation of achievable risk in a zeroth-order randomized benchmarking experiment.	70
2.2	Calculation of m with optimal Fisher information for interleaved randomized benchmarking.	71
2.3	Processing randomized benchmarking data with QInfer.	77
C.1	Source code for Figure 2.5.	216
C.2	QInfer implementation of simple precession model of (2.11), with full metadata and Fisher score calculation.	217
C.3	Source code for Figure 2.8.	220
C.4	QInfer implementation of Fisher score for the randomized benchmarking model (2.46).	221
C.5	Example of a Rabi model with bright/dark referencing.	222
C.6	Implementation of Floquet-Leskes expansion of the stroboscopic Hamiltonian for a nitrogen vacancy center coupled to a carbon nucleus via a hyperfine interaction.	229

C.7 Generation of Walsh-Hadamard basis in sequency order. 231

Glossary

automorphism ($\text{Aut}(G)$) An isomorphism from a group G to itself. The automorphisms of G form a group under function composition. [16](#)

Bayes risk The expected risk for a given estimator, taken over a prior . [40](#), [43](#), [53](#), [59](#), [73](#)

Clifford group (C_n) The group of operations that map elements of the Pauli group to the Pauli group. That is, the group of automorphisms of the Pauli group.. [66](#), [102](#)

conditionally independent ($a \perp b \mid c$) a and b are said to be independent conditioned on c if $\Pr(a|b,c) = \Pr(a|c)$.

confidence region (\hat{X}_α) A region such that, for all hypotheses \underline{x} , the probability of obtaining data yielding a region containing \underline{x} is at least α .

CPTP Completely positive and trace preserving; a set of conditions on a **superoperator** that ensures that the represented map takes valid states to valid states . [38](#)

credible region (\hat{X}_α) A region containing the true value of a random variable with probability at least α . [49](#), [65](#)

data (D) A collection of experimental observations $D = \{d_1, \dots, d_k\}$, typically used to estimate parameters of or select models for that experiment.

dominates ($p \blacktriangleright q$) If, for every **fitnesss** component f_i , $f_i(p) > f_i(q)$, p dominates q . [129](#)

estimate A random variate drawn from an estimator, given some data . [39](#), [49](#), [85](#)

estimator A function from data to estimate . [36](#), [39](#), [40](#), [49](#)

factor group (G/H) A group of *cosets* $gH := \{gh : h \in H\}$ of H within G . [14](#)

fitness ($f(I)$) A vector of selection criteria used in evaluating whether a given individual is propagates to the next generation. [xxi](#)

group A semigroup G that is associative and such that each element has an inverse . [13](#)

group action (\bullet) A binary operation $g \bullet s$ for g an element of a group G and s an element of a set S , such that $g_1 g_2 \bullet s = g_1 \bullet g_2 \bullet s$ and $e \bullet s$ for e the identity. [14](#)

Hamiltonian (H) Hermitian operator that generates unitary dynamics.

homomorphism A function $\phi : G \rightarrow H$ for groups G and H such that for all elements $g, g' \in G$, $\phi(g)\phi(g') = \phi(gg')$ symbol.

honest An approximation to a process is honest if it does not underrepresent the errors in the unapproximated process . [89](#)

hyperparameters (y) A vector of parameters that describes a distribution over model parameters. [9](#), [89](#)

individual (I) A possible solution to an optimization problem under consideration by a genetic algorithm.

isomorphism A homomorphism that is invertible .

likelihood function ($\Pr(d|\underline{x}; e)$) A probability distribution that describes the sampling of data d , conditioned on a hypothesis about the **model parameters** \underline{x} and a set of experimental controls e . [12](#), [36](#), [46](#), [68](#)

loss function A function of estimate and hypotheses that describes how bad an estimate is at representing that hypothesis . [39](#), [40](#)

model parameters (\underline{x}) A vector that specifies a particular instance from a class of models; for instance, a paramterization of a Hamiltonian. [xxii](#), [38](#), [40](#), [55](#), [60](#), [89](#)

Pauli group (\mathcal{P}_n) The group generated by the Pauli operators $\sigma_x, \sigma_y, \sigma_z$, along with phases of the identity \mathbb{I} . [102](#)

posterior ($\Pr(\underline{x}|D)$) Distribution encoding the state of knowledge about a system after some data record has been collected. [46](#)

prior (π) Distribution encoding the state of knowledge about a system before any data is collected. [37](#), [40](#)

quadratic loss (L_Q) Multiparameter generalization of the mean squared error; see [Definition 5](#).

risk The expected loss function for a given estimator, taken over all data . [39](#), [40](#), [91](#)

semigroup A semigroup is a set G with a binary associative operator $\cdot : G \times G \rightarrow G$ such that G is closed under \cdot . [15](#)

stabilizer group (S) A group of elements that leaves each member of a set invariant under the appropriate group action.

state ($|\psi\rangle$) Vector describing the probability of measurements on a quantum system. [9](#), [12](#)

strong simulation Evaluation of the likelihood function for a simulated system . [207](#)

subgroup (\subseteq) A group that is also a subset of another group. [13](#), [15](#)

superoperator (\hat{S}) A linear operator acting on $L(\mathcal{H})$, the space of linear operators on a Hilbert space \mathcal{H} . [xxi](#), [17](#), [21](#)

unitary (U) An operator $U \in L(\mathcal{H})$ is unitary if $UU^\dagger = U^\dagger U = \mathbb{1}$. The group of unitary operators acting on \mathcal{H} is denoted $Uni(\mathcal{H})$. [11](#)

weak simulation Sampling from the likelihood function for a simulated system . [207](#)

word A sequence of generators from the presentation of a group . [66](#), [67](#)

1 Introduction

1.1 Challenges for Large Quantum Systems

Over the past century, the ubiquity of information processing devices has transformed society in a myriad of ways. A considerable fraction of the world's energy budget is now spent on performing computational tasks, both on devices owned by individuals and on consolidated servers. The sheer scale of modern computational tasks thus highlights the difficulty intrinsic to finding solutions to many interesting problems with information processing devices. Though efforts have been taken to explore new and interesting computing architectures that reduce the energy and time costs in information processing, such as massively parallel computation with general purpose graphics processors [2] or integrated field-programmable gate arrays [3], these advances do not change the *scaling* of the problems that we wish to solve or approximately solve.

This apparent difficulty is made more concrete by the extended Church-Turing thesis, which conjectures that any physical computational device can be efficiently simulated by a mathematical model known as a *Turing machine* [4]. Under this conjecture, at least some computational problems that are intractable to modern information processors will remain intractable regardless of architectural and algorithmic advances. The extended Church-Turing thesis is challenged, however, by the observation that quantum systems can solve some computational tasks with better asymptotic scalings than any known classical algorithms. Recent work in linear optics [5], for example, has strongly

suggested that there exist probability distributions that can be efficiently sampled using quantum devices, but that do not admit tractable classical sampling algorithms.

In particular, Feynman famously observed [6] that quantum systems can in principle efficiently simulate other quantum systems, in spite of the apparent intractability for classical simulators. This insight suggests that by exploring a different set of computational axioms than is formalized in Turing machines, we can dramatically expand the set of problems that we can solve within accessible energy and time budgets.

In the decades following this observation, new advances in *quantum computing* have demonstrated a wide range of problems that can be accelerated by quantum information processing (QIP) devices. For example, important problems in cryptography [7], chemistry simulation [8], machine learning [9; 10; 11] and data analysis [12] have all been shown to admit quantum algorithms. Acceleration of machine learning methods in particular offer dramatic impact on society, given the ``web-scale'' application of ML to problems as wide ranging as recommendation engines [13], computer vision [14], security analysis [15], human/computer interaction [16], genome analysis [17] and economic forecasting [18]. Quantum algorithms thus offer improvements to a very wide range of computational tasks, such that building useful quantum information processing devices that can exploit these new algorithms is of immediate importance.

The task of building, controlling and debugging such devices presents difficult problems, however. Engineering quantum devices, for instance, is notoriously difficult, spurring the investigation of many different physical modalities such as electron spin resonance in organic systems [19], superconducting qubits [20], quantum dots [21], ion trap arrays [22], and nitrogen vacancy centers in diamond [23]. Designing control based on explicit models of quantum dynamics is made difficult due to the cost of simulating these models, and due to the cost of initially characterizing the system being controlled. Though closed-loop solutions based on genetic algorithms [24] and stochastic optimization [25] have been proposed to address these difficulties, we are still left with the difficulty of asserting that control has been implemented correctly [26]. Thus, even once built, controlling, characterizing and validating quantum information devices are all difficult problems that demand new solutions.

In addition to accelerating information processing tasks, quantum information theory has also enabled new advances in *metrology*, such that small quantum devices can be deployed as sensors that outperform classical sensing devices [27; 28]. Quantum metrology is enabled by engineering systems whose dynamics depend on a parameter of interest, such that the same characterization techniques that we develop for controlling quantum systems immediately offer parallel advances in the development of new

quantum sensing devices.

1.2 Small Quantum Devices and Co-Processors

In addition to solving the problems frustrating the development of large quantum resources, we are also interested in applying small quantum devices that can feasibly be experimentally realized within the next few years. Some quantum chemistry problems, for instance, can enjoy superclassical advantages with as few as 100 logical qubits [29].

Another compelling application of small quantum devices is found in the characterization and control of other quantum systems. Modern classical computers, by analogy, are not designed and tested manually, but with the assistance of the previous generations of information processors. Making this analogy concrete, we propose three main applications for a dedicated, small quantum device:

Characterization/Verification Small quantum devices can be used to learn properties of large quantum devices, most notably their dynamical generators (Hamiltonians).

Calibration Small quantum devices can be used to test for and learn calibration errors that, if left uncorrected, can lead to undesirable control terms, such as non-local couplings introduced by correlations in control lines.

Control Design Small quantum systems can also be used to design control sequences (pulses) for other devices that implement desired unitary operations (gates).

This last application in particular, explored in detail in [Section 6.1.3](#), demonstrates the application of using small quantum devices in a *quantum coprocessor* architecture. That is, much of the logic of control design and characterization algorithms can be made classical, leaving the use of quantum simulation as a ``black-box'' to which these classical algorithms have access. For instance, in [Figure 1.1](#), we illustrate the concept for a typical arrangement, in which a classical host queries the performance of a candidate control sequence, using a parallel array of quantum coprocessors in order to predict the performance on a system of interest.

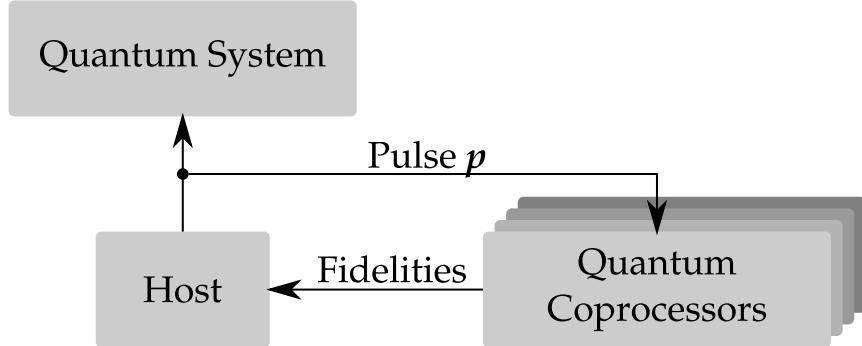


Figure 1.1: Overview of using a parallel array of small quantum coprocessors to evaluate the fidelity of a control pulse under different hypotheses about the true evolution of a system of interest. A classical host computer is used to design input and apply control to the quantum coprocessors. A specific algorithm is introduced in [Chapter 6](#) to utilize this arrangement.

1.3 Outline and Main Results

In this work, we focus on this problem of characterizing and controlling quantum systems in four distinct parts. In [Chapter 2](#), we explore novel characterization techniques based on classical simulation of quantum systems and demonstrate their utility to small quantum devices within current experimental capabilities. Next, in [Chapter 3](#), we show how characterization methods can also be applied to *verify* the correct operation of quantum systems. We then show in [Chapter 4](#) how to derive approximate models for larger systems that honestly capture the errors in a more complete description. In [Chapter 5](#), we discuss problems with controlling quantum systems using classical resources alone and introduce new design techniques to reduce the dimension in which control must be designed, as well as methods for incorporating realistic models of classical devices into quantum control algorithms.

Finally, in [Chapter 6](#), we discuss how to apply these ideas in large quantum systems by the use of small quantum coprocessors. By expressing the challenges of the previous chapters in terms of simulation, we are able to augment both optimal control ([Section 6.1.3](#)) and Hamiltonian learning ([Section 2.2](#)) algorithms with quantum resources in a natural way. In doing so, we also introduce *quantum bootstrapping*, in which small quantum resources are used to characterize, calibrate and verify larger quantum devices. Together, these provide a path to the development of useful and practical quantum de-

vices well beyond the scale that can be simulated classically.

The results presented here are based on a wide range of publications and collaborations under the supervision and advice of D. G. Cory:

- Work in Bayesian inference for the characterization of quantum systems is based on [30; 31], joint work with Christopher Ferrie.
- Application of the sequential Monte Carlo (SMC) algorithm to problems in Hamiltonian learning is based on [32], joint work with Christopher Ferrie and Nathan Wiebe. State-space applications represent previously unpublished work by the current author.
- Application of the sequential Monte Carlo algorithm to randomized benchmarking is based on [33], joint work with Christopher Ferrie.
- All experiments with nitrogen-vacancy centers in diamond represent previously unpublished work, joint with Ian Hincks, Fei Wang, Maryam Mirkamali and Osama Moussa.
- The use of SMC for region estimation and model selection is based on [32; 34], joint work with Christopher Ferrie and Nathan Wiebe, and includes a summary of related work by Christopher Ferrie [35; 36].
- Model selection in state-space models represents unpublished work, joint with Christopher Wood and Dimitri Pushin (neutron interferometry), and with Ian Hincks (nitrogen-vacancy centers).
- The honest approximation of quantum channels is based on [37; 38], joint work with Daniel Puzzuoli, Easwar Magesan, Ben Criger and Holger Haas.
- The extension of optimal control theory to include nonlinear models is based on [39], joint work with Ian Hincks and Troy Borneman.
- The design of composite pulse sequences for parallel control (wide quantum channels) is based on [40], joint work with Troy Borneman.
- The MOQCA algorithm for quantum-accelerated optimal control is unpublished work by the current author, with kind advice from Nathan Wiebe, Christopher Ferrie and Joshua Combes.

- Quantum Hamiltonian learning is based on [41; 34], joint work with Nathan Wiebe and Christopher Ferrie.
- The extension of quantum Hamiltonian learning to provide a quantum bootstrap-ping algorithm using information locality is based on [10], joint work with Nathan Wiebe.

Many of these results are presented along with example software implementations, making it easier to quickly integrate the methods presented here into experimental practice. In particular, we focus on the following software libraries:

QuantumUtils for Mathematica [42] Mathematica library for quantum information. Developed by the author, in collaboration with Christopher Wood and Ian Hincks.

QuantumUtils for MATLAB [43] MATLAB-language library for quantum information. Developed by Ian Hincks.

QInfer [44] Python library for statistical inference in quantum applications, using the sequential Monte Carlo algorithm. Developed by the author in collaboration with Christopher Ferrie, and with kind contributions by Ian Hincks and Yuval Sanders.

QuaEC [45] Python library for quantum error correction. Developed by the author in collaboration with Ben Criger.

QuTiP [1] Python library for manipulating and simulating quantum dynamics. Developed by P. D. Nation and J. R. Johansson, including contributions from the author.

1.4 Introduction to Classical Statistics

Quantum mechanics is inherently a *statistical* theory, such that before delving into quantum mechanics and quantum information processing, we must first introduce the terminology and concepts that we will later rely upon.

1.4.1 Distributions, Expectations and Variances

In this work, we will consider many different examples of *random variables* following probability distributions describing our lack of prior certainty about the outcome of

some event. A random variate is then a specific value taken on by a random variable. For instance, if we are to roll a die, we do not generally know *a priori* what the outcome will be; it is this very property that makes dice fun! We can, however, reason about what proportion of the time we will observe certain values. In this example, once we have rolled a die, the number that we read off of it is the random variate d , and the probability that we assign to each possible face of the die gives the distribution $\Pr(d)$.

Having defined such a distribution, we can now ask what will usually happen when we draw samples from it. Formally, one way of addressing this question is to consider the average over the distribution, also known as the expectation value. The expectation value $\mathbb{E}[x]$ of a random variable x is defined as

$$\mathbb{E}[x] := \sum_x x \Pr(x), \quad (1.1)$$

where the sum is taken over all possible values of the random variable, known as the support of x , $\text{supp}(x)$. We can also consider the expectation over functions of a random variable,

$$\mathbb{E}[f(x)] := \sum_x f(x) \Pr(x). \quad (1.2)$$

When the summation variable is not immediately clear from context, we will denote it as a subscript to the expectation operator. In the equation above, we write \mathbb{E}_x , for instance.

One very important function that we will often consider is the *variance* of a random variable,

$$\mathbb{V}[x] := \mathbb{E}[(x - \mathbb{E}[x])^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2. \quad (1.3)$$

The square root of the variance is called the *standard deviation*, and gives insight into how uncertain the outcome of a random variable is.

More generally, we can consider random variables such that the expectation and variance are now given by integrals,

$$\mathbb{E}[f(x)] := \int_x f(x) \Pr(x) \, dx. \quad (1.4)$$

In this case, the function $\Pr(x)$ is now a *density*, such that we must integrate over a range of values to obtain a probability. For instance, we shall later demand that we can find regions X such that

$$\Pr(x \in X) := \int_{x \in X} \Pr(x) \, dx \geq \alpha \quad (1.5)$$

for some value α . That is, that the probability of drawing an x that lies within X is at least some level α . Throughout the rest of this work, we will drop the measure dx when it is clear from context, as we will be primarily concerned with numerical implementations where the measure is fixed implicitly by the choice of parameterization.

In this work, we will mainly be concerned with random variables that are vector-valued. In this case, the expectation and variance are both also vector-valued,

$$\mathbb{E}[\underline{x}]_i = \mathbb{E}[x_i] \tag{1.6a}$$

$$\mathbb{V}[\underline{x}]_i = \mathbb{V}[x_i]. \tag{1.6b}$$

We can also consider the covariance matrix, which describes the degree to which the variation in one component of a vector-valued random variable depends on another component. That is, we write

$$\text{Cov}(\underline{x}) := \mathbb{E}[\underline{x}\underline{x}^T] - \mathbb{E}[\underline{x}]\mathbb{E}[\underline{x}]^T. \tag{1.7}$$

As we will see later, in [Section 3.1](#), for distributions that are approximately Gaussian (also known as normal), the covariance matrix is a compact description of the regions in which random variates of \underline{x} are most likely to be located.

1.4.2 Conditional Distributions, Marginalization and Bayes' Rule

Often, we will be interested in two or more random variables at the same time. For instance, it is particularly common to consider both experimental data d and a description \underline{x} of the experimental system as random variables. In this context, we cannot draw experimental data without also considering the system that we draw the data from. In particular, if we consider the experimental device itself as being drawn from an ensemble of similar devices, then the description of a particular device is not initially known, but rather is exposed by the dependence of d on \underline{x} .

In terms of distributions, we make this formal by considering that two random variables A and B are *jointly* distributed as $\Pr(A, B)$. For a given random variate drawn from A , we can *condition* on that hypothesis,

$$\Pr(A|B) := \frac{\Pr(A, B)}{\Pr(B)} = \frac{\Pr(A, B)}{\mathbb{E}_A[\Pr(B|A)]}. \tag{1.8}$$

The distribution $\Pr(B)$ that appears in the denominator is known as a *marginal distribution*, and is formed by taking an expectation value over the marginalized random variables.

Expectation values and variances of conditional distributions can be expressed in terms of marginal distributions by using the laws of total expectation and variance, respectively given by

$$\begin{aligned}\mathbb{E}[A] &= \mathbb{E}_B[\mathbb{E}_{A|B}[A|B]] = \sum_A A \Pr(A|B) \Pr(B) \\ \text{Cov}(A) &= \mathbb{E}_B[\text{Cov}_{A|B}(A|B)] + \text{Cov}_B(\mathbb{E}_{A|B}[A|B]).\end{aligned}\tag{1.9}$$

As we will see in [Section 2.2.4](#) and [Section 3.1](#), the law of total variance has a nice interpretation in terms of [hyperparameters](#).

Conditional distributions tell us about the way that two random variables are related, and what information each carries about the other. Of particular note is that if we expand [\(1.8\)](#) the other way to obtain $\Pr(B|A)$, we obtain *Bayes' rule*,

$$\Pr(A|B) = \frac{\Pr(B|A)}{\Pr(B)} \Pr(A).\tag{1.10}$$

Effectively, Bayes' rule allows for inverting the conditioning of distributions. This is especially useful in the case of *learning*, as we can invert a conditional distribution such as $\Pr(d|x)$ to obtain a distribution $\Pr(x|d)$ that describes our uncertainty about a device or system of interest.

While many other statistical approaches to learning exist, we focus here on the use of Bayes' rule, as it nicely allows for the incorporation of prior information into inference procedures.

1.5 Introduction to Quantum Mechanics

Before proceeding to address the challenges of characterizing, controlling and verifying large quantum systems, it is helpful to first establish some background and common notation for quantum mechanics itself, as well as several key formalisms built on top of quantum theory. Here, we will quickly describe a few key areas along with useful software resources for further exploration.

1.5.1 States, Measurements, Processes and Dynamics

In quantum mechanics, measurements on a system are described in terms of a [state](#) vector $|\psi\rangle$ in some Hilbert space \mathcal{H} . Throughout this work, we will assume that \mathcal{H} is the

finite dimensional space $\mathcal{H} = \mathbb{C}^d$ of d -dimensional vectors containing complex numbers. In particular, we will write $|i\rangle$ to be the elementary vector with a 1 in the i^{th} component and zeros everywhere else. The *basis* of such vectors, $\{|i\rangle : i \in \{0, 1, \dots, d-1\}\}$ is called the computational basis. In the special case that $d = 2$, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ describes the state of a *qubit*, the generalization of a classical bit to the full space of allowed states in quantum mechanics. In matrix representation, we write that

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1.11)$$

We will also denote the set of linear operators acting on these states as $L(\mathcal{H})$.

The duals (conjugate transposes) $\langle\phi|$ of state vectors describe measurements, such that the inner product $\langle\phi|\psi\rangle$ is the probability *amplitude* for observing $|\phi\rangle$ if we have prepared a state $|\psi\rangle$. The probability is then given in terms of the probability amplitude by Born's rule [46]

$$\Pr(\phi|\psi) = |\langle\phi|\psi\rangle|^2 = \langle\phi|\psi\rangle\langle\psi|\phi\rangle. \quad (1.12)$$

Here, we have implicitly used the symbols ψ and ϕ to label preparations and measurements, respectively, as random variables, such that we reason about what the outcome of a measurement will be *conditioned* on having performed a given preparation procedure $|\psi\rangle$.

This description then suggests that instead of deterministically preparing a state $|\psi\rangle$, we can instead have a *mixture* such that $|\psi\rangle$ is a random variable drawn from the ensemble $\{(p_i, |\psi_i\rangle)\}_i$. That is, that $|\psi\rangle$ is chosen to be $|\psi_i\rangle$ with probability p_i . Then, marginalizing over Born's rule for this ensemble gives us that

$$\Pr(\phi|\psi) = \mathbb{E}_i[\Pr(\phi|\psi_i)] = \langle\phi| \left(\sum_i p_i |\psi_i\rangle\langle\psi_i| \right) |\phi\rangle. \quad (1.13)$$

We then identify $\sum_i p_i |\psi_i\rangle\langle\psi_i|$ as a useful object in its own right, the density operator ρ . More generally, density operators are trace-1 positive semidefinite operators acting on \mathcal{H} , $\rho \in L(\mathcal{H})$, representing mixtures of pure preparations.

By the von Neumann equation, the state of a quantum system evolves in time according to the differential equation

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

where the Hamiltonian $H \in \text{Herm}(\mathcal{H})$ is a Hermitian operator acting on a Hilbert space \mathcal{H} that describes the dynamics of the system. This differential equation is then solved by the operator $U = e^{-iHt}$,

$$\rho(t) = U\rho(0)U^\dagger, \quad (1.15)$$

where U^\dagger is the Hermitian conjugate (conjugate transpose) of U .

For any Hamiltonian, the resulting evolution operator U satisfies $UU^\dagger = U^\dagger U = \mathbb{1}$. Operators are called unitary; the set of all such operators for a given Hilbert space is written $U(\mathcal{H})$. Since we will commonly consider conjugation by an operator, we use a shorthand notation $U \bullet \rho := U\rho U^\dagger$. This will be especially useful later, when we discuss group actions, as \bullet can then be discussed as a function $\bullet : U(\mathcal{H}) \rightarrow (\mathcal{L}(\mathcal{H}) \rightarrow \mathcal{L}(\mathcal{H}))$ in its own right ¹.

With these definitions in mind, we can now formally describe quantum mechanics in terms of *postulates* about the allowable states and measurements [47].

Postulate 1 (States) *The state of a quantum system is described by a density operator $\rho \in \mathcal{L}(\mathcal{H})$, a trace-1 positive semidefinite linear operator acting on the Hilbert space \mathcal{H} .*

Postulate 2 (Evolution of States) *Closed systems evolve under the action of **unitary** operators $U(t_2, t_1) \in U(\mathcal{H})$,*

$$\rho(t_2) = U(t_2, t_1) \bullet \rho(t_1) = U(t_2, t_1)\rho U(t_2, t_1)^\dagger. \quad (1.16)$$

Postulate 3 (Measurement) *A measurement of a quantum system is described by a positive operator-valued measure (POVM); that is, a set of operators $M = \{E_i\}$ such that $\sum_i E_i^\dagger E_i = \mathbb{1}$ and such that each E_i is positive semidefinite. The probability of observing outcome i from a measurement of M is then given by*

$$\Pr(i|\rho, M) = \text{Tr}(E_i \rho E_i^\dagger), \quad (1.17)$$

¹Note that we have *curried* the definition of this group action, such that $U \bullet$ ``returns'' a function from $\mathcal{L}(\mathcal{H})$ to itself. This is equivalent to the perhaps more familiar definition $\bullet : U(\mathcal{H}) \times \mathcal{L}(\mathcal{H}) \rightarrow \mathcal{L}(\mathcal{H})$, under the celebrated isomorphism $\text{Hom}(X \times Y, Z) \cong \text{Hom}(X, \text{Hom}(Y, Z))$. That is, all two-argument functions can be thought of as one-argument functions which return one-argument functions.

In this case, our choice of definition will be useful in [Section 1.5.3](#), where we identify $U \bullet$ as being an important special case of a more general kind of object called a *superoperator*, which maps linear operators to other linear operators.

and the post-measurement state ρ' is given by Lüder's rule,

$$\rho' = \frac{E_i \rho E_i^\dagger}{\text{Tr}(E_i \rho E_i^\dagger)}. \quad (1.18)$$

Postulate 4 (Composition) *The joint state space of two or more quantum systems, having individual state spaces $\{\mathcal{H}_1, \dots, \mathcal{H}_n\}$ is given by the tensor product $\mathcal{H} = \bigotimes_i \mathcal{H}_i$.*

In order to predict and model measurements of a quantum system, we therefore must know the [state](#) of the system. Since the space of allowable states of quantum systems compose under the tensor product, the dimension of the Hilbert space on which ρ acts grows exponentially with the size of a system. This makes it exceedingly difficult to learn the state of a quantum system unless we take a statistical approach [48].

For small systems, however, reconstructing the complete state from measurements is feasible. Quantum state tomography [49; 50; 51; 52] reconstructs ρ from a measurement record drawn from an *informationally-complete* POVM (IC-POVM) [53; 54; 55; 56]. Tomography has been used in a wide range of experimental contexts, and for a wide variety of tasks, such as showing fidelity of preparation, entanglement, etc. [57; 21].

More explicitly statistical approaches to learning states, processes and Hamiltonians can be enabled by interpreting Born's rule (1.12) as a [likelihood function](#) that depends on a hypothesis. Conditioning the likelihood function on a hypothetical state, for example, has allowed for applying both maximum-likelihood estimation [57; 58; 52] and Bayesian estimation [51; 59] to the problem of learning states.

In this work, we put special emphasis on the problem of learning Hamiltonians, as accurate knowledge about the dynamics of a quantum system is a critical resource for the development and implementation high-fidelity control. The problem of characterizing dynamical generators has previously been considered, using assumptions such as locality [60], and complete positivity [61]. We improve upon previous work by demonstrating a *numerical* algorithm that can be readily extended to consider a wide range of experimental concerns, and that, as discussed at length in [Chapter 6](#), serves as a platform for incorporating quantum resources.

1.5.2 The Stabilizer Formalism

Here, we quickly review the stabilizer formalism, and its application to quantum error correction, along with examples of software implementations using the QuaEC library

for Python [45]. Because the stabilizer formalism is expressed in the language of group theory, we first recall the definitions of **group** and **subgroup**.

Definition 1 (Group) A group (G, \cdot) is a set G and a binary operation $\cdot : G \times G \rightarrow G$ such that the following properties hold:

1. The set G is closed under \cdot ; that is, for any $g, h \in G$, $g \cdot h \in G$.
2. There exists an identity element $e \in G$ such that for all $g \in G$, $e \cdot g = g \cdot e = g$.
3. The binary operation is associative, such that for all $a, b, c \in G$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
4. Each element g of G has an inverse $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = e$.

For brevity, we omit the operation in the definition of a group when this is implied by context. If a subset $H \subseteq G$ is also a group under \cdot , then H is a subgroup of G , denoted by $H \leq G$.

For example, the set of unitary operators $U(\mathcal{H})$ on a Hilbert space \mathcal{H} is a group under matrix multiplication, as every unitary operator U has an inverse U^\dagger . If we restrict the determinants to $\det U = 1$, then we obtain the special unitary group $SU(\mathcal{H}) \leq U(\mathcal{H})$. In particular, the states of a qubit transform under $SU(\mathbb{C}^2)$ as global phases are irrelevant. This group is more commonly written as $SU(2)$.

A particularly important example of unitary transformations is that of the Pauli matrices,

$$\underline{\sigma} = (\mathbb{1}, \sigma_x, \sigma_y, \sigma_z) \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.19)$$

The Pauli matrices form a basis for $L(SU(2))$, so that any linear operator A acting on the state space of a qubit can be written as

$$A = a\mathbb{1} + b\sigma_x + c\sigma_y + d\sigma_z \quad (1.20)$$

for some $a, b, c, d \in \mathbb{C}$. If A is Hermitian, then $a, b, c, d \in \mathbb{R}$, giving a real parameterization of A . For this reason, the Pauli matrices are an especially important subset of $U(2)$. In the interest of brevity, we will sometimes write $X = \sigma_x$, $Y = \sigma_y$ and $Z = \sigma_z$, or will index the Pauli matrices by integers, $\{0, 1, 2, 3\}$.

These matrices are quite useful in quantum information, as they exhibit a number of very nice properties beyond being a convenient basis for $L(\mathbb{C}^2)$. For instance, $\sigma_\mu \sigma_\nu$ is, up

to a phase, also a Pauli matrix for any $\mu, \nu \in \{0, 1, 2, 3\}$, such that the Pauli matrices form a subgroup of $SU(2)$. In addition to representing the Pauli group as a subgroup of the unitary matrices, we can easily define the Pauli group \mathcal{P}_n on n qubits directly in terms of its *generators*. That is, a set of elements such that all other elements of the group can be written as products of the generators. We use the notation $\langle \cdot \rangle$ to indicate that all products of the contained elements are to be included. For example, $\langle a, b \rangle = \{e, a, b, aa, ab, ba, \dots\}$, where $e = aa^{-1}$.

Definition 2 (Pauli group) *The Pauli group on n qubits is the group*

$$\mathcal{P}_n := \langle i\mathbb{1}, X_1, \dots, X_n, Z_1, \dots, Z_n \rangle, \quad (1.21)$$

where X_i is the n -qubit operator that acts as X on the i^{th} qubit and as $\mathbb{1}$ on all other qubits, and where Z_i is defined similarly. The weight $\text{wt}(P)$ of an element P of the Pauli group is the number of qubits on which P acts nontrivially.

We will also consider the *factor group* $\hat{\mathcal{P}}_n := \mathcal{P}_n / \langle i\mathbb{1} \rangle$ that disregards phases. That is, $\hat{\mathcal{P}}_n$ is the group of sets $\{P, iP, -P, -iP\}$ for $P \in \mathcal{P}_n$. For notational convenience, we will often denote elements of $\hat{\mathcal{P}}_n$ by a single representative from each such set.

Writing the Pauli group in this way emphasizes that we need not keep track of the matrix form of each element, but can write down a string of operators such as $YZYX = \sigma_y \otimes \sigma_z \otimes \sigma_y \otimes \sigma_x$. Each such Pauli group element can be recorded on a classical qubit using $2n + 2$ bits, as opposed to the $O(2^{2n})$ bits required to record a matrix representation.

In QuaEC, the $+1$ representatives of the elements of $\hat{\mathcal{P}}_n$ can be enumerated as a Python iterator²:

```
1 >>> import qecc as q
2 >>> print list(q.pauli_group(2))
[ i^0 II, i^0 IX, i^0 IY, i^0 IZ, i^0 XI, i^0 XX, i^0 XY, i^0 XZ,
  i^0 YI, i^0 YX, i^0 YY, i^0 YZ, i^0 ZI, i^0 ZX, i^0 ZY, i^0 ZZ]
```

The elements of the Pauli group admit a *group action* $\bullet : \mathcal{P}_n \rightarrow (\mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n})$ on the Hilbert space \mathbb{C}^{2^n} of n qubits. In particular, the Pauli group acts on states by left multiplication,

$$P \in \mathcal{P}_n, |\psi\rangle \in \mathbb{C}^{2^n} : P \bullet |\psi\rangle := P|\psi\rangle. \quad (1.22)$$

²Line breaks have been added to the listing for the sake of readability.

We shall also write the group action of \mathcal{P}_n on density operators ρ as $P \bullet \rho := P\rho P^\dagger$, by analogy to the group action of $U(\mathbb{C}^{2^n})$ on density operators. This notation has the advantage that transformations are specified in the same way for pure and mixed states.

We say that a state $|\psi\rangle$ is *stabilized* by an operator P if $P \bullet |\psi\rangle = |\psi\rangle$. For a given state, the elements which stabilize that state form a **semigroup**, since $PQ \bullet |\psi\rangle = P \bullet (Q \bullet |\psi\rangle) = P \bullet |\psi\rangle = |\psi\rangle$ for any operators P and Q which each individually stabilize $|\psi\rangle$. Moreover, since we draw the operators from the Pauli group, the stabilizer semigroup is also a group. We can then readily extend this definition to *sets* of states.

Definition 3 (Stabilizer group) *Given a set of states $\{|\psi_i\rangle\} \subseteq \mathbb{C}^{2^n}$, the stabilizer group $\text{stab}(\{|\psi_i\rangle\})$ is the set of elements P of \mathcal{P}_n such that $P \bullet |\psi\rangle = |\psi\rangle$ for all $|\psi\rangle \in \{|\psi_i\rangle\}$.*

A famous result of Gottesman [62] is that the subgroups of \mathcal{P}_n which are stabilizer groups for some set of states can be completely characterized.

Theorem 1 *A subgroup $S \leq \mathcal{P}_n$ of the Pauli group \mathcal{P}_n is a stabilizer group for a set of states if and only if S is an Abelian group and $-\mathbb{1} \notin S$.*

Importantly, this means that we can *specify* some states by listing the group that stabilizes them. Such states are known as *stabilizer states*. For instance, the Bell state $|\beta_{00}\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is the unique state stabilized by $\langle XX, ZZ \rangle = \{\mathbb{1}\mathbb{1}, XX, -YY, ZZ\}$. Using QuaEC, we can readily calculate that this is the case.

Listing 1.1: Bell state description as a stabilizer state.

```

1 >>> import qecc as q
>>> stab = q.StabilizerGroup(['XX', 'ZZ'], [], [])
>>> print stab.stabilizer_subspace()
array([[ 0.70710678+0.j,  0.00000000+0.j,  0.00000000+0.j,
        0.70710678+0.j]])
```

The dimension of the space of states stabilized by a subgroup $S \leq \mathcal{P}_n$ is reduced by a factor of two for each independent generator, such that n such generators are needed to stabilize a unique state.

Since elements of the Pauli group \mathcal{P}_n can each be identified on a classical computer using $2(n+1)$ bits of data, this implies that a pure stabilizer state can be identified using classical resources that are polynomial in the number of qubits. To complete the specification of the stabilizer formalism as an efficiently simulatable subtheory, we then need to show that transformations between stabilizer states are also classically tractible.

Considering transformations of the Pauli group and its subgroups, then, we define the *Clifford group* as the group of **automorphisms**, noting that every such automorphism also has a representation as a unitary operator.

Definition 4 (Clifford group) *Given the Pauli group \mathcal{P}_n on n qubits, the Clifford group \mathcal{C}_n is $\mathcal{C}_n := \text{Aut}(\mathcal{P}_n)$, the group of automorphisms of \mathcal{P}_n . Equivalently, the Clifford group is the group of operators that map Pauli group elements into the Pauli group,*

$$\mathcal{C}_n := \left\{ U \in \text{U}(\mathbb{C}^{2^n}) \mid \forall P \in \mathcal{P}_n : U \bullet P \in \mathcal{P}_n \right\}. \quad (1.23)$$

Because each element of \mathcal{C}_n is an automorphism of \mathcal{P}_n , its action on \mathcal{P}_n can be specified by listing the action on a presentation of \mathcal{P}_n in terms of a generating set. One particularly useful such set is that given in (1.21), $\{X_1, \dots, X_n, Z_1, \dots, Z_n\}$. This is the representation used by QuaEC. For instance, the controlled-NOT gate (also known as a CNOT gate) is the gate which maps $X\mathbb{1} \mapsto XX$ and $\mathbb{1}Z \mapsto ZZ$, leaving the other two generators invariant:

```

1 >>> print q.cnot(2, 0, 1)
XI | -> +XX
IX | -> +IX
ZI | -> +ZI
IZ | -> +ZZ

```

Two other important gates are the Hadamard and phase gates:

```

>>> print q.hadamard(1, 0)
X | -> +Z
Z | -> +X
>>> print q.phase(1, 0)
5 X | -> +Y
Z | -> +Z

```

These three gates are actually a generating set for the Clifford group,

$$\mathcal{C}_n = \langle \text{CNOT}_{i,j}, H_i, P_i \rangle_{i,j \in \{1, \dots, n\}, i \neq j}. \quad (1.24)$$

We now have everything we need to offer the intuition behind the Gottesman-Knill theorem [63]: since a Clifford group element maps Pauli elements to Pauli elements and can be described using no more than $2n \times 2(n+1)$ classical bits, the action of a circuit composed entirely of Clifford gates, stabilizer preparations and Pauli measurements can be simulated classically using polynomially many resources in the number of qubits.

1.5.3 Superoperators and Supergenerators

We now take a somewhat different tack, and describe how to reason about *open* quantum systems. That is, those systems whose dynamics are uncertain or that are coupled to an environment that we cannot directly measure or observe. In doing so, we rely on the superoperator formalism, which utilizes a larger Hilbert space to represent open processes and dynamics. We shall also show examples of *Wood diagrams*, a graphical calculus for manipulating the open quantum system objects [64].

1.5.3.1 Liouville Space

Consider a state ρ that is subject to one of two unitary operators, U or V , based on the outcome of a coin flip. We can describe this by the map Λ

$$\Lambda : \rho \mapsto \frac{1}{2}U \bullet \rho + \frac{1}{2}V \bullet \rho, \quad (1.25)$$

since the resulting density operator should be a *mixture* of the two possibilities. This follows by the same argument as (1.13), marginalizing instead over our choice of unitary operator.

This seems somewhat unwieldy, however, in that to specify the operation that acts on ρ , we naïvely seem to have to include ρ itself. Note, however, that $\frac{1}{2}U \bullet$ is a *linear* function, in that

$$\frac{1}{2}U \bullet (\rho + \sigma) = \frac{1}{2}U(\rho + \sigma)U^\dagger = \frac{1}{2}(U\rho U^\dagger + U\sigma U^\dagger) = \frac{1}{2}U \bullet \rho + \frac{1}{2}U \bullet \sigma. \quad (1.26)$$

Because the group action of U on density operators is linear, we can then represent it by a linear operator acting on a vector space, $\hat{U}[\rho] := U \bullet \rho$. In particular, \hat{U} acts on the vector space of density operators,

$$\hat{U} |\rho\rangle\langle\rho| = |U \bullet \rho\rangle\langle\rho|, \quad (1.27)$$

where $|\rho\rangle\langle\rho|$ represents a *vectorization* of the density operator ρ . We can now express the map of (1.25) as the convex combination of two **superoperators** \hat{U} and \hat{V} ,

$$\hat{S}_\Lambda = \frac{1}{2}\hat{U} + \frac{1}{2}\hat{V}, \quad (1.28)$$

such that the action of Λ on ρ is given by $\hat{S}_\Lambda |\rho\rangle\langle\rho|$.

Concretely, any density operator ρ on a Hilbert space \mathcal{H} is an element of the set $L(\mathcal{H})$ of linear operators acting on \mathcal{H} . This set is a vectorspace with the Hilbert-Schmidt inner product $\langle A, B \rangle := \text{Tr}(A^\dagger B)$, and can be written as the span of *elementary* operators $\underline{E}_{ij} := |i\rangle\langle j|$ for $i, j \in \{1, \dots, \dim \mathcal{H}\}$. Accordingly, we can assign a vector to each such basis element to represent $L(\mathcal{H})$. One convention for doing so is the row-stacking convention that

$$|\underline{E}_{ij}\rangle\rangle = ||i\rangle\langle j|\rangle\rangle = |i\rangle\otimes\langle j|^T. \quad (1.29)$$

For 2×2 matrices,

$$\left| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \right\rangle\rangle = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \quad (1.30)$$

thus motivating the name: the rows of the matrix are transposed and stacked to form the corresponding vector. The column-stacking convention $||i\rangle\langle j|\rangle\rangle = \langle j|^T \otimes |i\rangle$ is also quite common, and is used by quantum information software packages such as QuTiP [1] and QuantumUtils/MATLAB [43]. Importantly, either isomorphism preserves the inner product,

$$\text{Tr}(A^\dagger B) = \sum_i \langle i | A^\dagger B | i \rangle = \sum_{ij} (A^\dagger)_{ij} B_{ji} = \sum_{ij} \overline{A}_{ji} B_{ji} = \langle\langle A | B \rangle\rangle. \quad (1.31)$$

Listing 1.2: Example of vectorization in QuTiP [1].

```
>>> from qutip import *
>>> print operator_to_vector(Qobj([[1, 2], [3, 4]]))
Quantum object: dims = [[[2], [2]], [1]], shape = [4, 1], type = operator-ket
4 Qobj data =
[[ 1.]
 [ 3.]
 [ 2.]
 [ 4.]]
```

For computational basis vectors $\{|i\rangle : i \in \{1, \dots, \dim \mathcal{H}\}\}$, $\langle j|^T = |j\rangle$, such that we can quickly identify that $|\underline{E}_{ij}\rangle\rangle \in \mathcal{H} \otimes \mathcal{H}$. This then gives us the operator-vector correspondence $L(\mathcal{H}) \cong \mathcal{H} \otimes \mathcal{H}$ [65]. The space $\mathcal{H} \otimes \mathcal{H}$ is often called *Liouville space*.

Graphically, we can represent this correspondence using Wood tensor network diagrams [64; 66], in which each index of a tensor is depicted by a *wire* that can be bent to represent transpositions. The Wood diagrams for $|A\rangle\rangle$ in the row- and column-stacking conventions are shown in [Figure 1.2](#).

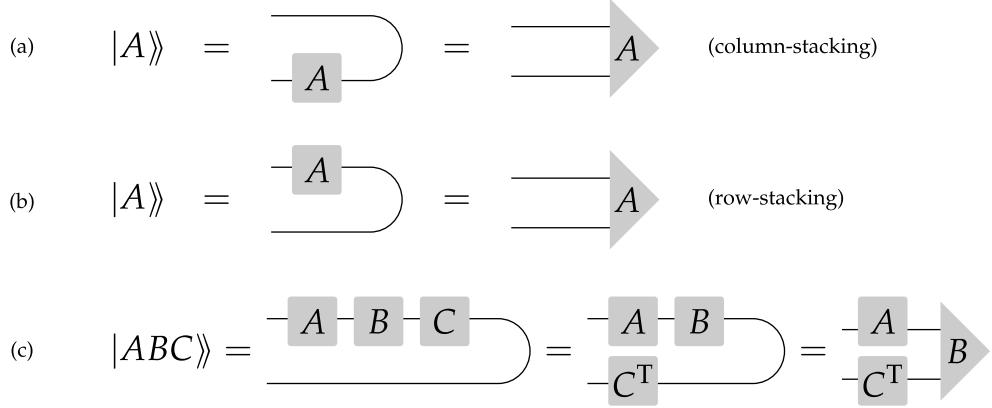


Figure 1.2: Examples of Wood diagrams [64]. Indices of tensors (e.g.: vectors and matrices) are indicated by wires, the tensor product by vertical concatenation and matrix products by horizontal juxtaposition. Bent wires indicate contractions of those indices, boxes indicate tensors and triangles represent vectorized operators. (a) Definition of column-stacking and (b) row-stacking conventions for $|A\rangle\langle$, and (c) a graphical proof of Roth's Lemma (1.34) for row-stacking.

We can use other bases for $L(\mathcal{H})$ as well as the row- and column-stacking conventions. For instance, if $\dim \mathcal{H} = 2^n$ for some n , tensor products of the Pauli matrices are commonly used, forming the Pauli basis

$$\{\sigma_{i_1} \otimes \sigma_{i_2} \otimes \cdots \otimes \sigma_{i_n} / 2^n : i \in \{0, 1, 2, 3\}^n\}. \quad (1.32)$$

More generally, one can use the Heisenberg-Weyl operators, but we focus here on the qubit case. Since each element of the Pauli basis is a Hermitian operator, we can expand any other Hermitian operator as a real vector in the Pauli basis. In particular, since $\mathbb{1}^{\otimes n}$ is the only traceful member of the Pauli basis, density operators can be represented by a real vector r of dimension $\dim \mathcal{H} - 1$, known as a Bloch vector r . For instance, a qubit density operator can be written as

$$\rho = \frac{\mathbb{1}}{2} + \frac{r \cdot (\sigma_x, \sigma_y, \sigma_z)}{2}, \quad (1.33)$$

where the zeroth element is implied by demanding that $\text{Tr}(\rho) = 1$. In the special case of a single qubit, it is straightforward to show that ρ is positive semidefinite if and only if $\|r\|_2 \leq 1$, such that the set of Bloch vectors representing valid states forms a ball of

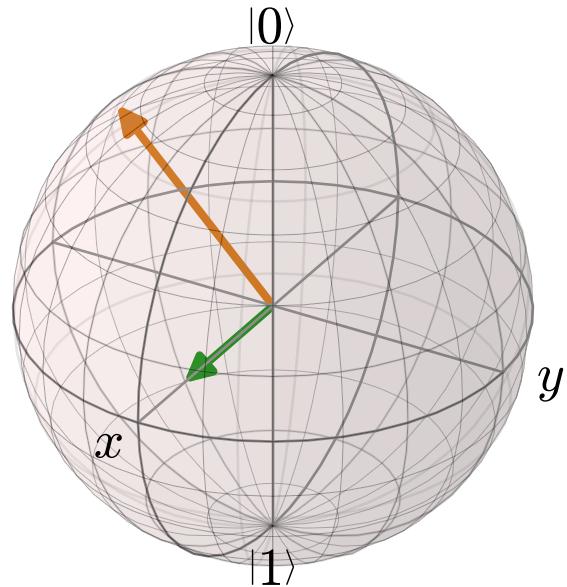


Figure 1.3: States on the Bloch sphere corresponding to vectors $\underline{r} = (0.7, 0, 0)$ (green) and $\underline{r} = (-1, 0, 1)/\sqrt{2}$ (orange).

Listing 1.3: Source code for above figure, using QuTiP.

```

from qutip import *
2 from itertools import starmap
from operator import mul
from numpy import sqrt

sigma = lambda: (sigmax(), sigmay(), sigmaz())
7
def bloch_vector(r):
    r"""
        Returns the state  $\rho_{\underline{r}} = \frac{1}{2}(\mathbb{I} + \underline{r} \cdot \underline{\sigma})$ .
        Here, starmap is used to produce terms of the form  $r_i \sigma_i$ .
    """
12
    return sum(starmap(mul, zip(r, sigma()))), identity(2)) / 2

b = Bloch()
b.add_states(bloch_vector([0.7, 0, 0]))
17 b.add_states(bloch_vector([0, -1 / sqrt(2), 1 / sqrt(2)]))

b.show()

```

radius 1 around the origin, known as the *Bloch sphere* (shown in Figure 1.3). The surface of this sphere corresponds to pure states, such that $\text{Tr}(\rho^2) = 1$.

Once we have adopted a basis for $L(\mathcal{H})$, superoperators, linear operators in $L(L(\mathcal{H}))$, can thus be represented as elements of $L(\mathcal{H} \otimes \mathcal{H})$. It is straightforward to show that in the row-stacking convention,

$$|ABC\rangle\langle = A \otimes C^T |B\rangle\langle. \quad (1.34)$$

This fact, known as Roth's Lemma, allows us to represent group actions by superoperators, such that for a unitary operator U ,

$$|U \bullet \rho\rangle\langle = U \otimes U^{\dagger T} |\rho\rangle\langle = U \otimes U^* |\rho\rangle\langle, \quad (1.35)$$

where U^* is the complex conjugate of U .

We need not consider only those superoperators corresponding to unitary evolution, however. By the same argument as used to introduce density operators, we can also consider convex combinations of unitary evolution,

$$\hat{S} |\rho\rangle\langle = \sum_i p_i \hat{S}_{U_i} |\rho\rangle\langle, \quad (1.36)$$

for $\sum_i p_i = 1$. Such *mixed unitary channels* describe the case in which the system evolves under a unitary U_i with probability p_i .

Superoperators of this form suggest two very useful properties that we will demand of any physical process: they map positive operators to positive operators even when composed with reference states, and they preserve the trace of their inputs. Together, these properties imply that such superoperators map all valid states to valid states. More generally, we can take the set of maps that satisfy these two properties, called completely positive trace-preserving (CPTP) maps, to be the most general class of *linear* operators acting on density operators [67; 65]. More general maps can be considered by including correlations between the system of interest and its environment [68; 66]. That is, the reduced action on a single subsystem need not be a well-defined CPTP map, even if the extended map acting on the system and its environment is CPTP. Here, we shall restrict our focus to those maps that are linear and CPTP.

Note that though we arrived at superoperators by considering mixed unitary channels, there are many CPTP maps which cannot be written in that form. For instance, the *amplitude damping channel*

$$\Lambda[\rho] = |0\rangle\langle 0| \text{Tr}(\rho) \quad (1.37)$$

is a CPTP map that takes all inputs to a single fixed state. In particular, $\Lambda[\mathbb{1}] \neq \mathbb{1}$, such that this is not a mixed unitary channel.

We also note that not all maps acting on quantum states need be ``rectangular.'' That is, a process can add or remove degrees of freedom, for instance, by introducing an additional register. Consider the map $\Lambda_{\text{prep}} : \rho \mapsto \rho \otimes |0\rangle\langle 0|$, which adds a new register prepared in a fiducial input state. This is, as before, a linear function of ρ , such that we can write down the corresponding Liouville-representation superoperator. Assuming $\rho \in L(\mathbb{C}^2)$ is tensored with another qubit,

$$\hat{S}_{\text{prep}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\0 & 0 & 0 & 0 \end{pmatrix}, \quad (1.38)$$

with the corresponding Wood diagram shown in [Figure 1.4](#). In QuTiP, we can quickly implement this preparation by using the `super_tensor()` function to handle the required reordering of tensor indices³.

Listing 1.4: Calculation of state preparation superoperator using QuTiP.

```
>>> import qutip as qt  
>>> q = qt.tensor(qt.identity(2), qt.basis(2))  
>>> s_prep = qt.sprepost(q, q.dag())
```

Similarly, we can write down a superoperator for $\Lambda_{\text{Tr}_1} : \rho \mapsto \text{Tr}_1(\rho)$ for the partial trace over the first of two qubits, noting that $\text{Tr}_1(\rho) = \sum_i \langle i_1 | \rho | i_1 \rangle$. We can quickly find the column-stacking Liouville representation

using the `QuantumUtils`` package for Mathematica [42], with the code snippet below.

³Thanks to Robert Johansson for suggesting the use of `sprepost()` here.

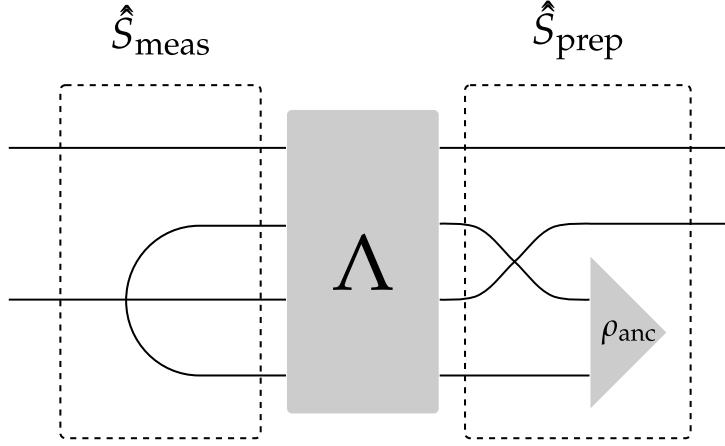


Figure 1.4: Wood diagram for a superoperators \hat{S}_{prep} and \hat{S}_{meas} representing respectively preparation of an ancilla in the state ρ_{anc} , and measurement of that ancilla.

Listing 1.5: Column-stacking partial trace superoperators in Mathematica.

```
In[1]:= Needs["QuantumUtils`"];
In[2]:= PartialTrSuperoperator[{2, 2}, {1}] // First // TexForm
Out[2]//TexForm= \left( \begin{array}{cccccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)
```

1.5.3.2 Representations of Superoperators

Though the Liouville space representation of quantum maps is convenient in a number of different contexts, there are several other useful representations that we will want to consider. The Choi representation, for instance, is a natural choice both for deciding properties of channels [65] and for performing ancilla-assisted process tomography [69]. For a map Λ , the Choi representation $J(\Lambda)$ is defined up to normalization by acting Λ on one half of a maximally entangled state $|1\rangle\langle 1|$,

$$J(\Lambda) := (\Lambda \otimes \mathbb{1}_{L(H)})[|1\rangle\langle 1|], \quad (1.40)$$

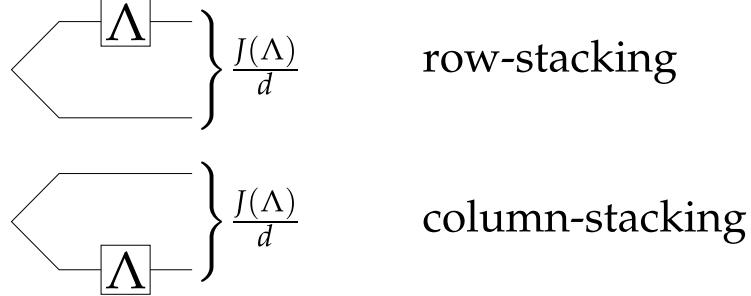


Figure 1.5: Circuits for producing Choi states $J(\Lambda)/d$.

where for clarity, we have denoted by subscripts which space each identity acts upon⁴. The map Λ is then completely positive if and only if $J(\Lambda) \geq 0$, is Hermicity preserving if and only if $J(\Lambda)$ is Hermitian and is trace-preserving if and only if $\text{Tr}_2(J_\Lambda) = \mathbb{1}$ [65], such that it is straightforward to reason about Λ given $J(\Lambda)$. The definition (1.40) also immediately implies an operational description, in which one performs *state* tomography on the output of the circuit in Figure 1.5 in order to reconstruct $J(\Lambda)$.

The Choi representation also nicely points to another useful representation, as can be seen by considering the singular value decomposition of the Choi representation for a map Λ ,

$$J(\Lambda) = \sum_i s_i |\tilde{A}_i\rangle\langle\tilde{B}_i| \quad (1.41)$$

where $\{s_i\}$ are the singular values of $J(\Lambda)$, and where $\{|\tilde{A}_i\rangle\}$ and $\{|\tilde{B}_i\rangle\}$ are the left and right singular vectors, respectively. Let $|A_i\rangle\langle\cdot| = \sqrt{s_i} |\tilde{A}_i\rangle\langle\tilde{B}_i|$ and $|B_i\rangle\langle\cdot|$ be defined similarly. Then, by Roth's Lemma (1.34),

$$J(\Lambda) = \sum_i (A_i \otimes \mathbb{1}_{\mathcal{H}}) |\mathbb{1}_{\mathcal{H}}\rangle\langle\mathbb{1}_{\mathcal{H}}| (B_i \otimes \mathbb{1}_{\mathcal{H}})^\dagger, \quad (1.42)$$

such that the action of Λ alone can now be given in terms of the A_i and B_i operators,

$$\Lambda[\rho] = \sum_i A_i \rho B_i^\dagger. \quad (1.43)$$

This representation is known as the Kraus representation, and is useful for expressing the action of Λ without the use of operators acting on a larger space. This feature is

⁴ Note that this definition depends on one's choice of basis for Liouville space. In the column-stacking convention, $J(\Lambda) := (\mathbb{1}_{L(\mathcal{H})} \otimes \Lambda)[|\mathbb{1}_{\mathcal{H}}\rangle\langle\mathbb{1}_{\mathcal{H}}|]$.

especially important for numerically computing the action of separable channels [38; 1]. Note that for a positive semidefinite operator, the left and right singular vectors are equal, such that if Λ is completely positive, $B_i = A_i$.

A final useful representation is the χ -matrix, which we obtain by expanding the action of Λ in an operator basis such as the Pauli basis $\{\sigma_i\}$ or the Heisenberg-Weyl basis ⁵. In particular, let $\{B_\alpha : \alpha \in \{1, \dim^2 \mathcal{H}\}\}$ be a basis for $L(\mathcal{H})$. Then, we decompose the superoperator \hat{S}_Λ in two copies of this basis to obtain that [67]

$$\hat{S}_\Lambda |\rho\rangle\langle\rho| = \sum_{\alpha, \beta} \chi_{\alpha, \beta} \bar{B}_\beta \otimes B_\alpha |\rho\rangle\langle\rho| \quad (1.44a)$$

$$= \sum_{\alpha, \beta} \left| \chi_{\alpha, \beta} B_\alpha \rho B_\beta^\dagger \right\rangle\langle\rho|, \quad (1.44b)$$

such that

$$\Lambda[\rho] = \sum_{\alpha, \beta} \chi_{\alpha, \beta} B_\alpha \rho B_\beta^\dagger. \quad (1.44c)$$

This then defines the χ -matrix,

$$\underline{\underline{\chi}} := \sum_{\alpha, \beta} |\alpha\rangle\langle\beta| \chi_{\alpha, \beta}. \quad (1.45)$$

The χ matrix can be estimated from tomographic data using standard linear algebra methods [70].

Because the χ -matrix is a change of basis from the Choi matrix [64], $\underline{\underline{\chi}} \geq 0$ if and only if Λ is completely positive, and $\text{Tr}(\underline{\underline{\chi}}) = \dim \mathcal{H}$ if and only if Λ is trace-preserving. The χ -matrix formalism is particularly nice, however, for considering the special class of mixed-unitary channels wherein each unitary is a Pauli operator. Such channels are called *Pauli channels*, and are diagonal in the $\underline{\underline{\chi}}$ -matrix representation,

$$\Lambda[\rho] = \sum_{\alpha} p_\alpha \sigma_\alpha \bullet \rho = \sum_{\alpha, \beta} p_\alpha \delta_{\alpha, \beta} \sigma_\alpha \rho \sigma_\beta^\dagger. \quad (1.46)$$

That is, we immediately identify $\chi_{\alpha, \beta} = \dim \mathcal{H} \cdot p_\alpha \delta_{\alpha, \beta}$ in this important special case. One Pauli channel we will frequently consider in this work is the *completely depolarizing*

⁵In practice, one has to renormalize the Pauli or Heisenberg-Weyl operators in order to obtain an orthonormal basis, since $\text{Tr}(\sigma_i^\dagger \sigma_i) = \text{Tr}(1) = \dim \mathcal{H}$.

channel

$$\Omega[\rho] := \frac{\mathbb{1}}{\dim \mathcal{H}} = \frac{1}{\dim^2 \mathcal{H}} \sum_{P \in \mathcal{P}_n} P \bullet \rho. \quad (1.47)$$

The χ -matrix for Ω is simply the identity, up to a normalization, representing that Ω acts each Pauli operator on a state with equal probability. Having defined the completely depolarizing channel, we can now define a broader class of depolarizing channels by mixing the identity channel with a probability p of acting Ω , thus defining the *depolarizing strength*.

The Liouville, Choi, χ -matrix and Kraus representations are all straightforward to compute using libraries such as QuTiP.

Listing 1.6: Computation of different superoperator representations.

```
>>> from qutip import *
>>> S = rand_super(2)
>>> print S
4 Quantum object: dims = [[[2], [2]], [[2], [2]]], shape = [4, 4],
   type = super, isherm = False
Qobj data =
[[ 0.57867425 +0.0000000e+00j -0.00564534 +7.74610990e-02j
   -0.00564534 -7.74610990e-02j  0.42132575 +0.0000000e+00j]
 9 [ 0.00564534 +7.74610239e-02j  0.27849449 +4.23704191e-19j
   0.02802420 -4.10660463e-03j -0.00564534 -7.74610239e-02j]
[ 0.00564534 -7.74610239e-02j  0.02802420 +4.10660463e-03j
   0.27849449 -1.50538581e-19j -0.00564534 +7.74610239e-02j]
[ 0.42132575 +0.0000000e+00j  0.00564534 -7.74610990e-02j
 14 0.00564534 +7.74610990e-02j  0.57867425 +0.0000000e+00j]]
>>> print S.iscptp
True
>>> print to_choi(S)
Quantum object: dims = [[[2], [2]], [[2], [2]]], shape = [4, 4],
 19 type = super, isherm = True, superrep = choi
Qobj data =
[[ 0.57867425 +0.0000000e+00j  0.00564534 -7.74610239e-02j
   -0.00564534 -7.74610990e-02j  0.27849449 -1.50538581e-19j]
 24 [ 0.00564534 +7.74610239e-02j  0.42132575 +0.0000000e+00j
   0.02802420 -4.10660463e-03j  0.00564534 +7.74610990e-02j]
[-0.00564534 +7.74610990e-02j  0.02802420 +4.10660463e-03j
   0.42132575 +0.0000000e+00j -0.00564534 +7.74610239e-02j]
[ 0.27849449 +4.23704191e-19j  0.00564534 -7.74610990e-02j
 29 -0.00564534 -7.74610239e-02j  0.57867425 +0.0000000e+00j]]
>>> print to_chi(S)
Quantum object: dims = [[[2], [2]], [[2], [2]]], shape = [4, 4],
```

```

    type = super, isherm = True, superrep = chi
Qobj data =
[[ 1.71433748e+00 +0.00000000e+00j  1.66212807e-01 -1.66212807e-01j
 34   -1.43631438e-01 -1.43631438e-01j  0.00000000e+00 +0.00000000e+00j]
 [ 1.66212807e-01 +1.66212807e-01j  8.50864707e-01 +0.00000000e+00j
 0.00000000e+00 -5.60483947e-02j  6.96647132e-08 -6.96647139e-08j]
 [-1.43631438e-01 +1.43631438e-01j  0.00000000e+00 +5.60483947e-02j
 8.34438288e-01 +0.00000000e+00j -8.06172221e-08 -8.06172226e-08j]
 39  [ 0.00000000e+00 +0.00000000e+00j  6.96647132e-08 +6.96647139e-08j
 -8.06172221e-08 +8.06172226e-08j  6.00359521e-01 +0.00000000e+00j]]
>>> Ks = to_kraus(S)
>>> for K in Ks:
...     print K
44 ...
Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = False
Qobj data =
[[ 0.63894814 +5.68855226e-17j -0.01568380 +2.15201149e-01j]
 [ 0.01568380 +2.15201149e-01j  0.63894814 -1.08834735e-17j]]
49 Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True
Qobj data =
[[ 3.87414352e-01 -1.99652267e-17j  2.28563063e-08 -3.13616727e-07j]
 [ 2.28563064e-08 +3.13616728e-07j -3.87414352e-01 -4.20678116e-17j]]
Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = False
54 Qobj data =
[[-0.01036386 +1.42204905e-01j  0.41775541 +6.12169650e-02j]
 [ 0.42221689 -1.13419246e-17j -0.01036386 +1.42204905e-01j]]
Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = False
Qobj data =
[[ -2.61524781e-08 -3.58844275e-07j  4.43284481e-01 -3.13067175e-17j]
 [ -4.38600388e-01 +6.42715423e-02j  2.61524808e-08 +3.58844308e-07j]]

```

1.5.3.3 Liouvillians and Lindbladians

With the vectorspace structure of $L(\mathcal{H})$ in mind, we revisit the von Neumann equation (1.14), noting that it expresses the derivative d/dt as a superoperator \hat{L} ,

$$\frac{d}{dt} |\rho\rangle\langle\rho| = | -i[H, \rho] \rangle\langle\rho| = -i(|H\rho\rangle\langle\rho| - |\rho H\rangle\langle\rho|) = -i(H \otimes \mathbb{1} - \mathbb{1} \otimes H^T) |\rho\rangle\langle\rho| =: -i\hat{L} |\rho\rangle\langle\rho|. \quad (1.48)$$

This superoperator is often called the *Liouvillian* of the system, and generates dynamics for $|\rho\rangle\langle\rho|$ in the same manner that a Hamiltonian H generates dynamics for $|\psi\rangle\langle\psi|$,

$$|\rho(t)\rangle\langle\rho(t)| = e^{-i\hat{L}t} |\rho(0)\rangle\langle\rho(0)|. \quad (1.49)$$

Unlike Hamiltonians, however, Liouvillians can incorporate dissipative, or open quantum system, dynamics. In particular, we can represent open dynamics by

$$|\rho(t)\rangle\rangle = e^{\hat{G}t} |\rho(0)\rangle\rangle, \quad (1.50)$$

where $\hat{G} := -i\hat{L} + \hat{D}$, and where \hat{D} is a superoperator represents the non-unitary portion of the evolution, and is called a *dissipator*.

Here, we derive a simple example of a dissipator in Lindblad form [71] for general amplitude damping, in which the environment interacts with a system such that a qubit state ρ decays to an equilibrium state $\rho_{\text{eq}} := p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$. We shall assume that this decay happens with a characteristic time T_1 such that over a time interval Δt , the environment acts by a map with Kraus decomposition [67]

$$A_0 = \sqrt{p} \left(\mathbb{1} - \frac{\delta t}{2T_1} E_- + O(\Delta t^2) \right) \quad (1.51a)$$

$$A_1 = \sqrt{p} \left(\sqrt{\frac{\Delta t}{T_1}} \sigma_+ \right) \quad (1.51b)$$

$$A_2 = \sqrt{1-p} \left(\mathbb{1} - \frac{\delta t}{2T_1} E_+ + O(\Delta t^2) \right) \quad (1.51c)$$

$$A_3 = \sqrt{1-p} \left(\sqrt{\frac{\Delta t}{T_1}} \sigma_- \right), \quad (1.51d)$$

where $E_+ = |0\rangle\langle 0|$ and $E_- = |1\rangle\langle 1|$ are the elementary matrices along the diagonal, and where $\sigma_{\pm} = (\sigma_x \pm i\sigma_y)/2$. Taking the limit as $\Delta t \rightarrow 0$, we can write this map as a derivative operator,

$$\begin{aligned} \partial_t \rho(t) &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (\rho(t + \delta t) - \rho(t)) \\ &= \frac{p}{T_1} \left[\sigma_+ \rho \sigma_- - \frac{1}{2} \{E_-, \rho\} \right] + \frac{1-p}{T_1} \left[\sigma_- \rho \sigma_+ - \frac{1}{2} \{E_+, \rho\} \right], \end{aligned} \quad (1.52)$$

where $\{A, B\} := AB + BA$ is the anti-commutator.

Finally, vectorizing $|\partial_t \rho\rangle\rangle$ gives the dissipator superoperator \hat{D} such that $\hat{D} |\rho\rangle\rangle =$

$|\partial_t \rho\rangle\langle\rho|$. In the column-stacking convention,

$$\begin{aligned}\hat{D} = & \frac{p}{T_1} \left(\sigma_-^T \otimes \sigma_+ - \frac{1}{2} (\mathbb{1} \otimes E_- + E_-^T \otimes \mathbb{1}) \right) \\ & + \frac{1-p}{T_1} \left(\sigma_+^T \otimes \sigma_- - \frac{1}{2} (\mathbb{1} \otimes E_+ + E_+^T \otimes \mathbb{1}) \right).\end{aligned}\tag{1.53}$$

Thus, the evolution of ρ under this process alone has the solution $|\rho(t)\rangle\langle\rho| = e^{\hat{D}t} |\rho(0)\rangle\langle\rho|$. This form is *generic* for single-parameter families of superoperators $\hat{S}(t)$ such that $\hat{S}(t+s) = \hat{S}(t)\hat{S}(s)$ [72], known as quantum dynamical semigroups. Below, we consider the cumulant expansion, which incorporates both dynamical generators for dissipative processes and stochastic Hamiltonian evolution to find effective superoperators.

1.5.3.4 Magnus, Floquet and Cumulant Expansions

Often, we will want to consider not just time-independent Hamiltonians and Liouvillians, but also the effects on a quantum system due to time-dependence such as is introduced by stochastic processes or by control pulses. We can do this by using the Magnus expansion [73; 74; 75] to find an *effective* generator H_{eff} or \hat{L}_{eff} . In particular, consider $H = H(t)$, such that the unitary evolution is generated by a time-ordered exponential

$$U(T) = \text{T exp} \left(-i \int_0^T H(t) dt \right).\tag{1.54}$$

We can then find an operator H_{eff} that generates $U(T)$ by an ordinary matrix exponential

$$U(T) = \exp(-iTH_{\text{eff}}).\tag{1.55}$$

In particular, the Magnus expansion gives that if $\int_0^T \|H(t)\|_2 dt < \pi$ [76], then

$$H_{\text{eff}} = H^{(0)} + H^{(1)} + \dots\tag{1.56a}$$

$$H^{(0)} = \frac{1}{t} \int_0^T H(t) dt\tag{1.56b}$$

$$H^{(1)} = \frac{1}{2t} \int_0^T \int_0^{t_1} [H(t_1), H(t_2)] dt_2 dt_1\tag{1.56c}$$

$$H^{(2)} = \frac{1}{6t} \int_0^T \int_0^{t_1} \int_0^{t_2} ([H(t_1), [H(t_2), H(t_3)]] + [H(t_3), [H(t_2), H(t_1)]]) dt_3 dt_2 dt_1.\tag{1.56d}$$

\vdots

This expansion is especially useful in considering the effective Hamiltonian at stroboscopic intervals of a rotating or toggling interaction frame, owing to Floquet's theorem [77], which states that if $H(t) = H(t + T)$ for some period T , then

$$U(t + nT) = U(t)U(T)^n \quad (1.57)$$

for $n \in \mathbb{N}$. Thus, at stroboscopic intervals of T , the evolution is given by the effective Hamiltonian obtained from taking the Magnus expansion over a single period. In this limit, we can use the formalism of Floquet space [78] to explicitly find the stroboscopic Hamiltonian in terms of a Fourier decomposition of $H(t)$,

$$H(t) = \sum_j e^{ij\omega t} H_j \quad (1.58)$$

for a set of linear operators $\{H_j\}$ such that $H_j = H_{-j}^\dagger$. Leskes et al [79] then give that using this decomposition, the stroboscopic Hamiltonian H_{eff} is given by

$$H_{\text{eff}} = H_{\text{eff}}^{(0)} + H_{\text{eff}}^{(1)} + H_{\text{eff}}^{(2)} + \dots \quad (1.59a)$$

$$H_{\text{eff}}^{(0)} = H_0 \quad (1.59b)$$

$$H_{\text{eff}}^{(1)} = -\frac{1}{2} \sum_{n \neq 0} \frac{[H_{-n}, H_n]}{n\omega} \quad (1.59c)$$

$$H_{\text{eff}}^{(2)} = -\frac{1}{3} \sum_{\substack{n \neq 0 \\ n' \neq 0 \\ n \neq n'}} \frac{[[H_{n-n'}, H_{n'}], H_{-n}]}{nn'\omega^2} - \frac{1}{2} \sum_{n \neq 0} \frac{[[H_0, H_n], H_{-n}]}{n^2\omega^2}. \quad (1.59d)$$

⋮

Since in many experiments, the Fourier representation of $H(t)$ has a small number of modes for convenient choices of rotating frames, this gives us a very convenient expression for finding effective Hamiltonians in terms of a small number of finite sums.

By the same arguments, we can readily find effective Liouvillian operators \hat{L}_{eff} under the same stroboscopic conditions. We are often interested, however, in the ensemble average action

$$\hat{S}(t) = \left\langle T \exp \left(\int_0^t (\hat{G}(t)) dt \right) \right\rangle, \quad (1.60)$$

where $\hat{G}(t) = i\hat{L}(t) + \hat{D}(t)$ is drawn from a stochastic process. Deterministic evolution can be considered as well by letting $\hat{G}(t)$ be defined in the interaction frame of the deterministic unitary and dissipative dynamics. The cumulant expansion [80], which has

previously been used to analyze decoherence in quantum systems [81], then expands the Magnus expansion to include this case, giving that

$$\hat{S} = \exp(\hat{K}(t)) \quad (1.61a)$$

$$\hat{K} = \sum_{n=1}^{\infty} \frac{(-it)^n}{n!} \hat{K}_n = -it\hat{K}_1 - \frac{t^2}{2}\hat{K}_2 \quad (1.61b)$$

$$\hat{K}_1 = \frac{1}{t} \int_0^{t_1} \left\langle \hat{G}(t_1) \right\rangle dt_1 \quad (1.61c)$$

$$\hat{K}_2 = \frac{1}{t^2} T \int_0^{t_1} \int_0^{t_2} \left\langle \hat{G}(t_1) \hat{G}(t_2) \right\rangle dt_2 dt_1 - \hat{K}_1^2. \quad (1.61d)$$

Numerical integration of this expansion can then be used to evaluate the performance of a gate, for instance, given reasonable models of the physics of a system [38].

1.5.4 Error Correction and Stabilizer Codes

We close the introduction by introducing one more broad area of quantum information, namely the theory of *quantum error correction*. Naively, applying an error correction code to a quantum state may seem impossible on the face of it, since we cannot simply repeat a quantum state $|\psi\rangle$ to make an encoded state $|\bar{\psi}\rangle = |\psi\rangle|\psi\rangle\cdots|\psi\rangle$ without violating the no-cloning theorem. Instead, we choose an encoding of a state such that the location and kind of errors that occur can be revealed by measurements that do not at all depend on *which* state we have encoded.

This is formalized by the Knill-Laflamme conditions [82]. In particular, let $|\bar{i}\rangle$ be a *codeword* representing a computational basis state $|i\rangle$. Then, a set of error operators $\{A_a\}$ is recoverable if and only if for all basis states $|i\rangle$ and $|j\rangle$ and for all errors A_a and A_b ,

$$\langle \bar{i} | A_a^\dagger A_b | \bar{j} \rangle = \delta_{ij} c_{a,b}, \quad (1.62)$$

where $c_{a,b}$ does not depend on i and j . If these conditions are met, then we immediately have that for any states $\bar{\psi}$ and $\bar{\phi}$ in the *code space* $\bar{\mathcal{H}} := \text{span}\{|\bar{i}\rangle\}$,

$$\langle \bar{\phi} | A_a^\dagger A_b | \bar{\psi} \rangle = \langle \phi | \psi \rangle c_{a,b}, \quad (1.63)$$

such that action of the error operators $\{A_a\}$ preserves the structure of the codespace.

The Knill-Laflamme conditions also immediately imply by linearity that if $\{A_a\}$ is a set of recoverable errors for a given set of codewords, then so is $\text{span}\{A_a\}$. Since

$L(\mathbb{C}^2) = \text{span}\{\mathbb{1}, X, Y, Z\}$, if we can correct any single-qubit Pauli operator, we can therefore correct any single-qubit operator at all. This insight then motivates using the unique structure of the Pauli group to construct and reason about quantum error correction codes.

For instance, consider a bit flip code

$$|\bar{0}\rangle := |000\rangle \quad (1.64a)$$

$$|\bar{1}\rangle := |111\rangle = XXX |\bar{0}\rangle. \quad (1.64b)$$

This code then satisfies (1.62) for the error set $A_0 = \mathbb{1}$, $A_a = X_a$ for $a \in \{1, 2, 3\}$, as can be verified by noting that $A_a^\dagger A_b = \mathbb{1}$ if and only if $a = b$, such that $A_a^\dagger A_b$ maps the codewords $|\bar{i}\rangle$ into orthogonal copies of the codespace for each a and b . By measuring which such codespace a state is in, the error can thus be corrected without measuring the state itself.

To derive the measurement we should use for this recovery, it is helpful to rely on the stabilizer formalism [62], which makes precise the way in which the structure of the Pauli group relates to the Knill-Laflamme conditions. In the example of the bit flip code, we note that $S = \langle ZZ\mathbb{1}, \mathbb{1}ZZ \rangle = \text{stab}\{|\bar{0}\rangle, |\bar{1}\rangle\}$ such that we can reason instead about the group S . Then, we note that X_a for $a \in \{1, 2, 3\}$ anticommutes with at least one element of S . As a consequence, for all $|\bar{\psi}\rangle$ and $|\bar{\phi}\rangle$ stabilized by S , there exists an S_i for each X_a such that

$$\langle \bar{\phi}|X_a|\bar{\psi}\rangle = \langle \bar{\phi}|X_a S_i|\bar{\psi}\rangle = -\langle \bar{\phi}|S_i X_a|\bar{\psi}\rangle = -\langle \bar{\phi}|X_a|\bar{\psi}\rangle. \quad (1.65)$$

From this, we conclude that $\langle \bar{\phi}|X_a|\bar{\psi}\rangle = 0$. Similarly, if $a \neq b$, $\langle \bar{\phi}|X_a^\dagger X_b|\bar{\psi}\rangle = 0$ since the set of stabilizer group generators with which each X_a anticommutes is distinct. That is, by interpreting the group generators as a sequence of measurements, we obtain a *syndrome* that uniquely distinguishes an error from our correctable set. A code with this property is said to be *non-degenerate*. The last thing we must check to verify that (1.62) holds is that $\langle \bar{\phi}|X_a^\dagger X_a|\bar{\psi}\rangle = \langle \phi|\psi\rangle$; this, however, is immediate from the fact that each X_a in our recoverable set is unitary.

Generalizing, the set of states stabilized by an Abelian subgroup S of the Pauli group is a *stabilizer code* with recoverable operators given by those operators which are either in S or anticommute with at least one element of S . Those elements which commute with all elements of S but that do not necessarily stabilize the codespace are called the *normalizer* $N(S)$ of S , such that the uncorrectable errors are $N(S) \setminus S$. For the bit flip code above, our notation for (1.64) already suggests the normalizer for that code must include $\langle XXX\rangle$, since XXX maps $|\bar{0}\rangle$ to $|\bar{1}\rangle$, but commutes with both generators of S . We

complete the normalizer by noting that any single-qubit Z error is unrecoverable, such that $N(S)/S = \langle XXX, Z\mathbb{1}\mathbb{1} \rangle$, where we have used a factor group to indicate that any product of a stabilizer element and an unrecoverable error is also unrecoverable.

Note that $N(S)/S$ has the same structure as the Pauli group on a single qubit, as $\bar{X} = XXX$ and $\bar{Z} = Z\mathbb{1}\mathbb{1}$ have the same multiplication relations as X and Z , up to multiplication by an element of S . Moreover, the elements of $N(S)/S$ act as we would expect Pauli operators to act on the codespace,

$$\bar{X}|\bar{0}\rangle = |\bar{1}\rangle, \quad \bar{X}|\bar{1}\rangle = |\bar{0}\rangle \quad (1.66a)$$

$$\bar{Z}|\bar{0}\rangle = |\bar{0}\rangle, \quad \bar{Z}|\bar{1}\rangle = -|\bar{1}\rangle. \quad (1.66b)$$

We thus refer to $N(S)/S$ as the *logical* Pauli group for the codespace.

In particular, we will write that a code S is an $\llbracket n, k, d \rrbracket$ stabilizer code, where n is the number of physical qubits in which states are encoded, k is the number of logical qubits and where d is the minimum weight of an element of $N(S)/S$, called the *distance*. Since the bit flip code cannot correct phase flips, it has a distance of 1, such that the bit flip code is a $\llbracket 3, 1, 1 \rrbracket$ code.

A code that can correct a single arbitrary error must have distance at least 3, such that two different single-qubit errors are distinguishable. The smallest code with this property is the $\llbracket 5, 1, 3 \rrbracket$ perfect code [83],

$$S = \left\langle \begin{array}{c} XZZX\mathbb{1} \\ \mathbb{1}XZZX \\ X\mathbb{1}XZZ \\ ZX\mathbb{1}XZ \end{array} \right\rangle. \quad (1.67)$$

Here, $N(S)/S = \langle XXXXX, ZZZZZ \rangle$. Using QuaEC [45], we can readily verify that the perfect code has a distance of 3 by explicit enumeration over all elements of the normalizer group excluding those elements in perfect code S itself.

Listing 1.7: Explicit enumeration of $N(S)/S$ for the five-qubit perfect code.

```
>>> import qecc as q
>>> stab = q.StabilizerCode.perfect_5q_code()
>>> print stab
5-qubit perfect code
5
S = <i^0 XZZX, i^0 IXZZX, i^0 XIXZZ, i^0 ZXIXZ>
Xbars = PauliList(i^0 XXXXX)
Zbars = PauliList(i^0 ZZZZZ)
>>> S = list(stab.stabilizer_group())
>>> print min([
```

```
10 ...     P.wt for P in stab.normalizer_group()  
...     if P.mul_phase(-P.ph) not in S  
... ])  
3
```

2 Characterization with Classical Resources

In the course of developing useful quantum devices, we encounter a variety of different *learning* problems, in which we characterize some aspect of a quantum system. For instance, to design control using simulation-based optimal control theory [84] ([Chapter 5](#)) with high fidelity, we require accurate knowledge of the control dynamics for our quantum system. As an alternative, we can use closed-loop algorithms to modify control designs based on a characterization of the performance of an initial design [85; 86; 25] ([Section 6.1.3](#)).

As we will explore in further detail in [Chapter 3](#) and [Section 6.2](#), certification of quantum dynamics can be expressed as a characterization problem, allowing us to determine whether the predictions made by a quantum simulator can be trusted. This problem is especially timely since quantum simulation experiments are approaching a complexity where classical computers are unable to simulate their evolution [87; 88; 89]. Traditional solutions to this problem, such as tomographic methods [90; 91; 92; 93; 94; 95; 60], work well in small systems but are often impractical for learning parameters for large quantum systems, as well as for learning parameters such as decoherence times (T_2). These problems motivate the exploration of new Hamiltonian learning algorithms that can push past the limits of tomographic methods.

We approach this challenge and render the Hamiltonian learning process tractable by utilizing information about a system, rather than starting from worst-case assumptions

such as those made in traditional quantum process and state tomography. In practice, as we will discuss in [Section 2.1.1](#), we often have knowledge about the dynamics that describe the evolution of a system of interest, and wish to refine that knowledge by estimating specific parameters of a *model*. Thus, practical Hamiltonian learning can often be achieved via a suitable parameterization of the Hamiltonian, $H(x_1, \dots, x_d)$, reducing the problem to estimating the vector of parameters $\underline{x} = (x_1, \dots, x_d)$.

We also seek to characterize our uncertainty about estimation errors by providing a *region estimate* for the Hamiltonian parameters that encloses some fixed volume of parameter space in which the mean or the variance of the Hamiltonian parameters are expected to be found with high-probability. We generalize this concept to incorporate learning *hyperparameters*, which describe the distribution of the Hamiltonian parameters in cases where the parameters randomly drift between experiments, thus relaxing the assumption that an experiment is described by a single static Hamiltonian.

Here, we present a framework for learning Hamiltonian parameters, based on Bayes' rule. We demonstrate Bayesian analysis in an analytically tractable special case and discuss the *sequential Monte Carlo* (SMC) algorithm for more general cases. We then provide several examples of SMC in practice and discuss its robustness to common experimental concerns. In [Section 6.2](#), we show how this algorithm can be extended to employ quantum resources, and thus to verify the correct operation of large quantum simulators.

2.1 Bayesian Inference for Quantum Applications

To learn Hamiltonian dynamics, we adopt a methodology based on Bayes' rule, which provides a means of calculating the probability $\text{Pr}(\text{hypothesis}|\text{data})$ of a hypothesis being true, given experimental observations. In particular,

$$\text{Pr}(\text{hypothesis}|\text{data}) = \frac{\text{Pr}(\text{data}|\text{hypothesis})}{\text{Pr}(\text{data})} \text{Pr}(\text{hypothesis}), \quad (2.1)$$

where $\text{Pr}(\text{data}|\text{hypothesis})$ is the [likelihood function](#) for an experiment. The likelihood is a complete description of an experiment, as it specifies the probability for any measurement outcome (data) to occur, given a particular hypothesis. In the context of quantum mechanics, Born's rule [\(1.12\)](#) acts as a likelihood function if we interpret the preparation $|\psi\rangle$ as a hypothesis about the state of a system.

The posterior probability distribution $\text{Pr}(\text{hypothesis}|\text{data})$ represents what we know about a set of hypotheses from having performed measurements on a system. For instance, the mean over this distribution provides an [estimator](#) for which hypothesis is

most parsimonious with the data, as we will discuss in [Section 2.1.2](#). One advantage of employing a Bayesian approach is that the posterior distribution also characterizes the uncertainty in the estimates one obtains. This can be used, for instance, to derive region estimates that include the true model with high probability, as we will explore in [Section 3.1](#).

From equation [\(2.1\)](#), we see that simulation according to the likelihood function is intimately connected to our ability to draw samples from the posterior, and hence provides a means of learning. For those quantum systems we can efficiently simulate classically, this provides us with a very valuable resource, as we can sample from the posterior by simulating according to a hypothesis about our system. On the other hand, for large quantum systems, Bayesian methods are not as obviously applicable--- indeed, the conjectured intractability of simulating quantum dynamics would naively seem to preclude using simulation-based approaches to learning. In [Chapter 6](#), we will show that this apparent drawback becomes an advantage when we also include quantum simulation as a resource. For now, however, we will limit our examples to those that admit efficient classical simulations, including small systems and examples in which our experimental protocol results in a simpler effective description.

Bayesian methods have been widely used in quantum information to discriminate [\[96\]](#) or estimate states [\[51; 59\]](#), to incorporate models of noisy measurements [\[97\]](#), to characterize drifting frequencies [\[98\]](#), and to estimate Hamiltonians [\[99; 100; 101; 102; 103; 104; 32\]](#).

2.1.1 Prior Information

An important advantage to Bayesian methods is the natural incorporation of prior information. In particular, the [prior](#) distribution $\text{Pr}(\text{hypothesis})$ as in [\(2.1\)](#) represents what is known before any data is collected.

It is important to note, however, that if the prior is not chosen to be an accurate representation of the experimenter's initial knowledge, then this advantage can become a disadvantage, in that the prior may then support misleading or incorrect hypotheses. Though in some important examples, a bad prior may be overcome (such as in [Section 2.3.2](#)), this is at the cost of additional data, and can thus deny a Bayesian experimentalist the computational and data collection advantages that they might otherwise enjoy. In practice, this concern can be largely mitigated by choosing conservative priors, and by using model selection to ensure that a reasonable conclusion is being reached ([Chapter 3](#)). Moreover, that we work with models at all implies that this concern is not

unique to Bayesian methods [105]. In particular, our choices of models and parameterizations are informed by prior information, such that we preclude observations that are not supported by any assignment of model parameters.

2.1.1.1 Parameter Reduction

In many cases, physical or operational arguments allow for prior samples to be expressed using a smaller number of parameters. For instance, by assuming that the dynamical generator of a system admits only completely-positive and trace-preserving (CPTP) dynamics, Boulant *et al* [61] dramatically reduce the parameter count that they require in order to estimate dynamical generators. Their approach still requires exponentially many parameters, however, such that it is desirable to incorporate more information.

For instance, if a system is known to admit only Ising interactions, then it is not necessary to consider all Hermitian operators, as one can instead write down a list of couplings,

$$H(\underline{x}) = \sum_{\langle i,j \rangle} x_{ij} \sigma_z^{(i)} \sigma_z^{(j)}. \quad (2.2)$$

Thus, inference can be carried out on the **model parameters** \underline{x} instead of in a space that grows exponentially in the number of qubits. Parameter reduction can be achieved in a more general context by, for instance, assuming that the Hamiltonian of a system is k -local for some k , or has a subgraph of the complete graph for its interactions. We shall see later, in [Chapter 6](#), that we can also treat a reduced parameter model as an *approximation* to a higher-dimensional model, and then learn the parameters included by the approximate model. By reducing parameters in any of these ways, we remove a critical impediment to the application of inference in large quantum systems: even if we could somehow learn a parameter with constant effort, if there are exponentially many parameters that we must learn, then there is little that we can do.

In the context of learning quantum states, there is something deeper that can be said: explaining the data that we can obtain in the lab is a much less demanding problem than learning a complete description of a state, as the effective dimension for probably approximately correct (PAC) learning is substantially less than exponential in the number of qubits [48]. Put differently, as long as we do not take exponentially many measurements, our data is supported by a subexponential description. Similar arguments have been advanced for continuous-time dynamics, showing that those states reachable by physically-realizable Hamiltonians explore only a very small submanifold of Hilbert space [106].

2.1.2 Decision Theory

Before showing how to apply Bayesian inference in quantum information, it is helpful to first take a brief diversion into how we assess the quality of a learning algorithm, and the ultimate bounds on how well any algorithm can perform.

2.1.2.1 Loss and Risk

In order to assess the performance of a learning procedure, we define a **loss function** between an **estimate** and the true value of a model [107]. Throughout this work, we will focus on the quadratic loss, which generalizes the mean-squared error for multiple parameters.

Definition 5 (Quadratic Loss) *Given a true model \underline{x} and an estimate $\hat{\underline{x}}(D)$, the quadratic loss of that estimate is*

$$L_{\underline{Q}}(\underline{x}, \hat{\underline{x}}(D)) := (\underline{x} - \hat{\underline{x}}(D))^T \underline{Q} (\underline{x} - \hat{\underline{x}}(D)),$$

where \underline{Q} is a positive semidefinite matrix that defines the relative scale between the different parameters. Unless otherwise noted, we shall take $\underline{Q} = \underline{\mathbb{1}}$.

One can also consider other loss functions, depending on the application. For example, the Kullback-Leibler divergence is a commonly-used loss function with a well-motivated operational interpretation [108]. Within quantum information, the fidelity between two states may be more directly useful instead. We focus on the quadratic loss in this work, as it is generally applicable to a wide range of scenarios, can be easily calculated and admits useful bounds, as we shall see in [Section 2.1.2.2](#).

Once we have defined a loss function that is appropriate for our experiment, then we can reason about the average performance of an **estimator**.

Definition 6 (Risk) *Given a true model \underline{x} , an estimator $\hat{\underline{x}}(\cdot)$, and a loss function L , the **risk** is given by*

$$R(\underline{x}, \hat{\underline{x}}(\cdot); L) := \mathbb{E}_D[L(\underline{x}, \hat{\underline{x}}(D))].$$

For some important cases, such as that in [Section 2.1.3](#), we are able to analytically find approximations to the risk incurred by a particular estimator, such that heuristics can be designed that minimize the risk. In many cases of interest, however, this is not feasible.

Instead, in order to determine the risk incurred by an estimator, one approach that we can take is to repeatedly pick ``true'' models, generate simulated data from them, and compare to the loss incurred by the estimator given that simulated data. That is, we can perform a Monte Carlo sampling of the expectation value in [Definition 6](#).

[Definition 6](#) still leaves the assumption, however, that we have the true model to compare against. While this is the case when we are benchmarking a learning algorithm by repeatedly drawing data from a simulator with a true model that is hidden from the inference algorithm, we are often interested in how well we expect to learn without assuming a particular hypothesis. The [Bayes risk](#), then, is the expectation value of the [risk](#), taken over some prior.

Definition 7 (Bayes Risk) Given a [prior](#) distribution $\pi(\underline{x})$, an [estimator](#) $\hat{\underline{x}}(\cdot)$, and a [loss function](#) L , the [Bayes risk](#) is given by

$$r(\pi, \hat{\underline{x}}(\cdot); L) := \mathbb{E}_{\underline{x} \sim \pi}[R(\underline{x}, \hat{\underline{x}}; L)].$$

This definition then tells us what losses we should expect to incur *a priori*, given a particular estimation strategy and given our initial state of knowledge about \underline{x} . We are thus interested in [estimators](#) and experiment design strategies that minimize this quantity, as they provide techniques that give the smallest loss, averaged over the hypotheses that we are trying to learn.

For many commonly-used loss functions, the best estimator according to the Bayes risk is given by an expectation value over the posterior [[109](#); [110](#)],

$$\hat{\underline{x}}(D) = \mathbb{E}[\underline{x}|D]. \quad (2.3)$$

2.1.2.2 Fisher Information and Cramér-Rao Bound

Having thus defined the [loss function](#) that we will use in evaluating our estimates, and hence the [risk](#) on [estimators](#), we then wish to determine how small we can possibly make the risk and the Bayes risk for a given set of experimental designs. In the extreme case that the data set doesn't depend on the [model parameters](#) at all, we clearly cannot do better than to simply guess from the prior, illustrating that to at least some degree, the answer to this question depends on how strongly the likelihood is a function of the parameters of interest. This is made formal by considering the Fisher score q of a likelihood function,

$$q := \nabla_{\underline{x}} \ln \Pr(D|\underline{x}). \quad (2.4)$$

The Cramér-Rao bound [108] lower-bounds the covariance of any unbiased estimator of \underline{x} in terms of the Fisher information $\underline{\underline{I}} := \text{Cov}_D(\underline{q})$,

$$\text{Cov}_D(\hat{x}(D)) \geq \underline{\underline{I}}^{-1}, \quad (2.5)$$

where $\underline{\underline{A}} \geq \underline{\underline{B}}$ is the partial ordering defined by the condition that $\underline{\underline{A}} - \underline{\underline{B}}$ is a positive semidefinite matrix. Since the covariance of an unbiased estimator is equal to its risk under the quadratic loss function with $\underline{\underline{Q}} = \mathbb{1}$, the Cramér-Rao bound thus tells us how well we can hope to estimate \underline{x} given a distribution for D .

In Section 2.1.3, we will use the Cramér-Rao Bound to achieve a nearly-optimal strategy for estimation in an important special case.

2.1.2.3 Bayesian Cramér-Rao Bound and Adaptive Experiment Design

In general, the Bayesian mean estimator $\underline{x}(D) := \mathbb{E}[\underline{x}|D]$ may be biased, since the mean over the prior need not agree with the true value. Moreover, the derivative in the definition of the Fisher score \underline{q} is taken about a hypothetical value \underline{x} , such that given only a prior, we will want to marginalize over that hypothesis. Doing so results in the *Bayesian* information matrix (BIM)

$$\underline{\underline{J}}(\pi; e) := \mathbb{E}_{\underline{x} \sim \pi(\underline{x})}[\underline{\underline{I}}; e]. \quad (2.6)$$

Similar to the Cramér-Rao bound, the Bayesian information matrix provides a similar inequality known as the Bayesian Cramér-Rao bound (BCRB) or the van Trees inequality [111; 112],

$$r(\underline{x}, \pi) \geq \underline{\underline{J}}(\pi), \quad (2.7)$$

assuming the risk is defined in terms of the quadratic loss function with $\underline{\underline{Q}} = \mathbb{1}$.

This inequality bounds the Bayes risk directly in terms of quantities that we can, in principle, compute *a priori* given a particular likelihood. Moreover, the BCRB holds for the case in which experiments are designed adaptively, such that $e_{i+1} = e_{i+1}(d_1, \dots, d_i)$. Here, we exploit that the BCRB can be computed iteratively [113], as has been applied in the case of state-space models [114]. In particular, let $\underline{\underline{J}}_N(\pi) := \underline{\underline{J}}(\pi; e_1, \dots, e_N)$ be the BIM derived from taking the expectation value in the defintion of $\underline{\underline{I}}$ over the outcomes of the first N experiments. Then,

$$\underline{\underline{J}}_{N+1}(\pi) = \underline{\underline{J}}(\pi; e_{N+1}) + \underline{\underline{J}}_N(\pi), \quad (2.8)$$

where the base case is given by the information of the prior,

$$\underline{J}_0 := \mathbb{E}_{\underline{x} \sim \pi} [\nabla_{\underline{x}} \ln \pi \cdot \nabla_{\underline{x}}^T \ln \pi] \quad (2.9)$$

Adaptivity can now be expressed by integrating over the posterior at each step, rather than over the initial prior, such that we obtain the adaptive BIM $\underline{J}_{\text{ad},N}$

$$\underline{J}_{\text{ad},N+1}(d_1, \dots, d_N) = \underline{J}(\Pr(\underline{x}|d_1, \dots, d_N); \underline{\varrho}_{N+1}) + \underline{J}_{\text{ad},N}(\pi). \quad (2.10)$$

This definition explicitly allows for experiment design protocols to depend on the best information available at each step; this is known to be necessary [115] and sufficient [31] to obtain exponential improvements in some estimation contexts.

2.1.3 Sampling Periodic Distributions

In some important cases, Bayesian methods for learning Hamiltonians can be implemented analytically [99; 31]. For instance, consider a single qubit prepared in the initial state $|\psi\rangle = |+\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$, and allow the qubit to evolve under $H(\omega) = \omega\sigma_z/2$ for a time t before measuring in the eigenbasis of σ_x . Then, letting $d = 0$ represent the case where the state returns to the $+1$ eigenstate of σ_x , we can obtain a likelihood function from Born's rule,

$$\Pr(d|\omega; t) = \cos^2(\omega t/2). \quad (2.11a)$$

If the qubit is subject to dephasing with time constant $T_2 < \infty$, then the likelihood becomes

$$\Pr(d|\omega; t) = e^{-t/T_2} \cos^2(\omega t/2) + \frac{1 - e^{-t/T_2}}{2}. \quad (2.11b)$$

This model describes *Larmor precession* of a spin- $1/2$ particle about a static field B_0 along the \hat{z} -axis [116]. The frequency of this precession is then $\omega = \gamma B_0$, where γ is the gyromagnetic ratio of the spin undergoing precession. We identify the Larmor model as having a single model parameter $\underline{x} = (\omega)$, and the evolution time as an experimental control $\underline{\varrho} = (t)$.

If we assume that after a number of ``warm-up'' measurements, the posterior over \underline{x} conditioned on those measurements is approximately normal, then we can characterize the state of our knowledge by the mean and variance of the current posterior, μ and

σ , respectively. If the true model is supported by the prior, then as warm-up data is collected, the posterior will be dominated by the likelihood, such that the central limit theorem gives us that the posterior approaches a normal distribution. Then, a Bayes update with a new datum $d \in \{0, 1\}$ will produce a new posterior with mean μ' and variance σ' related to the previous moments by [31]

$$\begin{aligned}\mu' &= \frac{2 \left((2d - 1) e^{-\frac{1}{2} \sigma^2 t^2} (\sigma^2 t \sin(\mu t) - \mu \cos(\mu t)) + \mu \right)}{2 - (2d - 1) (1 + e^{2i\mu t}) e^{-\frac{1}{2} t(\sigma^2 t + 2i\mu)}} \\ \sigma' &= \mu^2 + \sigma^2 - \frac{2 \left((2d - 1) e^{-\frac{1}{2} \sigma^2 t^2} (\sigma^2 t \sin(\mu t) - \mu \cos(\mu t)) + \mu \right)}{2 - (2d - 1) (1 + e^{2i\mu t}) e^{-\frac{1}{2} t(\sigma^2 t + 2i\mu)}} \\ &\quad - \frac{2(2d - 1) \sigma^2 t e^{i\mu t} (\sigma^2 t \cos(\mu t) + 2\mu \sin(\mu t))}{(2d - 1) (1 + e^{2i\mu t}) - 2e^{\frac{1}{2} t(\sigma^2 t + 2i\mu)}}.\end{aligned}\tag{2.12}$$

Importantly, this allows us to *adaptively* design new experiments t . Taking an expectation value of (2.12) over d gives

$$\mathbb{E}_d[\mathbb{V}_{\omega \sim N(\mu, \sigma)}[\omega | d; t]] = \mathbb{E}_d[\mathbb{E}_{\omega \sim N(\mu, \sigma)}[(\omega - \hat{\omega})^2]],\tag{2.13}$$

which we recognize as the **Bayes risk**. We can again analytically perform the expectation over data to obtain that [31]

$$r(\hat{\omega}, N(\mu, \sigma)) = \sigma^2 \left(1 + \frac{t^2 \sigma^2 \sin^2(\mu t)}{\cos^2(\mu t) - e^{t^2 \sigma^2}} \right).\tag{2.14}$$

This function, plotted for two examples in Figure 2.1, admits an envelope

$$E(\mu, \sigma^2) = \sigma^2 \left(1 - t^2 \sigma^2 e^{-t^2 \sigma^2} \right) \leq r(\hat{\omega}, N(\mu, \sigma)).\tag{2.15}$$

The quality of the approximation $r(\hat{\omega}, N(\mu, \sigma^2)) \approx E(\mu, \sigma^2)$ improves as more data is collected, such that the oscillations in r become more dense. Thus, a reasonable adaptive protocol is to choose the next measurement time $t_{k+1} = \arg \min E(\mu_k, \sigma_k^2) = 1/\sigma_k$.

To complete the specification of the protocol, we then update $\mu_k, \sigma_k^2 \mapsto \mu_{k+1}, \sigma_{k+1}^2$ by analytically solving for the expectation and variance of the Bayes update

$$\Pr(\omega | d, \mu_k, \sigma_k^2) = \frac{\Pr(d | \mu_k, \sigma_k^2; t) \Pr(\omega | \mu_k, \sigma_k^2)}{\Pr(d | t)}.$$

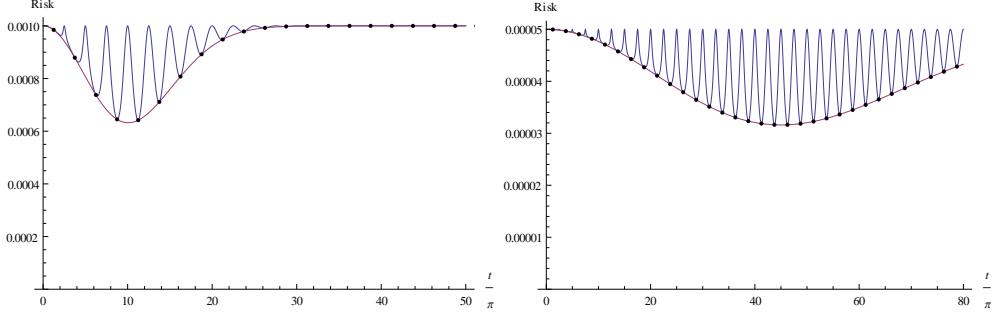


Figure 2.1: Bayes risk and envelope for Larmor precession model, assuming normal prior $\omega \sim N(\mu, \sigma^2)$, with $\mu = 0.4$. (left) $\sigma^2 = 10^{-3}$, (right) $\sigma = 5 \times 10^{-5}$.

Using a computer algebra system such as Mathematica, we can quickly find that

$$\mathbb{E}[\omega|d, \mu_k, \sigma_k^2] = \mu_k + \frac{\pi(2d-1)(-1)^l \sigma_k^2 (2l-1) \exp\left(-\frac{\pi^2 \sigma_k^2 (1-2l)^2}{8\mu_k^2}\right)}{2\mu} \quad (2.16a)$$

$$\mathbb{V}[\omega|d, \mu_k, \sigma_k^2] = \sigma_k^2 - \frac{\pi^2 (2d-1)^2 \sigma_k^4 (2l-1)^2 \exp\left(-\frac{\pi^2 \sigma_k^2 (1-2l)^2}{4\mu_k^2}\right)}{4\mu^2}, \quad (2.16b)$$

where $l = \text{round}\left[\frac{\mu_k}{\pi\sigma_k} + \frac{1}{2}\right]$ describes the intersections of the risk r and the envelope E . In the asymptotic case $k \rightarrow \infty$,

$$\frac{r(\hat{\omega}, N(\mu_k, \sigma_k^2))}{\sigma_k^2} \rightarrow 1 - e^{-1} \approx 0.632, \quad (2.17)$$

and is achieved by

$$t_k \sim \frac{1}{\sigma_0 (1 - e^{-1})^{k/2}} \approx \frac{1.26^k}{\sigma_0}. \quad (2.18)$$

Thus, the risk decays exponentially, better than any nonadaptive protocol [115]. We note that this is near the best possible scaling, since each bit of experimental data can reveal at most one bit of ω . This precise scaling is only valid in the asymptotic limit, but the moments of the Bayes updated posterior (2.16) hold for finite numbers of samples, insofar

as the assumption of a normal prior approximately holds, as will be the case following the ``warm-up'' measurements.

The Cramér-Rao bound for this model can also be calculated explicitly, giving that

$$I(\omega|t_1, \dots, t_k) = \sum_i t_i^2. \quad (2.19)$$

Notice that I is independent of ω , such that the expectation over ω leaves I invariant. Thus, the Bayesian and frequentist CR bounds agree in this case, and give an ultimate scaling based on how long one evolves for. This shows that the generally-applied strategy of sampling repeatedly at each of several Nyquist times $t_{k,\text{Nyq}} = k/(2\omega_{\max})$ is quite suboptimal given the guarantee of a single mode in the likelihood, as it yields the CRB

$$r_{\text{Nyq}}(k) \geq \frac{1}{4n\omega_{\max}^2 \sum_i t_i^2} \in O(k^{-3}), \quad (2.20)$$

where n is the number of samples collected at each time. While it may seem surprising that our protocol dramatically outperforms Nyquist-Shannon sampling, it is worth noting that we are not subject to the preconditions of the Sampling Theorem [117] in two important ways. First, we are not interested in a complete reconstruction of an arbitrary likelihood supported over a window in Fourier domain, but are interested in learning a single parameter. Second, we are not directly sampling a *function*, but a family of probability distributions related by that function. Because of this, intuition developed from the case described by Nyquist-Shannon need not apply to our problem, especially in light of adaptive protocols [115].

If we do not have perfect measurement, but instead measure the final outcome with some finite visibility $\eta \in [0, 1]$, then the visibility simply factors out of the CR bound, giving $I = \eta^2 \sum_i t_i^2$. Thus, we do not lose anything of interest to the inference procedure by focusing on the case $\eta = 1$.

In general, $\sum_i t_i^2$ is an *upper* bound for the Fisher information that can be extracted from measuring after evolving under a Hamiltonian for times $\{t_1, \dots, t_i\}$ [34]. Thus, the advantage we derive in this case comes not only from adaptive experiment design and Bayesian approaches to data processing, but also from our ability to evolve for long times. Our approach thus gives the most obvious advantage in systems where the reset time is comparable to evolution times, or is even long by comparison. In particular, our approach is useful for examples such as NV centers (Section 2.3.3), which have long decoherence times and significant reset times, and can be processed on a shot-by-shot basis. The case in which $T_2 < \infty$ is considered in depth in [31].

This approach to sampling Larmor precessions has recently been employed to significantly reduce data collection costs in quantum dots [98], giving much better performance with relation to the bounds given by T_2 .

2.2 Sequential Monte Carlo

Having thus established the utility of a Bayesian approach to Hamiltonian learning in an important special case that admits analytic solutions, we are interested in generalizing and applying Bayesian approaches to other examples as well. To do so, we turn to the sequential Monte Carlo algorithm [118], which uses a novel approximation of the posterior distribution to reduce inference to repeated calls to a black-box simulator.

In this way, we build on the insight that we have developed in [Section 2.1.3](#): in particular, that access to a [likelihood function](#) is a critical resource for learning, as it gives us access to the [posterior](#) distribution.

2.2.1 Overview of Algorithm

2.2.1.1 Particle Approximation

The sequential Monte Carlo (SMC) algorithm [118], also known as a *particle filter*, accomplishes reduces learning to simulation by representing distributions as weighted sums of δ -distributions,

$$\Pr(\underline{x}) \approx \sum_{i=1}^n w_i \delta(\underline{x} - \underline{x}_i), \quad (2.21)$$

where $\{w_i\}_{i=1}^n \subset \mathbb{R}$ are the *weights* of *particles* located at $\{\underline{x}_i\}_{i=1}^n \subset \mathbb{R}^d$. Initially, all of the weights are set to be uniform, $w_i = 1/n$. Updating this distribution given a new datum d then can be represented as an operation on the weights,

$$w_i \mapsto w_i \times \Pr(d|\underline{x}_i)/\mathcal{N}, \quad (2.22)$$

where \mathcal{N} is a normalization constant such that the updated weights obey $\sum_i w_i = 1$. The update can thus be performed using n calls to the likelihood function. Moreover, the SMC approximation is closed under Bayes updates for all likelihoods, such that the current state of the algorithm is represented by recording the particle weights and locations. This is summarized in [Figure 2.2](#) and is illustrated in [Figure 2.3](#).

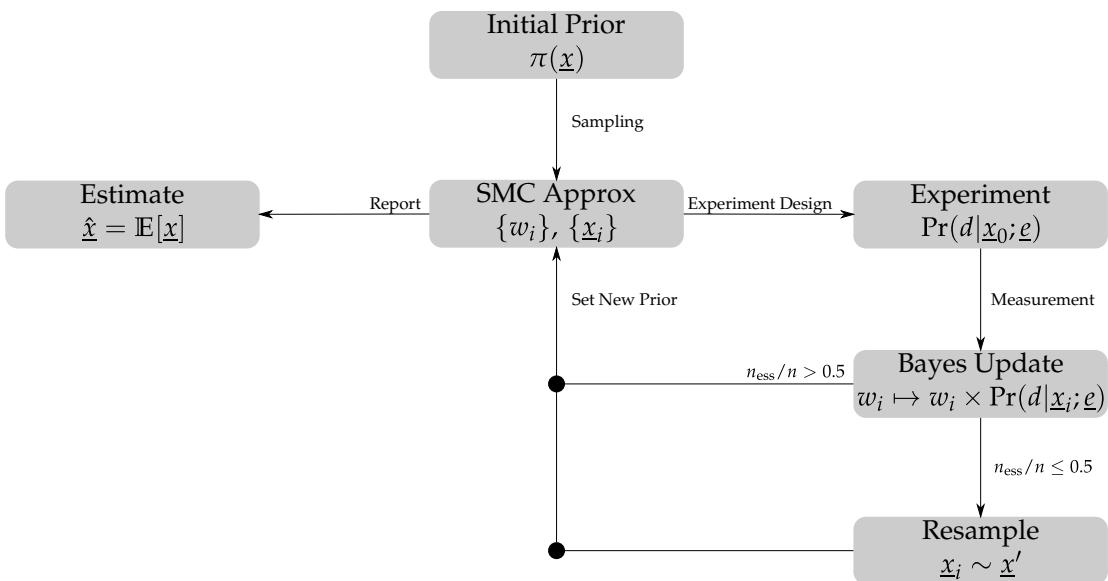


Figure 2.2: Flow of data and hypotheses in the SMC algorithm. An initial prior $\pi(\underline{x})$ is sampled to produce the initial SMC approximation $\{\underline{x}_i\} \sim \pi$, which is then used to adaptively or heuristically design experiments. The results of experiments are used to update this approximation, and resampling is applied if necessary to preserve the effective sample size n_{ess} . At each iteration, the current best estimate is reported. Each step of this process is explained in more detail throughout [Section 2.2](#).

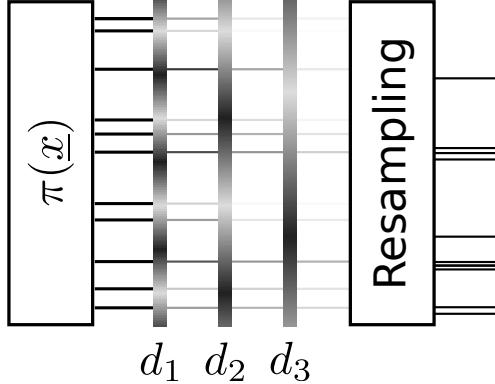


Figure 2.3: Flow of data through the SMC algorithm for a periodic likelihood. The gray bands indicate likelihood evaluations, while the horizontal lines indicate particles; lighter particles are less parsimonious with the observations $D = \{d_1, d_2, d_3\}$. Particles are effectively *filtered* by the likelihood evaluations to form posterior approximations.

Thus, sequential Monte Carlo is an *iterative* algorithm, and can be readily used either online or in postprocessing. The memory requirements of SMC thus do not grow as data is collected, such that SMC can be usefully applied even to very large datasets. Moreover, the parallel nature of the Bayes update step (2.22) means that SMC can be accelerated with the use of parallel computing techniques such as symmetric multiprocessing, GPU-based computing or FPGAs [119]. Indeed, both GPUs [120; 121] and FPGAs [122] have been useful in simulating quantum systems, such that an advantage may reasonably be expected to extend to sequential Monte Carlo implementations as well, offering the additional benefit of integrating with newly-developed techniques in experimental control [123].

In particular, we note that there is no explicit dependence between $\Pr(d|\underline{x}_i; e)$ and $\Pr(d|\underline{x}_j; e)$ for distinct i and j , such that model parameter vectors can be readily distributed to evaluate the likelihood function for each SMC particle. Given that in typical applications, this cost dominates all others, parallelization at this step is effective in mitigating the limit on bandwidth that the computational cost of SMC would impose in a purely serial context.

Because of these advantages, sequential Monte Carlo has recently been applied in a wide variety of classical contexts [124; 125; 126], as well as in quantum information [59; 32; 127]. To implement SMC in quantum information, we have developed QInfer, an open-source Python-based implementation of SMC [44]. Our library is documented

in [Appendix D](#), and we will give examples through the rest of this Section.

2.2.1.2 Expectation Values and Covariances

As a distribution is updated, [estimates](#) can be found online by taking expectation values over the current posterior. Because the SMC approximation uses a mixture of δ -distributions, this expectation takes the form of a finite sum,

$$\hat{x} := \mathbb{E}[x|D] = \sum_i w_i \underline{x}_i. \quad (2.23)$$

Similarly, the covariance of the current posterior can be found by taking finite sums,

$$\text{Cov}(\underline{x}|D) = \mathbb{E}[\underline{x}\underline{x}^T|D] - \mathbb{E}[\underline{x}]\mathbb{E}[\underline{x}^T|D] = \sum_i w_i \underline{x}_i \underline{x}_i^T - \hat{x} \hat{x}^T. \quad (2.24)$$

Note, however, that in analogy to the argument made in [Section 2.1.3](#), $\text{Cov}(x|D)$ can be equivalently defined as the expected loss of the Bayesian mean [estimator](#),

$$\begin{aligned} \text{Cov}(x|D) &= \mathbb{E} \left[(\underline{x} - \mathbb{E}[\underline{x}|D])(\underline{x} - \mathbb{E}[\underline{x}|D])^T \mid D \right] \\ &= \mathbb{E} \left[(\underline{x} - \hat{x}(D))(\underline{x} - \hat{x}(D))^T \mid D \right] = \mathbb{E}[L_1(\underline{x}, \hat{x}(D))]. \end{aligned} \quad (2.25)$$

In several different examples, explored in [Section 2.3.1](#) and [Section 2.3.2](#), we will see that given enough particles, the estimated loss tracks the actual loss quite well, giving a convenient means to quickly assess the performance of the algorithm. Moreover, in [Section 3.1](#), we will use the covariance matrix for \underline{x} to form a [credible region](#) estimator, such that the covariance can be seen as a characterization of the estimation errors.

The particle approximation also allows for the expedient computation of marginal distributions over some subset of the model parameters. For instance, if $\dim \underline{x} = n_{\text{mp}} > 1$, then the marginal over the first parameter is given by the observation that

$$\delta(\underline{x} - \underline{x}_i) = \prod_{j=1}^{n_{\text{mp}}} \delta(x_j - x_{i;j}),$$

and thus that

$$\mathbb{E}_{x_1} [\delta(\underline{x} - \underline{x}_i)] = \int_{X_1} \prod_{j=1}^{n_{\text{mp}}} \delta(x_j - x_{i;j}) dx_1 = \prod_{j=2}^{n_{\text{mp}}} \delta(x_j - x_{i;j}),$$

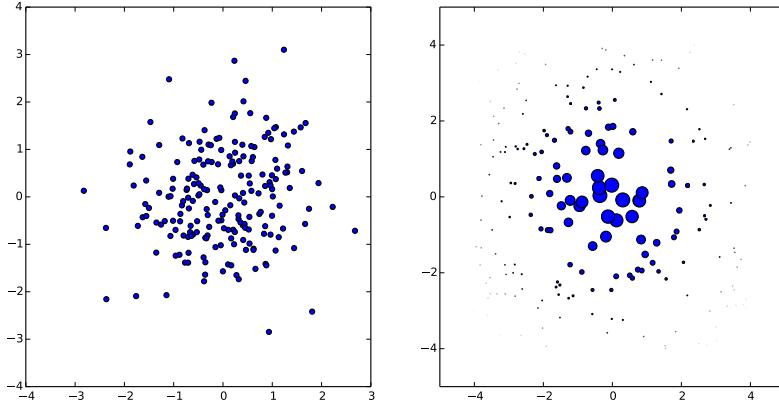


Figure 2.4: Two different representations of the multinormal distribution $N(0, \mathbb{I})$, using particle density (left) and particle weights (right).

where X_1 is the range of valid values for x_1 , the first element of \underline{x} . That is, by *slicing* the model parameter vectors, marginals can be efficiently computed. This in turn is useful for implementing plotting functionality, as for anything but quite small models, visualizing posterior distributions is quite difficult.

2.2.1.3 Numerical Stability and Impoverishment

A particular distribution may be represented in SMC using particles of uniform weight, storing information in the particle density, or by using the weights to carry information. These two extremes are shown in Figure 2.4. Storing information in particle weights results in some of the particles having much less of an impact on expectation values than others, such that the *effective* number of particles is reduced. This is made formal by defining the effective sample size n_{ess} , [118; 128]

$$n_{\text{ess}} := \frac{1}{\sum_i w_i^2}. \quad (2.26)$$

Since Bayes updates map the particle weights to new weights that are more strongly peaked, as more experiments are performed, $n_{\text{ess}} \rightarrow 0$, such that an additional step is needed to recover numerical stability.

2.2.2 Resampling

To move information from particle weights into particle locations, we can resample the current posterior, drawing new samples from a resampling distribution and assigning uniform weights to the new samples. In this way, we guarantee that $n_{\text{ess}} = n$ after resampling.

The Liu and West resampling algorithm [129] does this by drawing the new samples from a kernel density estimate adaptively chosen from the covariance of the current posterior [130]. In particular, each new sample \underline{x}' is distributed according to

$$\Pr(\underline{x}') = \sum_i w_i \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2} (\underline{x}' - \underline{\mu}_i)^T \underline{\Sigma}^{-1} (\underline{x}' - \underline{\mu}_i)\right) \quad (2.27a)$$

$$\underline{\mu}_i = a\underline{x}_i + (1-a)\mathbb{E}[\underline{x}] \quad (2.27b)$$

$$\underline{\Sigma} = h \text{Cov}(\underline{x}), \quad (2.27c)$$

where $d = \dim \underline{x}$, and where a and h are parameters that can be set to modify the behavior of the resampler. Setting $a^2 + h^2 = 1$ ensures that $\text{Cov}(\underline{x}') = \text{Cov}(\underline{x})$, preserving the uncertainty in the posterior distribution. Pseudocode for an implementation of the Liu and West algorithm is provided in [Algorithm 2](#). Performing a resampling step when $n_{\text{ess}}/n \leq 0.5$ is often sufficient to preserve numerical stability. Checking this condition immediately following each Bayes update gives the update step described in [Algorithm 1](#).

Algorithm 1 Sequential Monte Carlo update algorithm.

Input: Particle weights $w_i(D)$, $i \in \{1, \dots, n\}$, Particle locations \underline{x}_i , $i \in \{1, \dots, n\}$, New datum d_{j+1} , obtained from an experiment with control c_{j+1} .

Output: Updated weights $w_i(D \cup d_{j+1})$.

```

function UPDATE( $\{w_i(D)\}$ ,  $\{\underline{x}_i\}$ ,  $d_{j+1}$ ,  $c_{j+1}$ )
    for  $i \in 1 \rightarrow n$  do
         $\tilde{w}_i \leftarrow w_i(D) \Pr(d_{j+1} | \underline{x}_i, c_{j+1})$ 
    end for
    return  $\{\tilde{w}_j / \sum_i \tilde{w}_i\}$             $\triangleright$  We must normalize the updated weights before returning.
end function
```

Effectively, choosing $a < 1$ mixes the current posterior with a Gaussian distribution such that the new particles are located near old particles, each perturbed by a normal random variate. This allows the posterior distribution to explore the space of model

parameters with a finite number of samples. Liu and West recommend $a = 0.98$, while in some applications, the resampler can be made significantly more aggressive about mixing with a normal distribution; in [41; 131], $a = 0.9$ was found to be effective at reducing the number of particles required, owing to the approximate normality in the model considered.

Using $a < 1$ can fail, however, for multimodal distributions, in that the Liu and West resampler may then move particles away from high density regions in the true posterior. In Figure 2.5, we demonstrate this using the likelihood (2.11) with the prior $\pi(\omega) = 1/2$ for $\omega \in [-1, 1]$. Because $\cos^2(\omega t)$ is an even function of ω , this is a manifestly bimodal problem, violating the assumptions of the Liu-West algorithm when $a^2 + h^2 = 1$. For both subfigures, $t_k = (9/8)^k$ and 50 data points are collected, as shown in Listing C.1.

This failure modality can be mitigated in the limit that $a \rightarrow 1$. In particular, $a = 1$ and $h = 0$, yields the bootstrap filter [132], similar to that used in some computer vision algorithms [133]. The bootstrap filter is effective in state-space models [134], in which parameters vary in time, but is less effective for parameter estimation, as multiplicity of the new particles does not result in further exploration of the parameter space. This in turn can be mitigated by adding additional variance to the resampling distribution, thus emulating state-space methods. In particular, we let $a = 1$ and $h > 0$, such that the additional variance is proportional to the current variance of the posterior [130]. We demonstrate this approach in Figure 2.5 by setting $a = 1$ and $h = 0.005$.

Algorithm 2 Liu and West resampling algorithm.

Input: Particle weights $w_i, i \in \{1, \dots, n\}$, Particle locations $\underline{x}_i, i \in \{1, \dots, n\}$, Resampling parameters $a, h \in [0, 1]$.

Output: Updated weights w'_i and locations \underline{x}'_i .

```

function RESAMPLE( $\{w_i\}, \{\underline{x}_i\}, a, h$ )
     $\mu \leftarrow \text{MEAN}(\{w_i\}, \{\underline{x}_i\})$ ,  $\underline{\Sigma} \leftarrow h^2 \text{Cov}(\{w_i\}, \{\underline{x}_i\})$ 
    for  $i \in 1 \rightarrow n$  do
        draw  $j$  with probability  $w_j$                                  $\triangleright$  Choose a particle  $j$  to perturb.
         $\underline{\mu}_i \leftarrow a\underline{x}_j + (1 - a)\underline{\mu}$            $\triangleright$  Find the mean for the new particle location.
        draw  $\underline{x}'_i$  from  $\mathcal{N}(\underline{\mu}_i, \underline{\Sigma})$             $\triangleright$  Draw a perturbed particle location.
         $w'_i \leftarrow 1/n$                                           $\triangleright$  Reset the weights to uniform.
    end for
    return  $\{w'_i\}, \{\underline{x}'_i\}$ 
end function
```

In practice, when implementing resampling, we must also *postselect* on valid models.

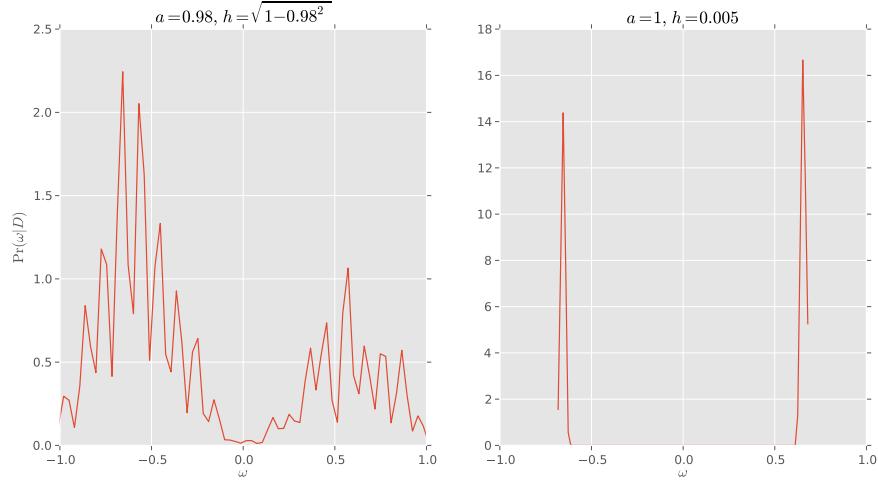


Figure 2.5: Failure of Liu-West mode ($a^2 + h^2 = 1$) for bimodal posterior, contrasted with $a = 1, h > 0$ modification.

For example, if one of the parameters in a model represents a probability, then resampled particles that lie outside of $[0, 1]$ must be rejected. This postselection is performed by QInfer, based on the validity constraints defined by the user. In cases where the same model is identified by more than one model parameter vector, we can also *canonicalize* model parameter vectors to prevent resampling problems arising from exactly-degenerate models. In either postselection or canonicalization, however, we explicitly violate that the mean and variance of the pre- and post-resampling distributions are identical, as is guaranteed by the Liu-West algorithm. Thus, care must be taken with distributions that lie close to postselection or canonicalization boundaries.

2.2.3 Adaptive Experiment Design

With access to particle-approximated distributions over the model parameters, SMC also enables reasoning about the [Bayes risk](#) for future experiments in an online fashion. As illustrated in [Figure 2.6](#), after each Bayes update, the resultant posterior can be used to guide the next decision, such that new experiments can be chosen that minimize the Bayes risk.

Doing so requires performing hypothetical Bayes updates for each possible outcome

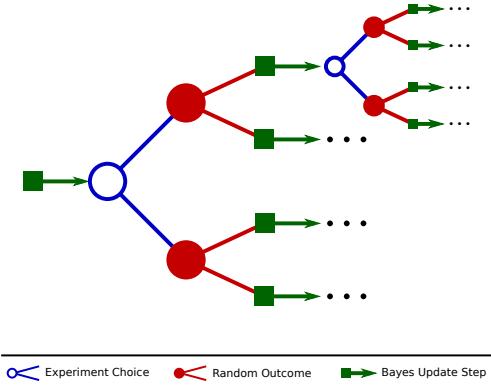


Figure 2.6: Decision process for adaptive experiment design. As random outcomes are observed (open circles), the posterior distribution is updated using Bayes' rule (filled squares), and a decision as to the next experiment is made (filled circles). Each such decision then results in a new random outcome, which is used in turn to update the posterior.

of a proposed measurement, then taking the expectation value over the unobserved datum; this is a computationally-expensive task, as it requires performing an SMC update for each hypothetical outcome and for each experiment design candidate, but may offer advantages when the experiment itself is expensive. Moreover, approximation techniques exist that use lower-quality approximations of the current posterior for designing experiments [32].

As noted in Section 2.1.2.3, the Bayesian Cramér-Rao bound in this case can be computed online, such that adaptive performance can also be bounded. For likelihood models which define a `Model.score` method, QInfer will do this automatically, and will compare the adaptive and non-adaptive strategies. The `ScoreMixin` class¹ will provide this functionality by numerically estimating gradients, such that a specification of the likelihood function is sufficient to find Cramér-Rao bounds for low-dimensional models.

Additionally, some of the simulation cost may be mitigated by choosing a subset of the full particle set to use in optimization [32], or by performing Monte Carlo sampling over the possible outcomes as well as over model parameters.

¹Contributed to QInfer by Ian Hincks.

2.2.4 Hyperparameters and State-Space Methods

Thus far, we have assumed that the **model parameters** \underline{x} have some true, constant value x_0 that is randomly sampled from a prior distribution. This need not be the case in general, however. The model parameters could, for instance, be sampled from a distribution with each distinct experiment, or could follow a stochastic process.

In the former case, we can model this by positing the existence of another set of parameters \underline{y} , known as hyperparameters, such that \underline{x} is sampled from a distribution parameterized by \underline{y} ,

$$\underline{x} \sim \Pr(\underline{x}|\underline{y}).$$

We then run SMC not on the model parameters, but directly on the hyperparameters, such that our new likelihood function is a marginalization over the unknown sample \underline{x} ,

$$\Pr(d|\underline{y}; e) = \mathbb{E}_{\underline{x} \sim \underline{y}}[\Pr(d|\underline{x}; e)]. \quad (2.28)$$

Note that we have assumed in this derivation that d is conditionally independent of the hyperparameters \underline{y} given the model parameter vector \underline{x} ($d \perp \underline{y} \mid \underline{x}$); that is, that if we know the model parameters \underline{x} precisely for some experiment, then the hyperparameters have no further effect on the outcome of that experiment. In this way, the hyperparameters' only role is to determine the distribution over \underline{x} .

In some cases, we can perform the marginalization in (2.28) analytically to obtain a likelihood for \underline{y} . For the Larmor precession model given by (2.11), for instance, we can allow $\omega \sim N(\mu, \sigma^2)$ to obtain a new likelihood function with $\underline{y} = (\mu, \sigma^2)$,

$$\Pr(0|\underline{y}; e) = \frac{1}{2} \left(1 + e^{-2\sigma^2 t^2} \cos(2\mu t) \right). \quad (2.29)$$

Assuming ω is instead drawn from a Cauchy distribution with location ω_0 and scale γ gives

$$\Pr(0|\underline{y}; e) = \frac{1}{2} \left(1 + e^{-t\gamma} \cos(t\omega_0) \right). \quad (2.30)$$

Identifying $\gamma = T_2^{-1}$, we see that the Cauchy hyperparameter model is equivalent to an unknown- T_2 model. We note that no such identification can be easily made for the Gaussian hyperparameter model, as the t^2 dependence prevents writing that model as the result of evolving under a Lindblad master equation. We will see later, in Section 3.1, that hyperparameters can be connected to region estimation, such that we can reason about which portion of the variance in \underline{x} can be explained by sampling according to \underline{y} , and which portion can be explained by a lack of knowledge about \underline{y} .

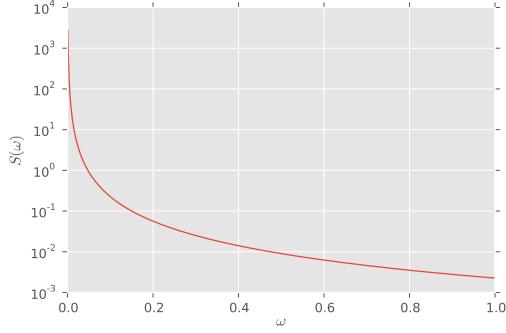


Figure 2.7: Spectral density function used to generate realizations for state-space tracking example.

We can also consider the case where there are no explicit hyperparameters, but where \underline{x} is not constant in time. Indeed, in many cases of interest, the parameters being learned *drift* as experiments are performed [98], such that failing to incorporate this into the model can cause the inference procedure to assign false certainty to an estimate. This will be the case, for instance, in our study of nitrogen-vacancy centers (Section 2.3.3), wherein the visibility is a dynamic parameter that follows a random walk. One way of incorporating drift or diffusion in model parameters is to, in addition to Bayes updating particle weights, performing a diffusive update [133; 135]

$$\underline{x}(t_{k+1}) \sim \Pr(\underline{x}(t_{k+1}) | \underline{x}(t_k)). \quad (2.31)$$

If no parameters of \underline{x} are left constant by this diffusive update, then the bootstrap filter ($a = 1, h = 0$) discussed in Section 2.2.2 no longer fails to increase the effective sample size, as the multiplicity of particles is broken by sampling the diffusive rule.

We demonstrate state-space tracking by sampling a ``true'' trajectory for ω from the Lorentz spectral density function

$$S(\omega) = \frac{2A^2}{\tau_c} \cdot \frac{1}{(\omega - \omega_0)^2 + (1/\tau_c)^2}, \quad (2.32)$$

using the method of [136], where $A = 150$, $\tau_c = 20$ and $\omega_0 = 0.1$ are chosen to represent a ``slow'' drift. The spectral density and an example trajectory are shown in Figure 2.7. We then track the slow drift using a normal distribution $\omega(t_{k+1}) | \omega(t_k) \sim N(\omega(t_k), (t_{k+1} - t_k)\sigma^2)$ as the diffusive update rule for a model parameter vector $\underline{x} = (\omega, \sigma^2)$. The QInfer model for this experiment is shown in Listing C.3. The results for an SMC estimate of the

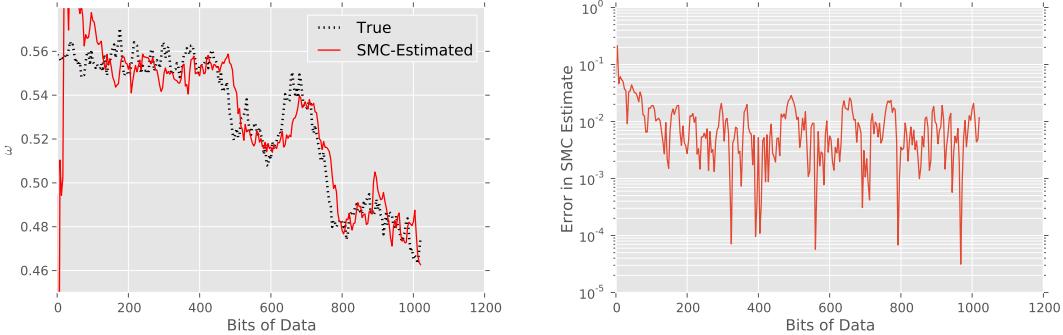


Figure 2.8: Tracking of a stochastic process, using a diffusive update in SMC. (Left) True and SMC-estimated trajectories. (Right) Error in SMC-estimated trajectory.

trajectory are shown in Figure 2.8. Each experiment is chosen according to the heuristic $t_k = \pi \min(k, 50 + w)/16$, where w is a random integer mod 16. This heuristic warms up to the maximum value, then selects random times near that maximum so as to break accidental patterns from repeated sampling at the same interval.

Note that we include σ^2 as a model parameter, even though the Larmor precession likelihood does not depend directly on the value of σ^2 . Instead, we rely on coevolution between the drift variance parameter σ^2 and the frequency parameter ω to inform about the rate at which ω drifts. That is, the diffusive step in the SMC algorithm effectively couples σ^2 and ω , such that the Bayes update will provide information about σ^2 through this correlation. We will see a similar effect later, when we consider using evolutionary strategies to couple optimization parameters to pulses in a control design algorithm (Section 6.1.3).

2.3 Examples of SMC

2.3.1 Robust Hamiltonian Learning

The primary application of SMC that we will be interested in is its ability to directly estimate parameters of unknown Hamiltonians $H(x)$. In this section, we show several examples of this application with data drawn from simulated and experimental models.

Algorithm 3 Complete adaptive Bayesian experiment design algorithm, using sequential Monte Carlo approximations.

Input: A number of particles n to be used, A prior distribution π over models, A number of experiments N to perform, A resampling parameter $a \in [0, 1]$, A threshold $\text{resample_threshold} \in [0, 1]$ specifying how often to resample, A local optimization algorithm `LOCALOPTIMIZE`, A heuristic `GUESEXPERIMENT` for choosing experiment controls, and a number n_{guesses} of potential experiments to consider in each iteration.
Output: An estimate \hat{x} of the true model x_0 .

```

function ESTIMATEADAPTIVE( $n, \pi, N, a, \text{resample\_threshold}, \text{approx\_ratio}, \text{OPTIMIZE}, n_{\text{guesses}}, \text{GUESEXPERIMENT}$ )
     $w_i \leftarrow 1/n$                                       $\triangleright$  Start by initializing the SMC variables.
    draw each  $\underline{x}_i$  independently from  $\pi$ 

    for  $i_{\text{exp}} \in 1 \rightarrow N$  do                       $\triangleright$  We now iterate through each experiment.
         $\triangleright$  Heuristically choose potential experiments, and optimize each independently.
        for  $i_{\text{guess}} \in 1 \rightarrow n_{\text{guesses}}$  do
             $c_{i_{\text{guess}}} \leftarrow \text{GUESEXPERIMENT}(i_{\text{exp}})$ 
             $\hat{c}_{i_{\text{guess}}}, U_{i_{\text{guess}}} \leftarrow \text{LOCALOPTIMIZE}(\text{UTILITY}, c_{i_{\text{guess}}}, \{w_i\}, \{\underline{x}_i\})$ 
        end for

         $i_{\text{best}} \leftarrow \text{argmax}_{i_{\text{guess}}} U_{i_{\text{guess}}}$            $\triangleright$  Pick the controls that maximize the optimized utility.
         $\hat{c} \leftarrow \hat{c}_{i_{\text{best}}}$ 
         $d_{i_{\text{exp}}} \leftarrow \text{the result of performing } \hat{c}$             $\triangleright$  Perform the best experiment.
         $\{w_i\}, \{\underline{x}_i\} \leftarrow \text{UPDATE}(\{w_i\}, \{\underline{x}_i\}, D, C)$        $\triangleright$  Find the new posterior distribution.

        if  $\sum_i w_i^2 < N \cdot \text{resample\_threshold}$  then           $\triangleright$  Resample if  $n_{\text{ess}}$  is too small.
             $\{w_i\}, \{\underline{x}_i\} \leftarrow \text{RESAMPLE}(\{w_i\}, \{\underline{x}_i\}, a, h)$ 
        end if
    end for

     $\triangleright$  After all experiments have been performed, return the mean as an estimate.
    return  $\hat{x} \leftarrow \text{MEAN}(\{w_i\}, \{\underline{x}_i\})$ 

end function

```

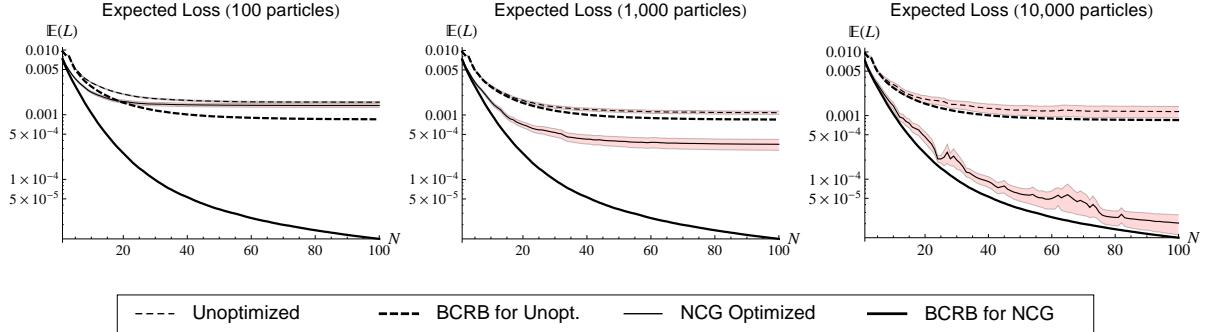


Figure 2.9: Left to right: the mean-squared error, as a function of the number of measurements N , of the sequential Monte Carlo algorithm for $n_{\text{mp}} = 100, 1\,000$, and 10 000 particles. Samples are drawn from the likelihood (2.11) with $T_2 = 100\pi$. The dashed lines indicate data taken without optimization, while the solid lines indicate trials in which initial guesses were optimized using the NCG method. For each data set, the corresponding thick line indicates the Bayesian Cramer-Rao bound. The red regions include the performance of 95% of the trials.

2.3.1.1 Larmor Model, Known and Unknown T_2

The first example that we consider is that of Section 2.1.3, so that we can compare the performance of SMC to the known optimal solutions for sampling a Larmor precession model. The implementation of this model in QInfer is provided in Listing C.2.

In all the examples in this section, we use the likelihood model of (2.11) with the units chosen such that $\pi(\omega) = \text{Uni}(0, 1)$. We first consider the case in which $T_2 = 100\pi$ is known. This choice of T_2 demonstrates both the exponential learning rate in the $t_k \ll T_2$ regime and the $1/\sqrt{k}$ learning rate for $t_k \approx T_2$, as predicted (2.14). We then vary over the number of particles, comparing to the optimum given by the CRB and to the posterior variance. From Figure 2.9, we see that only 1,000 particles are needed to obtain near-optimal error scaling in the case where the $t_k = (9/8)^k$ heuristic is used.

We also show in Figure 2.9 the performance that we obtain when we optimize the Bayes risk as estimated by SMC, using the Newton Conjugate-Gradient method as implemented by SciPy [137] to perform the minimization. In this case, we see that more particles are needed to optimize the choice of experiment than are needed to simply process data offline.

In Figure 2.10, we plot the mean-squared error loss not against the number of experiments performed, but against the number of simulator calls n_{like} that we must for

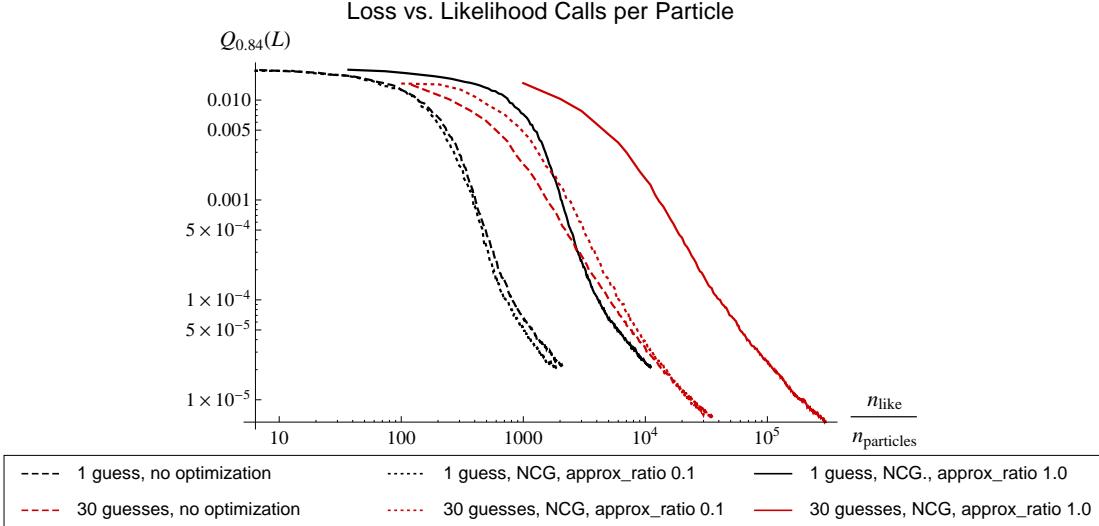


Figure 2.10: Comparison of mean-squared error as a function of the number of likelihood calls per particle, drawing from the Larmor model (2.11) with known $T_2 = 100\pi$ and with 5,000 particles. The parameter `approx_ratio` controls the proportion of the posterior particles used in optimizing, as described in [32]. The expected loss incurred by each optimization strategy is shown in terms of the 84th percentile $Q_{0.84}$ of the loss, such that no more than 16% of trials incur loss greater than the shown percentile.

each particle make to obtain that loss. This shows that even though local optimization of the Bayes risk about some number of initial guesses can be useful, it is computationally expensive such that if experiments are relatively inexpensive, less informative experiments may be preferable to the cost of finding good experiments. In some cases, such as considered in Section 2.1.3 and Chapter 6, this can be mitigated by heuristics that do not depend on likelihood calls.

Having thus compared to the case where we can analytically derive good adaptive strategies, we now expand our model to include unknown T_2 . In including T_2 , it is convenient to choose a parameterization with the same units as ω , such that the **model parameters** are $\underline{x} = (\omega, T_2^{-1})$.

Even as small a change as this breaks the strategies outlined in Section 2.1.3, as the Fisher information for each datum is a singular matrix and cannot be inverted to find a Cramer-Rao bound. Instead, we must consider longer data records, or must integrate over a prior. Both of these are difficult to do analytically, such that SMC can provide an advantage.

In [Figure 2.11](#), we show the performance of SMC for the unknown- T_2 Larmor model, demonstrating that significant gains can be made in learning T_2 . These results also are consistent with the observation that because T_2 manifests as an effective visibility in [\(2.11\)](#), that the parameter is exponentially more difficult to learn than a frequency parameter, such that we will see a ``corner'' in the learning rate once T_2 dominates the quadratic loss over ω .

2.3.1.2 Correlated Model

We can also apply SMC for higher-dimensional models, such as for a coupling between two spin- $1/2$ particles, using a correlation spectroscopy (COSY) experiment [\[138\]](#). In the Jeener COSY experiment, a nonselective pulse is used to induce a transfer of coherence, such that a measurement of the two spins becomes correlated. Using a two-dimensional Fourier transform, the presence or absence of an off-diagonal peak then provides evidence of a coupling between the two spins, such as a dipolar or scalar coupling [\[116\]](#). Multi-dimensional nuclear magnetic resonance experiments such as COSY have motivated the use of exponentially-sparse design heuristics and maximum-entropy reconstruction in the case of [\[139; 140\]](#). Here, we will show that similar sampling strategies can be applied to a model with projective measurement by using SMC instead of maximum-entropy methods.

In particular, we consider a simplified three-parameter correlated model derived from the Hamiltonian

$$H(J, \omega_1, \omega_2) = \frac{\omega_1}{2} \sigma_z^{(1)} + \frac{\omega_2}{2} \sigma_z^{(2)} + J \sigma_z^{(1)} \sigma_z^{(2)}. \quad (2.33)$$

We then prepare an initial state $|+\rangle \otimes |+\rangle$ and evolve first under $H(0, \omega_1, 0)$ for a time t_1 , under $H(0, 0, \omega_2)$ for t_2 , then under $H(J, 0, 0)$ for $t_1 + t_2$. An example sequence implementing this evolution is given in [Figure 2.12](#). Finally, we measure against the initial state $|+\rangle \otimes |+\rangle$. This gives a likelihood for our model which correlates the three model parameters,

$$\begin{aligned} \Pr(0|J, \omega_1, \omega_2; t_1, t_2) = & |\cos(J(t_1 + t_2)) \cos(t_1 \omega_1 / 2) \cos(t_2 \omega_2 / 2) + \\ & i \sin(J(t_1 + t_2)) \sin(t_1 \omega_1 / 2) \sin(t_2 \omega_2 / 2)|^2 \end{aligned} \quad (2.34)$$

This model serves to demonstrate how a strongly correlated model behaves when analyzed with SMC. In particular, this model allows us to decide whether the resonances at ω_1 and ω_2 are connected.

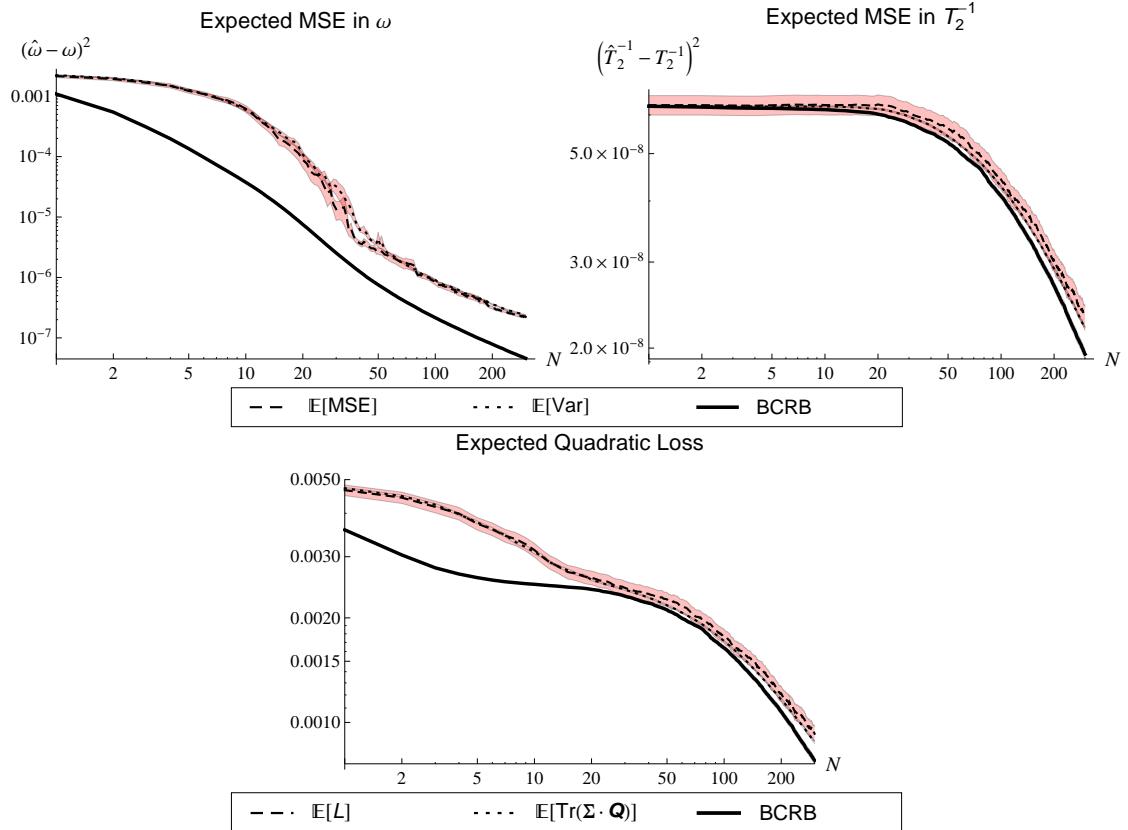


Figure 2.11: The estimated, actual and optimal performance of SMC drawing N measurements from the Larmor precession model (2.11) with unknown T_2 . The mean-squared error is shown for each parameter individually, along with the quadratic loss for $\underline{Q} = \text{diag}(1, 100)$. The prior is taken to be $\pi(\omega, T_2^{-1}) \sim N([0.5, 0.001], \text{diag}(0.0025, 0.00025^2))$. For the optimization, 30 initial guesses are drawn from an exponential distribution with mean 1,000 and then improved using NCG.

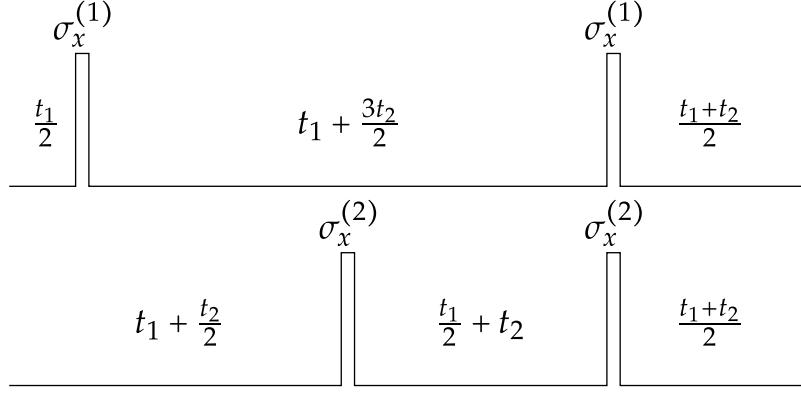


Figure 2.12: Gates implementing correlated model of (2.34). Each of the two qubits is shown as a line, with σ_x gates drawn as pulses. The lengths of each evolution time are shown between the gates.

As an experiment design heuristic, we choose an exponentially-sparse sampling strategy

$$t_{1,k}, t_{2,k} = \cos(\theta_k)\tau_k, \sin(\theta_k)\tau_k \quad (2.35)$$

where $\theta_k \sim \text{Uni}(0, \pi/2)$ and $\tau_k = (65/64)^k$. This base is chosen to be much more conservative than the $(9/8)^k$ heuristic used in Section 2.1.3, such that we will tend to avoid multimodality. The results of applying SMC to this model without any optimization are shown in Figure 2.13.

2.3.1.3 Rabi Model with Referencing

The third example of a simple Hamiltonian that we consider is that of a Rabi experiment on a spin- $1/2$ particle, with our model parameters including an unknown Rabi frequency ω_R , and an off-resonance term $\delta\omega$. For this example, we will consider a simple model for optically-detected magnetic resonance [141]. In particular, we shall suppose that a photon is emitted whenever the measurement result is 1 (that is, when we return to the initial state $|\psi\rangle = |0\rangle$), such that

$$\Pr(\text{emit}|\underline{x}; e) = |\langle 0|e^{-iH(\underline{x})t}|0\rangle|^2, \quad (2.36)$$

where

$$H(\underline{x}) = \omega_R\sigma_x + \delta\omega\sigma_z. \quad (2.37)$$

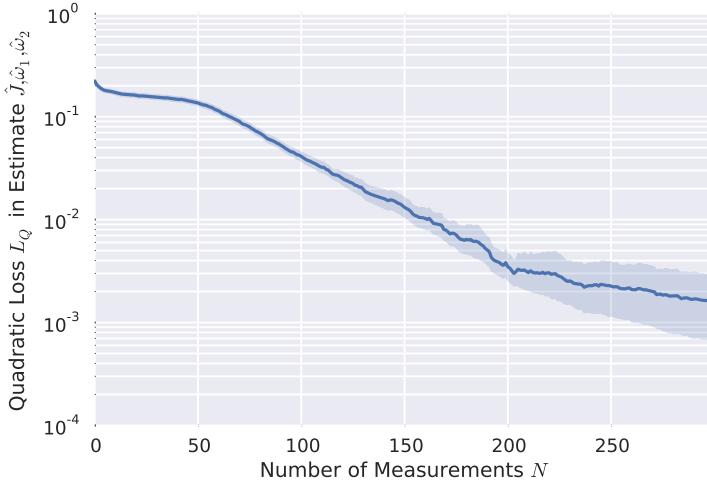


Figure 2.13: Mean quadratic loss and 99% confidence interval for the correlated precision model of (2.34) over 600 trials, versus the number of experiments N performed.

Moreover, in order to lay the groundwork for experiments such as those discussed in Section 2.3.3, we include *referencing* according to a simple model for photodetection

$$\begin{aligned}\alpha &:= \Pr(\text{click}|\text{emit}) \\ \beta &:= \Pr(\text{click}|\neg\text{emit}),\end{aligned}\tag{2.38}$$

where α and β are the two parameters describing bright and dark counts, respectively.

Effectively, this referencing model modifies another model, in the sense that it uses a likelihood function of the form $\Pr(\text{emit}|\underline{x}; \underline{e})$ to define a new likelihood function

$$\Pr(\text{click}|\underline{x}; \underline{e}) = \begin{cases} (\alpha - \beta) \Pr(\text{emit}|\underline{x}; \underline{e}) + \beta & \text{mode = experiment} \\ \alpha & \text{mode = bright reference,} \\ \beta & \text{mode = dark reference} \end{cases}\tag{2.39}$$

where α and β are taken to be elements of \underline{x} , and where mode is an element of \underline{e} . The model is implemented by Listing C.5.

In Figure 2.14, we show the results of applying this model to data obtained from an experiment in nitrogen vacancy centers [142]. We consider this application in more detail in Section 2.3.3. These results show that sequential Monte Carlo can be used in practice to learn Hamiltonian parameters from experimental data.

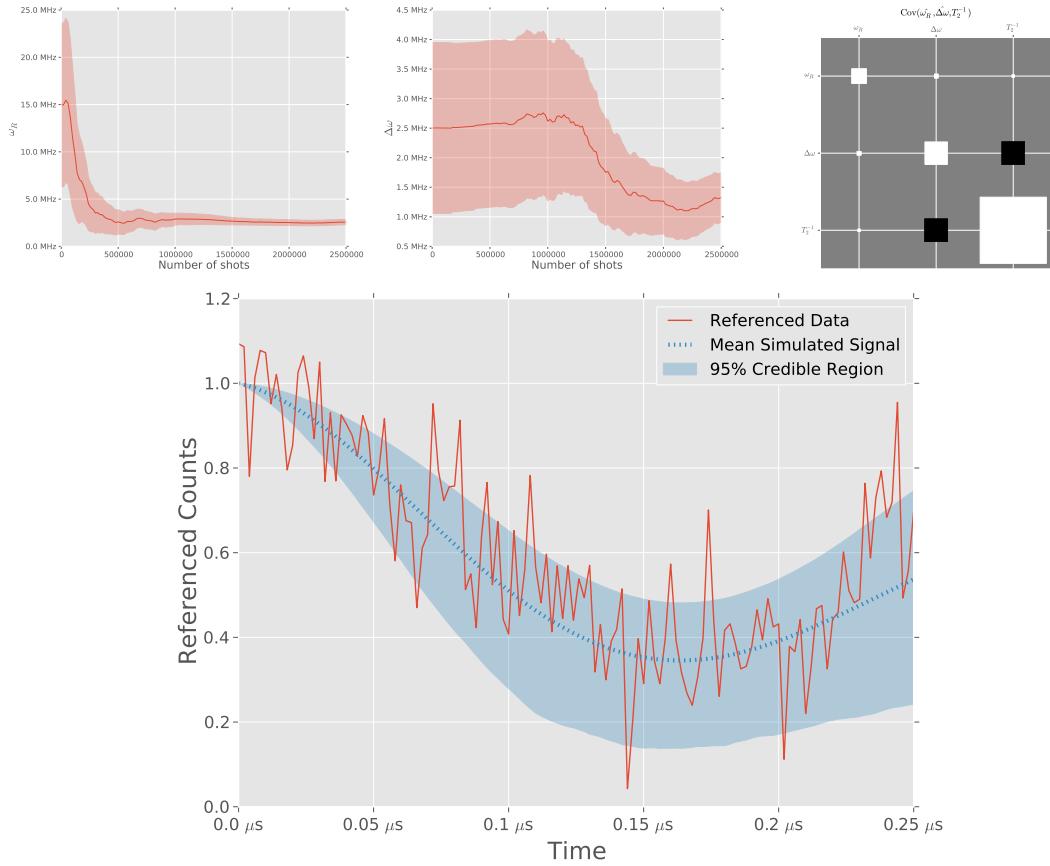


Figure 2.14: Experimental demonstration of referenced Rabi model (2.36). (Top left and top middle) Mean and standard deviation of posterior for ω_R and $\delta\omega$, respectively. (Top right) Covariance of final posterior, shown as a Hinton diagram. White squares indicate correlation, black squares indicate anticorrelation; the size of each square indicates magnitude. (Bottom) Referenced experimental data, shown with signals simulated using the SMC posterior. The shaded region indicates a 95% **credible region** on the mean of the referenced signal.

2.3.2 Randomized Benchmarking

In many cases, we are not interested in a full characterization of a quantum gate, but wish only to know its fidelity, perhaps averaged over some distribution. For instance, this is the target functional in many common control theory optimizations, such as those explored in [Chapter 5](#), such that a means of evaluating the fidelity of a gate or gateset will give a means of confirming the successful optimization and implementation of a pulse sequence. As shown by the Ad-HOC and ACRONYM algorithms [86; 25], evaluating the fidelity of a candidate pulse in-place is also useful in refining or finding new pulses.

Randomized benchmarking [143; 144; 145] gives a means of evaluating the fidelity of a gate or gateset without characterizing a gate explicitly by using *twirling* to reduce that gateset to one that can be simulated efficiently on a classical computer, and that has the same fidelity as the original gate or gateset. In particular, as illustrated in [Figure 2.15](#), in randomized benchmarking experiments we choose random sequences of **Clifford group** elements, each of which is implemented by a [word](#) of one or more generating gates [146; 147]. For example, the Clifford group \mathcal{C}_1 acting on a single qubit can be written as

$$\mathcal{C}_1/\mathcal{P}_1 = \langle H, S \rangle / \mathcal{P}_1 = \{ \mathbb{1}, HSH, HS\bar{H}S, S, H, HS \} / \mathcal{P}_1. \quad (2.40)$$

That is, each such Clifford operator can be written as the product of up to four gates from $\{H, S\}$ and either zero or one Pauli operators.

The Clifford group then forms a *2-design*; that is, for any polynomial $p(U)$ of degree at most 2, the integral over the Haar measure of $p(U)$ is given by uniformly sampling over the Clifford group [146]. In particular,

$$\int p(U) dU = \frac{1}{\mathcal{C}_n} \sum_{C \in \mathcal{C}_n} p(C_i). \quad (2.41)$$

Thus, sampling from the Clifford group implements the twirling superchannel W , which produces a depolarizing channel with the same probability of returning the initial state to itself [68]. [Figure 2.16](#) sketches the proof given by [145] that choosing sequences in this way implements twirling, even for imperfect implemented gates. Thus, this procedure results in an analytically tractable survival probability. Within this framework, the final gate is chosen by Gottesman-Knill simulation [63] to be the inverse of the preceding elements, such that the measurement being performed always has probability 1 of succeeding when implemented with ideal gates. Random sequences can be generated by choosing random Clifford group elements [148], decomposing them into gates from the gateset [149], and then applying each prescribed gate.

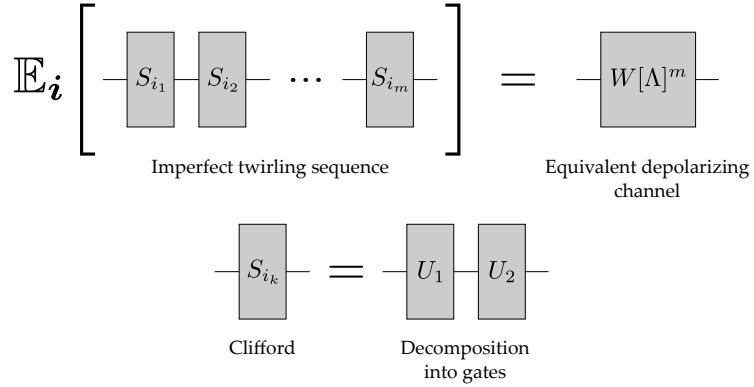


Figure 2.15: Overview of randomized benchmarking. Twirling is performed by sampling from the expectation value over all Clifford sequences (upper-left), producing a depolarizing channel of equivalent fidelity (upper-right). To implement each sequence, the Clifford unitaries are decomposed into words over a presentation of the Clifford group (bottom).

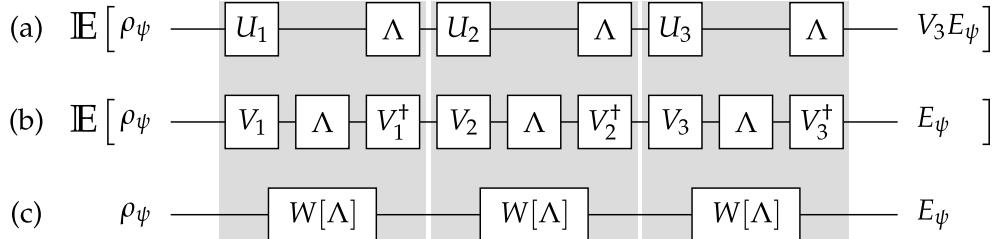


Figure 2.16: Sketch of Magesan et al derivation of zeroth-order model [145]. (a) Sequence of length $m = 3$ Clifford operations. (b) Change of variables to V_i , factoring out previous gates U_{i-1} . The V gates then form a 2-design. (c) Expectation value over random gates in (a) and (b), giving the twirling superchannel W acting on Λ .

Recent experiments have demonstrated the applicability of randomized benchmarking techniques across a wide variety of systems [150; 151; 152; 153; 154; 155; 156; 157]. Moreover, benchmarking has been extended to address a wide range of related problems, such as crosstalk characterization [158], identifying distortions [159] and spectral filtering of noise sources [160]. Thus, improvements to the robustness and costs of current randomized benchmarking techniques offer advantages to many different experimental goals.

In this Section, we will interpret the zeroth-order model for randomized benchmarking as a [likelihood function](#), and will apply sequential Monte Carlo to learn the average fidelity of the gateset with less experimental resources than are required with previous methods. This will be especially useful in cases where we want to draw accurate conclusions from small amounts of data. This is the case when randomized benchmarking is in the inner loop of a larger protocol, such as in the Ad-HOC or ACRONYM algorithms [85; 25].

The source code for all of the results in this Section is available online in the supplementary materials for [33], and is based on the QInfer [44] and QuaEC [45] libraries.

2.3.2.1 Sampling Variance and Derivation of Marginalized Likelihood

In this derivation, we will focus on the zeroth-order model of Magesan et al [145], which gives that the average fidelity $F_g(m)$ over all sequences of length m is given by

$$F_g(m) = A_0 p^m + B_0 \quad (2.42)$$

for constants A_0 and B_0 describing the state preparation and measurement (SPAM) errors, where $1 - p$ is the depolarizing strength of $W[\mathbb{E}_{C \sim C_n}[\hat{S}_C]]$, and where W is the superchannel [68] that maps every channel to a depolarizing channel with the same average gate fidelity.

We are interested in the single-shot limit, where each measurement consists of first selecting a sequence, then measuring once the survival probability for that sequence. Since this protocol makes no use of the sequence other than its length, we can describe the protocol by marginalizing over the choice of sequence, giving a probability distribution of the form $\Pr(\text{survival}|m)$, where m is a sequence length.

To derive this, we first pick a length m , and then consider the choice of sequence i out of all length- m sequences to be a random variate. Thus, there exist probabilities

$$p_{m,i} := \Pr(\text{survival}|i, m) = \text{Tr}(E_\psi \hat{S}_i[\rho_\psi]) \quad (2.43)$$

for each individual sequence that we could have chosen, such that marginalizing over results in

$$\Pr(\text{survival}|m) = \mathbb{E}_{\underline{i}}[\Pr(\text{survival}|\underline{i}, m)]. \quad (2.44)$$

If each sequence is drawn with uniform probability, then

$$\Pr(\text{survival}|m) = \frac{1}{|\mathcal{C}_n|^m} \sum_{\text{is.t. } \text{len } \underline{i} = m} p_{m,\underline{i}}. \quad (2.45)$$

We recognize this as being the average sequence fidelity $F_g(m)$ modeled by Magesan,

$$\Pr(\text{survival}|m) = F_g(m) = A_0 p^m + B_0. \quad (2.46)$$

To interpret $F_g(m)$ as a likelihood directly, note that we had to consider the Bernoulli trial (single-shot) limit; had we instead taken K distinct sequences and measured each $N > 1$ times, we would have arrived at a quite different quantity

$$\hat{F}_g(m) = \sum_{k=1}^K \hat{F}(m, i_k), \quad (2.47)$$

where $\hat{F}(m, i_k)$ is the estimate of the sequence fidelity for the *particular* sequence i_k .

The difference is made clear by considering an example with fixed sequence length m , and the variance for a datum $d \sim \Pr(\text{survival}|m)$ (labeling ``survival'' as 1 and the complementary event as 0),

$$\mathbb{V}_d[d|m] = \mathbb{V}_{\underline{i}}[\mathbb{E}_d[d|i, m]] + \mathbb{E}_{\underline{i}}[\mathbb{V}_d[d|i, m]]. \quad (2.48)$$

The second term corresponds to the mean variance over each fixed sequence i_m , and governs how well we can estimate each $F(m, i)$ individually. The first term, however, is more interesting, in that it measures the variance *over sequences* of the per-sequence survival probability $p_{m,i} = \mathbb{E}_d[d|i, m]$. By the argument of Wallman and Flammia [161], this is small when the fidelity being estimated is close to 1; that is, when the gates being benchmarked are very good. For gates that are farther from the ideal Clifford operators, however, or for applications such as tomography via benchmarking [162], this term is not negligible, mandating that many different sequences must be taken for $\hat{F}_g(m)$ to be a useful estimate of $F_g(m)$.

By demanding that each individual shot be drawn from an independently chosen sequence, our approach avoids this and samples from $d | m$ directly. In this way, we see a similar effect as in [Section 2.1.3](#). It is not advantageous to concentrate one's sampling on one point, but to spread samples out and gain experimental variety. Here, the one shot per sequence limit plays the role of the one sample per time-point limit in the earlier discussion.

2.3.2.2 Fisher Information and Optimal Sequence Lengths

We can also consider the Fisher information for the likelihood model (2.46) in order to find the optimal sequence length that one should draw sequences from. Given p , A_0 and B_0 , the score vector of (2.46) is given by

$$\underline{q} = \frac{(-1)^d}{\Pr(d|p, A_0, B_0; m)} (A_0 m p^{m-1}, p^m, 1), \quad (2.49)$$

where $d \in \{0, 1\}$ is a label for whether the initial state survived or not. To find Fisher informations in QInfer then consists of providing a `Model.score` method for the `RandomizedBenchmarkingModel` class, as demonstrated in Listing C.4. With this implementation, it is then easy to find the Cramér-Rao bound for different protocols. For instance, if our experimental procedure is to pick K sequences each of length $m \in \{1, 2, \dots, m_{\max}\}$ and then measure each sequence once, the Fisher information for this protocol is just the sum of the Fisher information matrices for a binomial over the K Bernoulli trials at each different m . This calculation is shown in Listing 2.1.

Listing 2.1: Calculation of achievable risk in a zeroth-order randomized benchmarking experiment.

```

def achievable_non_interleaved(m_max, K, p, A, B):
    ms = np.arange(m_max)
    model = DifferentiableBinomialModel(RandomizedBenchmarkingModel())

    5    expparams = np.empty(ms.shape, dtype=model.expparams_dtype)
        expparams['m'] = ms
        expparams['n_meas'] = K

    true_model = np.array([[p, A, B]])

    10   fi = np.sum(
            model.fisher_information(true_model, expparams),
            axis=-1
        )[:, :, 0]

    15   return np.linalg.sqrtm(np.linalg.inv(fi))[0, 0]

```

2.3.2.3 Interleaved Randomized Benchmarking

If we are interested not in the performance over a gateset, but in the fidelity of a particular gate, then we can use a similar protocol to that described above by *interleaving* the gate

of interest with other gates randomly chosen from our gateset [163]. In the interleaved protocol, random sequences of Clifford group elements are chosen to form a reference, in addition to sequences in which every even-index gate is chosen to be the gate of interest U . Here, we shall assume that $U \in \mathcal{C}_n$, such that the final gate can be designed to perform the desired measurement.

We call by p_{ref} the depolarizing parameter for sequences of random Cliffords, while \tilde{p} is the depolarizing parameter for $W[U]$. The likelihood function for this protocol now depends on whether we are drawing a reference or interleaved sequence, such that

$$\Pr(\text{survival}|m, \text{mode}; A_0, B_0, p_{\text{ref}}, \tilde{p}) = \begin{cases} A_0 p_{\text{ref}}^m + B_0 & \text{mode is reference} \\ A_0 (p_{\text{ref}} \tilde{p})^m + B_0 & \text{mode is interleaved} \end{cases}. \quad (2.50)$$

Note that in this definition, we have defined the length m of a sequence to be the number of random Clifford elements that we include, such that an interleaved sequence of length m has $2m + 1$ Clifford group elements, of which every even one is taken to be U .

One advantage of analyzing this model with SMC rather than least-squares fitting is that by doing so, we can explicitly demand that $\tilde{p} \in [0, 1]$, while with naïve implementations of LSF, \tilde{p} is estimated by $\hat{p}_{\text{interleaved}} / \hat{p}_{\text{ref}}$, which is intrinsically unstable and does not guarantee that only valid probabilities are estimated.

In the same manner as in [Section 2.3.2.2](#), we can also consider the Fisher information of this model in order to find experiments that produce maximal information as a function of the ``true'' model parameters. In [Listing 2.2](#), we show this calculation for the interleaved case, producing the results shown in [Figure 2.17](#).

[Listing 2.2](#): Calculation of m with optimal Fisher information for interleaved randomized benchmarking.

```

from scipy.optimize import minimize
def best_m(p_tilde, p_ref, A, B):
    objective = lambda m: (
        A**2 *m**2 *p_tilde**(-2 + 2*m)*p_ref**(2*m))/
        ((-1 + B + A*p_tilde**m *p_ref**m)*(B + A*p_tilde**m *p_ref**m))
    )
    result = minimize(objective,
                      100, method='nelder-mead'
                      )
return result.x

```

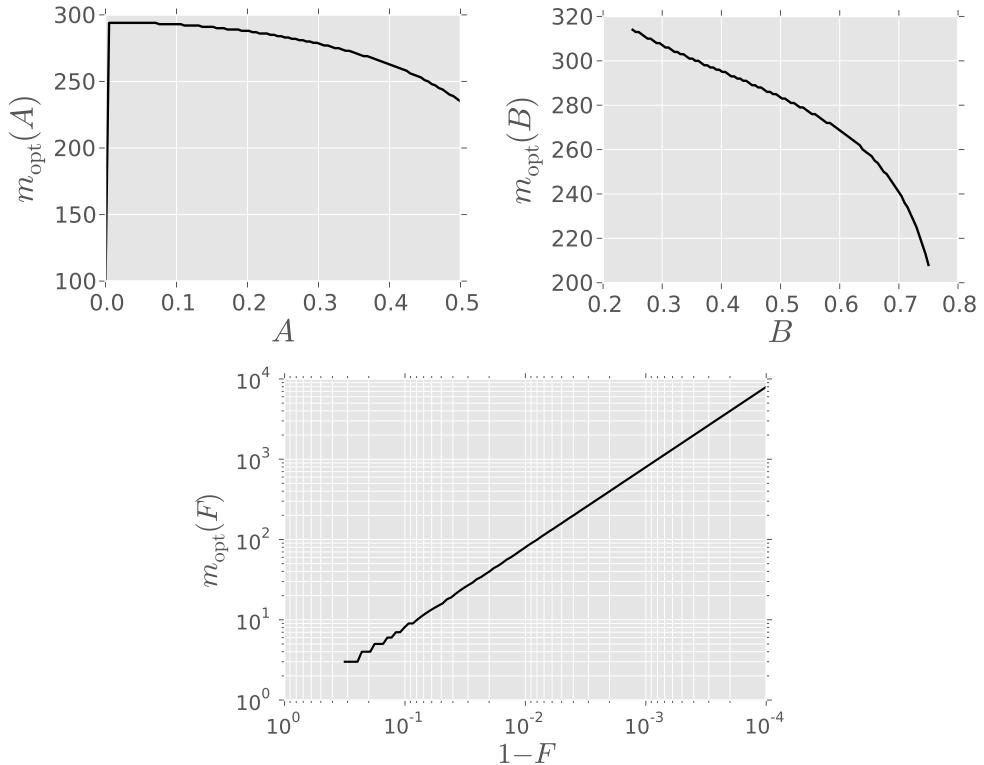


Figure 2.17: Optimal sequence length m according to the Fisher information as a function of zeroth-order randomized benchmarking parameters. The depolarizing parameters used correspond to the reference fidelity $F_{\text{ref}} = 0.9994$, and the fidelity of interest $F = 0.9989$. (Left) Varying over A , taking $B = 0.5$. (Right) Varying over B , taking $A = 0.25$. (Bottom) Limiting case for large systems, $A \rightarrow 1$, $B \rightarrow 0$, plotted versus infidelity of interest $1 - F$.

2.3.2.4 Results with Zeroth-Order Model

In order to test our implementation of SMC-accelerated randomized benchmarking, we draw many different “true” models from a prior $\pi(\underline{x})$, and then estimate each using SMC, such that we are sampling from the Bayes risk $r(\pi, \hat{x}_{\text{SMC}})$ and thus obtain a Monte Carlo estimate of the true Bayes risk. In particular, we use as our prior a multivariate normal distribution $\underline{x} = (p_{\text{ref}}, \tilde{p}, A, B) \sim N((0.95, 0.95, 0.3, 0.5), 0.01^2 \mathbb{1})$, truncated to lie within the region of valid models. The least-squares fit estimator is seeded with an initial guess drawn from this prior, so as to fairly compare the estimators. We then vary over the amount of data that SMC is allowed to use in estimating \underline{x} , then compare to the known-optimal Bayes risk given by the BCRB, the online-estimated risk given by the posterior trace-covariance, and to the performance of least-squares fitting.

The risk that we obtain from this procedure is shown in [Figure 2.18](#), demonstrating that SMC-accelerated randomized benchmarking gives marked advantage in the limit of small amounts of data, offering useful conclusions with as little as 150 bits drawn from the experiment. As the amount of data increases, such that the fidelity estimate at each individual sequence length m is approximately normal, we enter the regime in which least-squares fitting is more suited, such that it begins to catch up with the SMC-accelerated Bayes risk. Moreover, in all cases, the performance of SMC-accelerated RB cannot be practically distinguished from the optimal performance given by the BCRB. This indicates that by relaxing the demand made by least-squares fitting that each individual point in the randomized benchmarking signals be well-estimated, we can obtain a very significant advantage, in analogy to that gained by relaxing the same requirement in [Section 2.1.3](#).

2.3.2.5 Results with Physical Gates

Thus far in the analysis, we have used as a simulator the same zeroth-order model as is used to process and interpret the data. To demonstrate the utility of our approach in comparison with traditional LSF-based benchmarking, we now simulate gates according to a cumulant expansion ([Section 1.5.3.4](#)), with physically realistic models.

In particular, we use the superconducting model of [164] together with optimal control theory [84] ([Chapter 5](#)) to generate a set of gates implementing the target unitaries $\{\mathbb{1}, X, Y, Z, H, P\}$, where H is the Hadamard gate, and where $P = |0\rangle\langle 0| + i|1\rangle\langle 1|$ is the phase gate. We then use the superoperators \hat{S}_U for implementing each target unitary U obtained from a cumulant simulation [80; 81] to sample from the likelihood function ([2.44](#)).

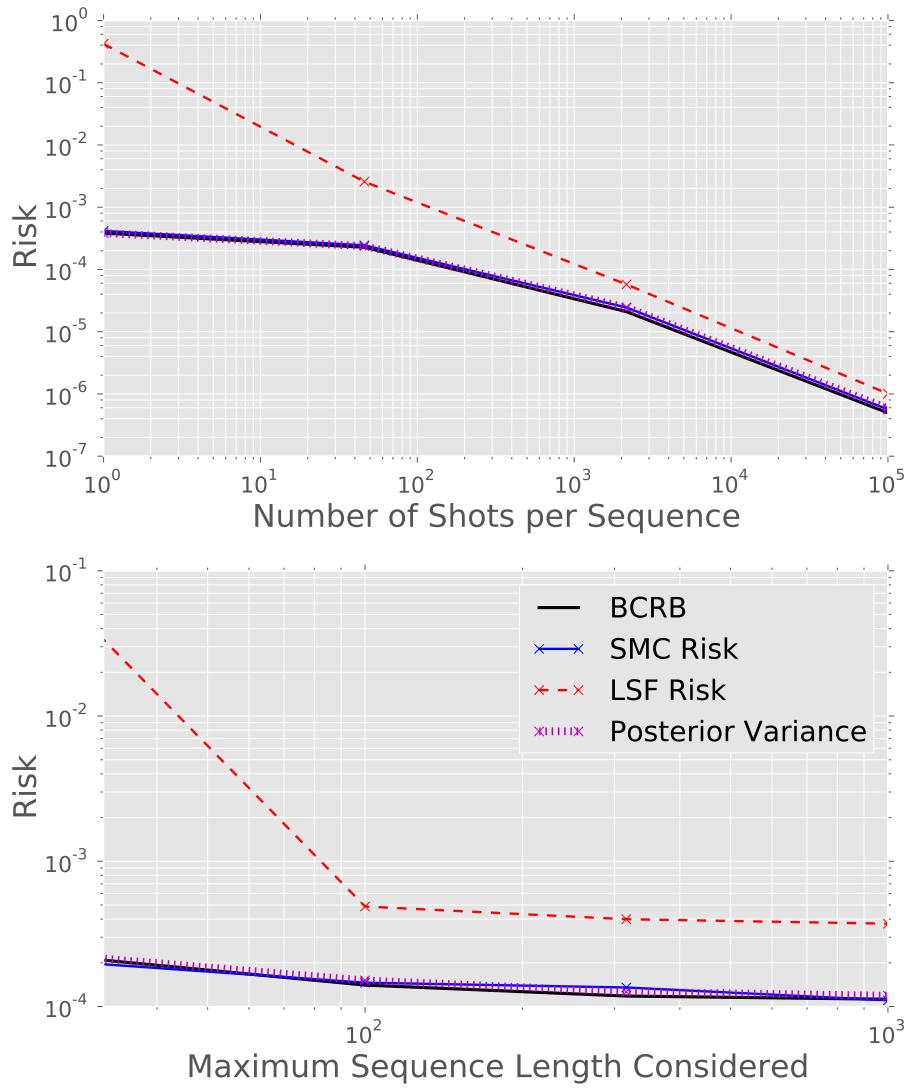


Figure 2.18: Bayes risk of SMC-accelerated randomized benchmarking, compared to the Bayesian Cramér-Rao bound, the posterior covariance and Bayes risk incurred by least-squares fitting. (Top) The Bayes risk is shown versus the number of shots K per sequence length. Data is collected from reference sequences of length $m_{\text{ref}} \in \{1, \dots, 100\}$ and interleaved sequences of length $m_{\text{interleaved}} \in \{1, \dots, 50\}$. (Bottom) The Bayes risk is shown versus the maximum sequence length m_{max} used, with sequences drawn from lengths $m \in \{1, 11, \dots, m_{\text{max}}\}$ for both the reference and interleaved cases, and with $K = 1000$ shots drawn from each length.

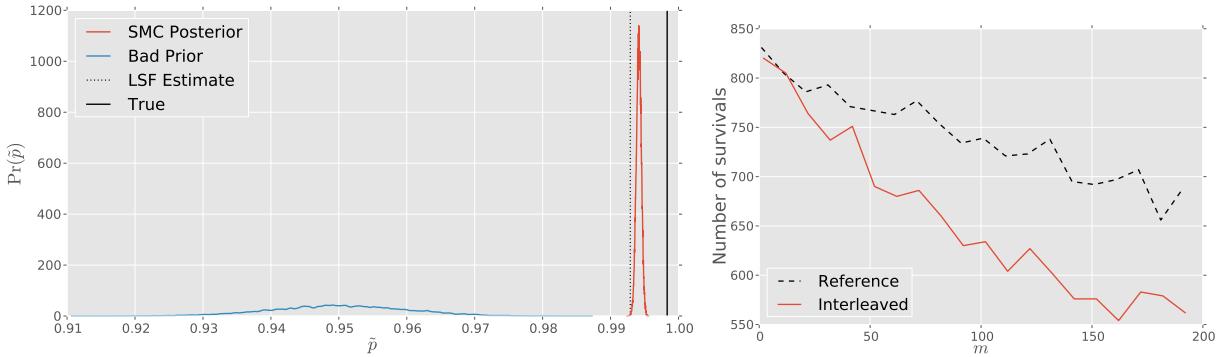


Figure 2.19: (Left) Comparison of prior distribution, SMC-approximated posterior, true value and LSF-estimate for \tilde{p} for a single run with $K = 1000$ shots at each of $m_{\text{ref}} \in \{1, 11, \dots, 191\}$ and $m_C \in \{2, 12, \dots, 192\}$. An intentionally inaccurate prior is used, such that the true value is approximately 6.9 standard deviations from the mean of the prior. As shown in Table 2.1, SMC does well by comparison to LSF, even with the poorly-chosen prior. (Right) Data gathered from simulation with physical-model gates.

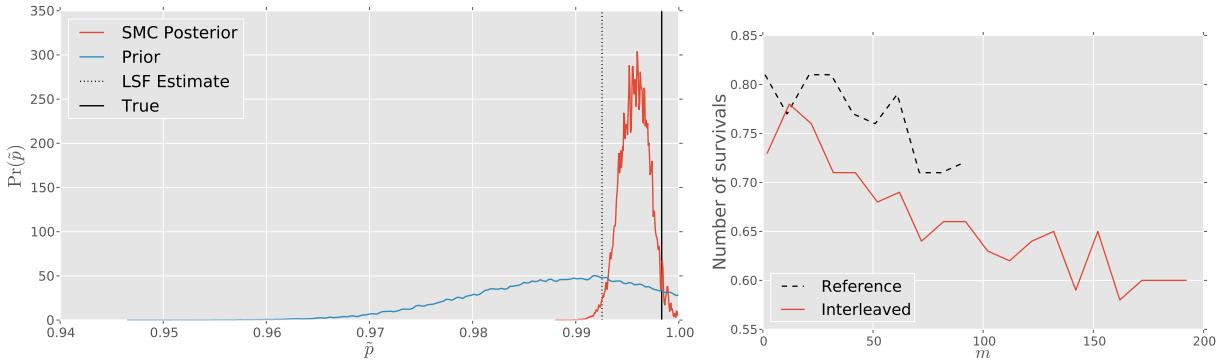


Figure 2.20: (Left) Comparison of prior distribution, SMC-approximated posterior, true value and LSF-estimate for \tilde{p} for a single run with $K = 100$ shots at each of $m_{\text{ref}} \in \{1, 11, \dots, 91\}$ and $m_C \in \{2, 12, \dots, 192\}$. (Right) Data gathered from simulation with physical-model gates.

Table 2.1: Results of using SMC and least-squares fitting to estimate the fidelity of $U = X$, simulated using the superconducting qubit gate set. (Left) Bad prior from [Figure 2.19](#), (right) accurate prior from [Figure 2.20](#).

	Bad Prior (40×10^3 bits)				Good Prior (3×10^3 bits)			
	\tilde{p}	p_{ref}	A_0	B_0	\tilde{p}	p_{ref}	A_0	B_0
True	0.9983	0.9957	0.3185	0.5012	0.9983	0.9957	0.3185	0.5012
SMC Estimate	0.9942	0.9971	0.3023	0.5075	0.9957	0.9969	0.2973	0.5010
LSF Estimate	0.9929	0.9974	0.3423	0.4827	0.9925	0.9986	0.5153	0.2782
SMC Error	0.0042	0.0014	0.0161	0.0062	0.0026	0.0011	0.0212	0.0003
LSF Error	0.0054	0.0017	0.0239	0.0185	0.0058	0.0029	0.1968	0.2230

To process these samples, we then use the zeroth-order likelihood function [\(2.50\)](#) both as a model for sequential Monte Carlo and as a trial function for least-squares fitting. Since the actual implemented gates are known, we can compute the true parameters for comparison. In [Table 2.1](#), we show the true parameters, the result obtained using SMC, and the result obtained using least-squares fitting. The most important thing to note is that correct parameters are a distance 6.90σ from the prior (meaning the true parameters are outside of the 99.999998% credible ellipse). This shows that even in the case when the prior information fails to accurately capture the uncertainty in the true model, SMC still does well, providing evidence that our accelerated methods may also be *robust*, even when used to measure the fidelities of sets of gates with errors that are correlated between distinct gate types, or that include non-trivial unitary components ². We show this in more detail in [Figure 2.19](#), comparing the posterior and prior distributions over \tilde{p} to the true and LSF-estimated values.

Finally, in [Figure 2.20](#), we demonstrate the advantage of our method in the presence of physical gates together with a more reasonable prior, and using approximately 10-fold less data than in [Figure 2.19](#). Taken with other evidence of the robustness of SMC methods [\[131; 165\]](#), these results thus show that our method is useful and provides advantages in data collection costs in experimentally-reasonable conditions.

²Note that SMC did not act in a robust manner in all cases observed, but in those cases where SMC did not do well by comparison to LSF, the QInfer package was often able to warn by using the effective sample size criterion described in [\[32\]](#), such that the data processing could then be repeated if necessary, or such that a more appropriate prior could be chosen. Model selection ([Section 3.2](#)) offers a more formal way of ensuring protection against a bad prior.

2.3.2.6 QInfer Implementation

In [Listing 2.3](#), we show an example of how experimental data might be processed, using the SMC acceleration for randomized benchmarking built into QInfer [\[44\]](#).

[Listing 2.3](#): Processing randomized benchmarking data with QInfer.

```
import numpy as np

from qinfer.rb import BinomialModel, RandomizedBenchmarkingModel
from qinfer.distributions import (
    MultivariateNormalDistribution, PostselectedDistribution
)
from qinfer.smc import SMCUpdater

# Initialize a zeroth-order benchmarking model, using a binomial
# to count over repetitions of each sequence.
model = BinomialModel(RandomizedBenchmarkingModel())

# Set the prior to be a multivariate normal,
# postselected such that only valid samples are used.
prior = PostselectedDistribution(MultivariateNormalDistribution(
    mean=np.array([0.99, 0.99, 0.3, 0.5]),
    cov=np.diag([0.01, 0.01, 0.01, 0.01])**2
), model)

# Make an updater with 10000 particles using this prior and model.
updater = SMCUpdater(model, 10000, prior)

# As data arrives...
n_shots_per_seq = 20
for (datum, seq_length) in data:
    expparams = np.array(
        [(seq_length, n_shots_per_seq)],
        dtype=model.expparams_dtype
    )
    updater.update(datum, m)
```

2.3.3 Nitrogen Vacancy Centers

SMC is particularly useful in the case of nitrogen-vacancy (NV) centers addressed using optically-detected magnetic resonance (ODMR), where a carbon atom in a diamond lattice is replaced by a nitrogen and an adjacent vacancy, as illustrated in [Figure 2.21](#). NV

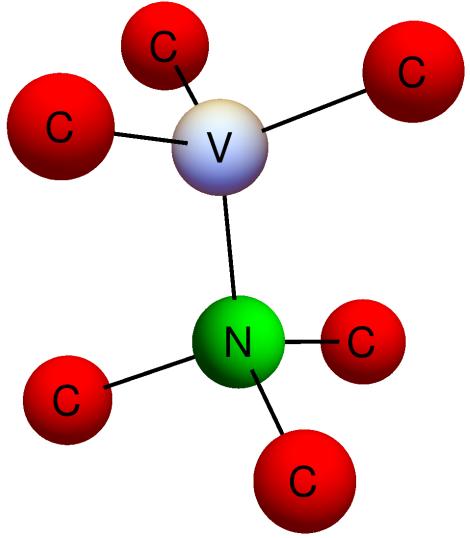


Figure 2.21: Nitrogen vacancy in a diamond lattice. A nitrogen defect (green) adjacent to a vacancy (pearl) is located within a carbon lattice (red). By convention, the \hat{z} -axis is taken to be parallel to the NV axis connecting the nitrogen and vacancy lattice sites.

centers have been proposed as a modality for quantum information processing [166; 23]. Nitrogen vacancy centers have also been used for metrology, and in particular have been used as scalar [167; 168], time-resolved [169], and vector magnetometers [170].

Owing to the wide range of applications for nitrogen vacancy centers, we are interested in developing better coherent control in diamond by improving our knowledge of the NV Hamiltonian. Moreover, because quantities such as the magnetic field are manifested in the Hamiltonian, improvements in Hamiltonian learning directly translate into metrological advances.

With our experimental setup, we collect either zero or one photons from each experiment with very high probability, such that we extract one bit of data from each measurement, putting us well into the regime where SMC is most useful. Following the insights

developed in [Section 2.1.3](#) and [Section 2.3.2](#), we expect to derive significant advantage by taking less data at each distinct experiment, in exchange for a wider variety of experiments. In [Section 2.3.3.2](#), we will demonstrate this by using an exponentially-sparse sampling heuristic together with SMC to calibrate the amplifier used in our experiments.

The nitrogen vacancy center NV^- in diamond admits a spin-1 degree of freedom, $m_s \in \{0, \pm 1\}$, in the ground state of its electronic configuration. This spin degree of freedom has a large zero-field splitting $\Delta_{\text{zfs}} \approx 2.87\text{GHz}$ between the $m_s = 0$ and $m_s = \pm 1$ states. The $m_s = \pm 1$ states are then further split by a Zeeman interaction, giving a qutrit that can be manipulated using microwave pulses at carrier frequencies near the zero-field splitting. Measurement is implemented by applying a pumping laser, and counting the photons emitted due to state-dependent spontaneous emission rates [[171](#); [172](#)], as illustrated in [Figure 2.22](#).

We use a confocal microscope to excite and polarize the center, and to collect light emitted from the center [[173](#)]. This gives us a small visibility that we include in our description by referencing data in a similar fashion as described in [Section 2.3.1.3](#).

2.3.3.1 NV Center Hamiltonian and Floquet Theory

In the lab frame, the Hamiltonian of an NV center with electronic spin \underline{S} coupled to a single carbon with spin \underline{I} is

$$H = (\Delta_{\text{zfs}} + \delta\Delta_{\text{zfs}})S_z^2 + \gamma_e(\underline{B} + \underline{\delta B}) \cdot \underline{S} + \gamma_e B_1(t) \cdot S_x + \gamma_n(\underline{B} + \underline{\delta B}) \cdot \underline{I} + \underline{S} \cdot \underline{\underline{A}} \cdot \underline{I}, \quad (2.51)$$

where Δ_{zfs} is the zero-field splitting in the ground-state manifold, \underline{B} is the static magnetic field, B_1 is a Rabi drive along the x -axis, γ_e and γ_n are the gyromagnetic ratios of the electronic and nuclear spins, respectively, $\underline{\underline{A}}$ is the hyperfine tensor between the electron and nuclear spins, and where $\delta\Delta_{\text{zfs}}$ and $\underline{\delta B}$ are the uncertainties in the respective parameters.

We must include the uncertainties in Δ_{zfs} and \underline{B} in order to accurately simulate the data that we obtain from experiments. The zero-field splitting depends on strain of the crystal, and thus may not be known precisely, while the static field is generated, for example, by a permanent magnet that we position with finite accuracy.

Once we have the Hamiltonian in the lab frame, we enter the rotating frame of the microwave pulses, and obtain a Floquet-Magnus stroboscopic Hamiltonian as described in [Section 1.5.3.4](#). For example, using the `NVSim`` package [[174](#)] to perform this calculation, we find that the average Hamiltonian for the case when drive Hamiltonian is on

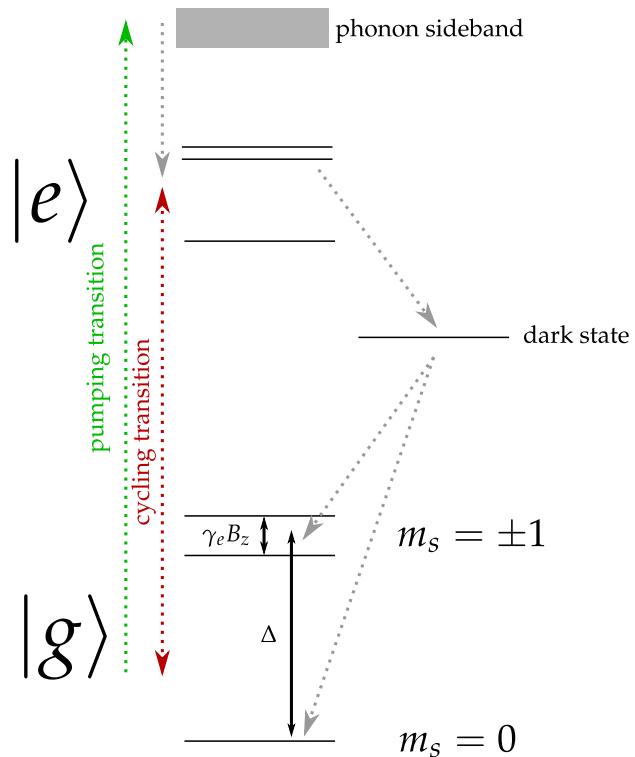


Figure 2.22: Schematic of the energy levels of a nitrogen vacancy center, showing pumping laser and state-dependent spontaneous emissions between the ground state $|g\rangle$ and excited $|e\rangle$ manifolds. The zero-field $\Delta_{\text{zfs}} S_z^2$ and Zeeman $\gamma_e B_z S_z$ splittings are illustrated in the ground-state manifold.

resonance with the zero-field splitting is given to zeroth-order by

$$\bar{H} = A_{xz}S_zI_x + A_{zz}S_zI_z + (B_z + \delta B_z)\gamma_e S_z + \frac{1}{2}\gamma_n(\underline{B} + \underline{\delta B}) \cdot \underline{I} \quad (2.52)$$

where

$$\underline{\underline{A}} = \begin{pmatrix} A_{xx} & 0 & A_{xz} \\ 0 & A_{yy} & 0 \\ A_{xz} & 0 & A_{zz} \end{pmatrix}. \quad (2.53)$$

Note that we can write $\underline{\underline{A}}$ in this form by an appropriate choice of the coordinate system, such that the representation of \underline{B} depends implicitly on the assumed form of $\underline{\underline{A}}$.

In practice, when we need to consider higher orders, we will perform simulations by directly implementing the Leskes formulas on the Fourier components of the rotating-frame Hamiltonian $\tilde{H}(t) = e^{+iH_{\text{rot}}t} \bullet H - H_{\text{rot}}$. Python code implementing the second-order Floquet-Magnus stroboscopic Hamiltonian in the off-resonant case is shown in [Listing C.6](#).

To include decoherence, we represent the Hamiltonian evolution H by a Liouvillian \hat{L} and add to it a Lindblad-form dissipator \hat{D} with a single operator S_z . This gives the generator

$$\hat{G} = -i\hat{L} + \frac{1}{T_2} \left[S_z^T \otimes S_z - \frac{1}{2}(S_z^2)^T \otimes \mathbb{1} - \frac{1}{2}\mathbb{1} \otimes S_z^2 \right]. \quad (2.54)$$

Finally, we must also add to [\(2.51\)](#) the hyperfine coupling to the spin-1 degree of freedom of the nitrogen adjacent to the vacancy. Since we cannot generally directly pulse on the nitrogen, it is convenient to represent this interaction as an effective field A_N along the NV axis \hat{z} . Moreover, at room temperature, and at the static fields that we are interested in working at, the nitrogen starts in a thermal state that we can approximate by the maximally mixed state $\rho_N = \mathbb{1}/3$. Thus, we can include the effect of this coupling by marginalizing the signal over the three basis states of the nitrogen spin, $a \in \{-1, 0, 1\}$. For instance, evolving under the internal Hamiltonian with no microwave pulses gives

$$\Pr(1|\underline{x}; t) = \mathbb{E}_a \left[\langle \langle \rho_0 | e^{-i\hat{L}(\underline{x}, a)t_p + \hat{D}(\underline{x})t} | \rho_0 \rangle \rangle \right]. \quad (2.55)$$

2.3.3.2 Power Calibration with Rabi Experiments

As a straightforward example of learning Hamiltonians in nitrogen vacancy centers, we revisit the Rabi experiment of [Section 2.3.1.3](#), replacing the likelihood function with an

explicit evaluation of the effective Hamiltonian

$$H = \Delta\omega_c S_z^2 + (\omega_e + aA_N)S_z + \Omega S_x, \quad (2.56)$$

where $\Delta\omega_c = (\omega_c - \Delta_{\text{zfs}})$ is the difference in the center frequency between the Rabi drive and the zero-field splitting, where ω_e is the Zeeman coupling, $a \in \{-1, 0, +1\}$ is the state of the nitrogen, A_N is the nitrogen hyperfine coupling strength, and Ω is the Rabi power. As in the previous section, we consider a dissipative generator and an incoherent sum over the nitrogen spin state, giving a likelihood function before referencing of

$$\Pr(1|\Omega, \omega_e, \Delta\omega_c, A_N, T_2^{-1}; t_p) = \mathbb{E}_a \left[\langle\langle \rho_0 | e^{-i\hat{L}t_p + \hat{D}t_p} | \rho_0 \rangle\rangle \right], \quad (2.57)$$

where t_p is the length of the Rabi pulse, $\rho_0 = |0\rangle\langle 0|$, $\hat{L} = \mathbb{1} \otimes H - H^T \otimes \mathbb{1}$ and \hat{D} is the dissipator for the Lindblad operator $T_2^{-1}S_z$.

Taking as our model parameters $x = (\Omega, \omega_e, \Delta\omega_c, A_N, T_2^{-1})$, we can now find the Rabi power Ω with few assumptions, and with a small number of measurements. In particular, we will use the sampling heuristic

$$t_p = \begin{cases} 10 \text{ ns} + k \times 12 \text{ ns} & k \leq 30 \\ 1.13^k \times 450 \text{ ns} & k > 30 \end{cases}, \quad (2.58)$$

for the k th datum. At each sample, 50,000 shots are collected, and the process is repeated 3 times for a total of 7,500,000 shots, taking approximately 5 minutes of experimental time.

The Rabi power, however, is controllable by varying the output level of the digital-to-analog converter (DAC) that generates pulses for the NV center experiment. The output of the DAC is passed into an amplifier, such that for an imperfectly calibrated amplifier, Ω is an unknown function of the DAC output level. By using that estimating with SMC is able to estimate Ω significantly more efficiently than traditional methods, we can more easily vary input powers and phases to learn the unknown calibration function. In [Figure 2.23](#), we show an example of learning the Rabi power as a function of both output phase and amplitude, along with a 68% credible interval for the estimated powers.

2.3.3.3 Hyperfine Estimation

We can also apply Bayesian inference with SMC to learn other properties of the nitrogen vacancy center Hamiltonian, such as hyperfine couplings. Here, we consider the coupling A_N between the electronic degree of freedom of the center itself, and the nuclear

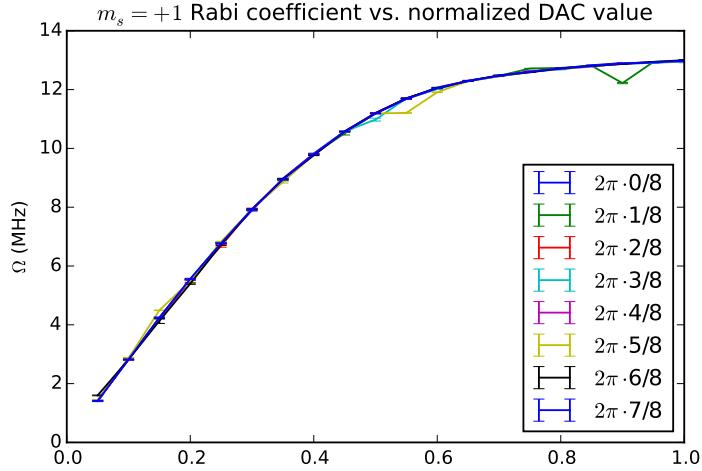


Figure 2.23: Estimated Rabi power curves as a function of normalized DAC output amplitude and phase, along with a 68% credible interval on each estimate.

spin of the nitrogen. In doing so, we will select an NV center with no nearby carbon-13 nuclei, such that $\underline{A} = 0$. We then identify as our model $\underline{x} = (\delta B, \delta \Delta_{\text{zfs}}, A_N, T_{2,e}^{-1}, \delta \omega_{\text{Rabi}})$, where $\delta \Delta_{\text{zfs}}$ is our uncertainty in the zero-field splitting, $\delta \omega_{\text{Rabi}}$ is our uncertainty in the Rabi frequency and where $T_{2,e}^{-1}$ is the dephasing time for the electronic degree of freedom.

In this experiment, for each average, we performed 30,000 shots at each of 200 Ramsey points and 100 Rabi points. In total, 50 such averages were performed, for approximately 450 million measurements. The SMC estimator is calculated from a subset of one-sixth of the total data collected (75 million shots), chosen at random. We show the results of this experiment in Figure 2.24, including a comparison of the actual data, and the signal simulated from the SMC estimate.

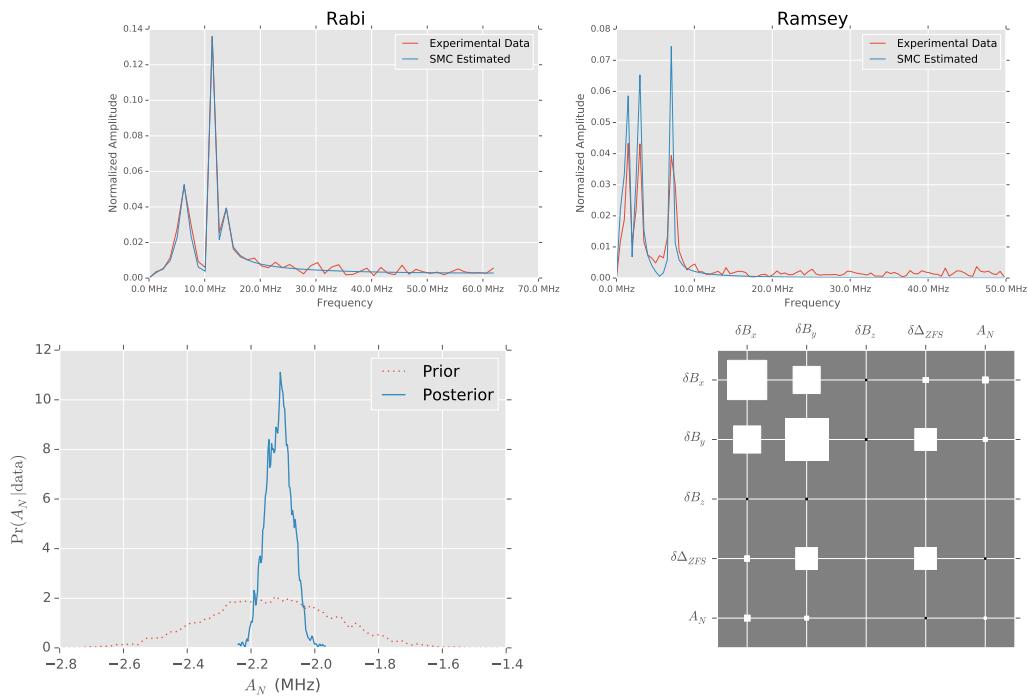


Figure 2.24: Simulation of Rabi and Ramsey experiments for hyperfine estimation in nitrogen vacancy centers. (Top) Comparison of simulated data according to SMC estimate and experimental data record. (Bottom left) Comparison of prior and posterior marginals over A_N . (Bottom right) Covariance of final posterior.

3 Verification of Quantum Systems

In addition to characterizing and controlling quantum systems, in producing useful quantum devices, we must be able to assert that the device under consideration is *correctly* implemented. This is, for instance, an existential problem in the field of quantum simulation [26]. Owing to the classical intractability of quantum simulation problems, no *a priori* answer can be used as a point of comparison.

Here, we address quantum verification by using classical simulation together with the sequential Monte Carlo algorithm, laying the groundwork for the inclusion of quantum resources in [Chapter 6](#). Because the SMC algorithm gives us an approximation of the complete posterior, we can leverage this information for many different aspects of verification, as we detail here.

In [Chapter 4](#), we will consider the problem of verification in a different guise by using *honest approximations* to reason about error correction and fault-tolerance.

3.1 Region Estimation

In the course of verifying quantum systems, we are commonly interested not just in a single [estimate](#) of the characterization of a system, but in learning some range in which

the true parameterization of the system is probably located. For instance, in [Section 5.2.3](#), we will show how to design control sequences that are effective over a range--- starting with a range that contains the true system with high probability thus allows us to control a system in a way that accounts for uncertainty in characterization.

Perhaps more importantly, however, is that by learning a region for the system, we have a probabilistic bound on how far the dynamics of a quantum system are from the claimed or nominal dynamics of the system, conditioned on the correctness of some model. Combined with model selection ([Section 3.2](#)) or model averaged estimation [[36](#)], this then gives us a way to *verify* the dynamics of a quantum system, and to assert that with some probability, a proposed model is correct.

3.1.1 Confidence and Credible Regions

There are two primary approaches to region estimation, confidence [[175](#)] and credible regions [[176](#)], respectively corresponding roughly to frequentist and Bayesian perspectives. Under the confidence region approach, a region $\hat{X}_\alpha(D)$ is sought such that

$$\forall \underline{x}_0 : \Pr(\underline{x}_0 \in \hat{X}_\alpha(D) | \underline{x} = \underline{x}_0) \geq \alpha; \quad (3.1)$$

that is, such that for any hypothetical model \underline{x}_0 , the confidence region estimator $X_\alpha(\cdot)$ includes \underline{x}_0 with probability no less than α , over all data we could obtain according to that hypothesis. Importantly, this does *not* state that given a specific dataset D , $\hat{X}_\alpha(D)$ contains the true model with some probability, nor does it make any statement about a particular confidence region estimate. Rather, confidence regions are concerned with developing procedures which will, for a percentage of experimental runs, find a correct estimate. For a specific data record and a specific region estimate, no statement can be made without inverting the conditional on the true model \underline{x}_0 . The confidence region definition can be a very useful statement, and provides quite strong guarantees, but it is not always the statement we wish to have. In particular, confidence regions do not give us the verification argument that we wish to formulate here. Indeed, confidence regions can be subtle to interpret, and can offer the same potential for misunderstanding as is seen with p -value hypothesis testing [[177](#); [178](#); [179](#); [180](#)]. In particular, neither confidence regions nor p -values inform as to the probability of a given hypothesis being correct, but instead describe how consistently a hypothesis will produce similar data when an experiment is repeated many times.

As an alternative, we consider the Bayesian approach of credible regions. By contrast with confidence regions, a region \hat{X}_α is α -credible if

$$\Pr(\underline{x} \in \hat{X}_\alpha(D) | D) \geq \alpha. \quad (3.2)$$

This definition is directly concerned with what we know given a particular dataset D , and thus can be interpreted as bounding below the probability that the true model can be found within $\hat{X}_\alpha(D)$. Importantly, credible regions can be formulated by using the posterior distribution to demand that

$$\int_{\hat{X}_\alpha(D)} d\Pr(\underline{x}|D) \geq \alpha. \quad (3.3)$$

3.1.2 Region Estimation with SMC

Within the SMC approximation, this is easy to meet by choosing particles $\{i_1, i_2, \dots, i_K\}$ such that $\sum_{k=1}^K w_{i_k} \geq \alpha$, but this is not a particularly nice or useful region, as it is a union of individual points within the model parameter space. Rather, we would like to find regions which contain these particles, but have other desirable properties. In particular, we would like for our credible regions to be of high density, and to admit short descriptions. Here, we develop the approach of using normal approximations and covariance ellipses to provide these features, but other approaches can be employed as well. Recent work by Ferrie [35] extends this development to use convex hulls and minimum-volume enclosing ellipses of SMC posteriors to offer short-description high-density credible regions for more general distributions.

For approximately normal posteriors, we can use covariance ellipses to find region estimates that have short descriptions ($(\dim \underline{x})^2$ parameters), and that approximately satisfy (3.2). In particular, for a multivariate normal distribution $N(0, \underline{\Sigma})$, the ellipse given by the matrix $\underline{\Sigma}^{-1}$ contains approximately $0.682^{\dim \underline{x}}$ of the posterior mass¹. More generally, if \underline{x} is normally distributed, then the covariance ellipse

$$\hat{X}_\alpha(D) = \left\{ \underline{x} \mid (\underline{x} - \mathbb{E}[\underline{x}|D])^\top \text{Cov}^{-1}(\underline{x}|D)(\underline{x} - \mathbb{E}[\underline{x}|D]) \leq Z^2 \right\} \quad (3.4)$$

contains the true model with probability $\alpha(Z) = \text{erf}(Z/\sqrt{2})^{\dim \underline{x}}$ [32].

In Figure 3.1, we show the true probability mass \tilde{m} of a covariance estimator for the Larmor precession model (2.11), compared with the corresponding probability mass m_{normal} that we would expect for a normal distribution.

¹This follows immediately from the CDF of the multivariate normal distribution, $\underline{\Sigma}^{-1}$ describes a covariance ellipsoid transformed from the multivariate standard normal, with z-score of 1 along each random variable.

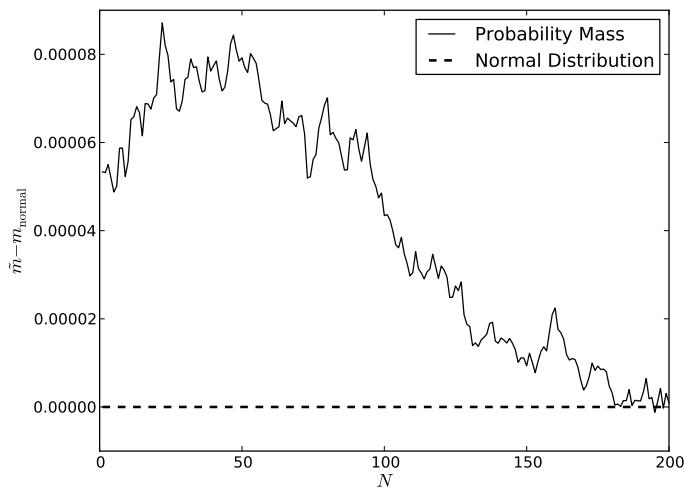


Figure 3.1: Probability mass (credibility) approximated by a covariance ellipse of the SMC posterior for the known- T_2 Larmor precession model, compared to the credibility for a true normal distribution, $m_{\text{normal}} \approx 0.9973$. The credibility is averaged over 20,000 trials, each of which is performed using 30 guesses around the $t_k = (9/8)^k$ heuristic, local optimization using the Newton Conjugate-Gradient solver and 1,000 SMC particles.

3.1.3 Hyperparameter Region Estimation

In [Section 2.2.4](#), we showed how **hyperparameters** can be used to enable estimation of a model whose parameters are themselves sampled from a distribution with each experiment. Taking the approach of the previous section directly in this case is not especially useful, however, in that a region on hyperparameter space is almost certainly not what we are most interested in, so much as finding a region such that the underlying **model parameters** are contained at some credibility α , including contributions both from uncertainty about the hyperparameters \underline{y} and the covariance of the model parameters \underline{x} conditioned on a hypothesis about \underline{y} .

In the context of covariance ellipse credible regions such as [\(3.4\)](#), this can be readily accomplished by using the chain rule for variance,

$$\text{Cov}_{\underline{x}, \underline{y}}(\underline{x}) = \mathbb{E}_{\underline{y}}[\text{Cov}(\underline{x}|\underline{y})] + \text{Cov}_{\underline{y}}(\mathbb{E}_{\underline{x}}[\underline{x}|\underline{y}]). \quad (3.5)$$

The first term, $\mathbb{E}_{\underline{y}}[\text{Cov}(\underline{x}|\underline{y})]$, describes the uncertainty we expect to be inherent to the hyperparameter model under consideration, while the second term, $\text{Cov}_{\underline{y}}(\mathbb{E}_{\underline{x}}[\underline{x}|\underline{y}])$ describes the uncertainty added to \underline{x} because of uncertainty about \underline{y} . Both can be readily calculated for special cases of interest, such as $\underline{x}|\underline{y}$ being a multivariate normal. More generally, Monte Carlo sampling can be used.

Once we have the covariance over \underline{x} without conditioning on a particular hypothesis, we can apply the same argument as in [Section 3.1.2](#) to treat it as an ellipsoidal region estimate. In particular, we can condition [\(3.5\)](#) on the data we've obtained, using that $\underline{x} \perp D | \underline{y}$ to remove extraneous conditionals. Doing so, we obtain that

$$\text{Cov}_{\underline{x}, \underline{y}}(\underline{x}|D) = \mathbb{E}_{\underline{y}}[\text{Cov}(\underline{x}|\underline{y})|D] + \text{Cov}_{\underline{y}}(\mathbb{E}_{\underline{x}}[\underline{x}|\underline{y}]|D). \quad (3.6)$$

A very desirable property for region estimators derived from hyperparameter posterior covariance ellipses would be for the covariance of \underline{x} to be an overestimate of the covariance of \underline{x} conditioned on the true model. In the single parameter case, this is easy to express as the demand that $\mathbb{V}[\underline{x}] \geq \mathbb{V}[\underline{x}|\underline{y} = \underline{y}_0]$. In [Figure 3.2](#), we plot the excess covariance in \underline{x} , $\mathbb{V}[\underline{x}] - \mathbb{V}[\underline{x}|\underline{y} = \underline{y}_0]$ for the hypernormal Larmor model [\(2.29\)](#) and show that, over a large number of runs, it remains positive, indicating an **honest** region estimate.

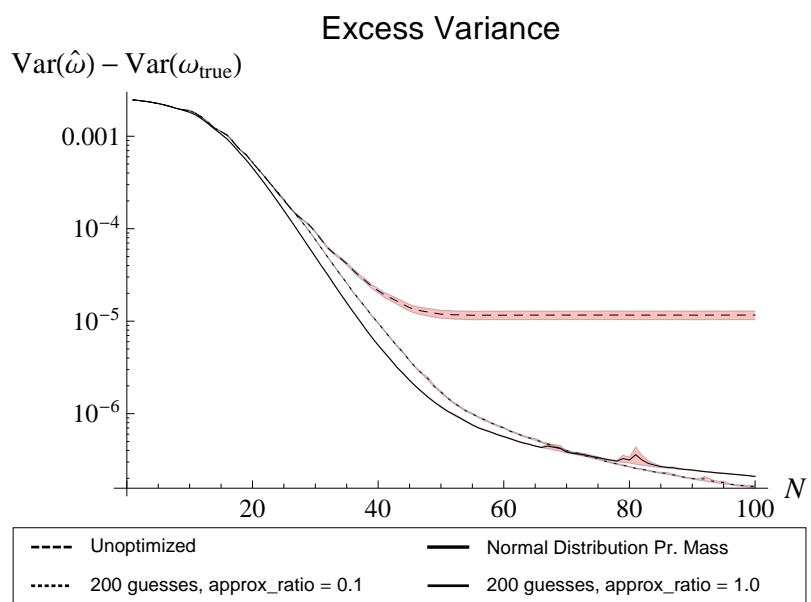


Figure 3.2: Excess covariance of the hyperregion estimator for the Gaussian-Larmor hyperparameter model (2.29), using 2,000 SMC particles, and the prior $(\mu, \sigma^2) \sim \mathcal{N}([0.5, 0.0025], [0.001^2, 0.0025^2])$.

3.2 Model Selection

Thus far, we have assumed that a model for our system is known, such that the true behavior of our system is described by a parameterization \underline{x}_0 of that model. In many cases, however, we may not know a model that contains the true behavior, or may wish to diagnose whether our model is rich enough to describe our system.

One way to do so is to use *model selection* methods to compare our model to other models, such that we can learn the validity of our model, rather than demand that it be assumed. For example, Blume-Kohout and van Enk have argued that model selection can be used to detect violations of the assumptions made by state tomography [181].

3.2.1 AIC, BIC and Bayes Factors

In order to choose a model that best predicts future data, we must first have a means to assess the ability of a model to do so. The Akaike information criterion (AIC) [182] is a commonly-used approach to choose a model that minimizes the **risk** incurred according to the Kulback-Leibler divergence loss function. In particular, the model M that maximizes

$$\text{AIC}(M) = \max_{\underline{x}} \log \Pr(D|\underline{x}) - k, \quad (3.7)$$

is taken to be correct, where $k = \dim \underline{x}$ is the number of parameters for the model M . That is, the AIC states that model parameters must be justified by an exponential improvement in the maximum likelihood. In this way, the AIC penalizes *overfitting*, in which the current dataset is precisely explained at the cost of predictive power. Thus, the AIC gives a formal way of finding models which have enough parameters to explain the data, without also inadvertently fitting to statistical fluctuations. In the context of quantum information, the AIC has been used, for example, to discriminate between classes of states [183], and in particular, to characterize the sources of entanglement [184].

From a Bayesian context, we can instead approximate the posterior as one that decays exponentially about its maximum to obtain the Bayesian information criterion [185; 186]

$$\text{BIC}(M) = \max_{\underline{x}} \log \Pr(D|\underline{x}) - k \ln |D| \approx \ln \Pr(M|D), \quad (3.8)$$

where $|D|$ is the number of samples that have been collected. Maximizing the BIC then gives the model with the largest posterior probability $\Pr(M|D)$. Note that even though the BIC involves the value of the likelihood at the maximum likelihood estimator, it does

not assume that we report the MLE as our estimate. The extra logarithmic factor between the BIC and the AIC can be interpreted as conservatism on the part of Bayesian inference, in that additional parameters much justify themselves much more strongly for the BIC than the AIC.

If one has access to the posterior directly, however, the BIC is not actually necessary, but rather approximates the logarithm of the Bayes factor

$$\text{BF}(M_A, M_B) := \frac{\Pr(M_A|D)}{\Pr(M_B|D)} = \frac{\Pr(D|M_A) \Pr(M_A)}{\Pr(D|M_B) \Pr(M_B)}. \quad (3.9)$$

In the special case of a uniform prior over models, this reduces to the likelihood ratio test [187].

The AIC, BIC and Bayes factors all provide, in addition to model selection, a way of deciding whether our models are useful at all. By using a model that is more general than the model of interest, we can reason about whether our chosen model can explain the data that we have observed. This technique has been used, for instance, to assess whether a given error model is correct, such that appropriate error-correction codes can be designed and applied [188]. This will be especially important when applying characterization to verify the correct performance of Hamiltonian evolution in [Chapter 6](#).

3.2.2 Bayes Factors with SMC

In the SMC update step (2.22), the normalization variable $\mathcal{N} = \sum w_i \Pr(d|x_i)$ is computed implicitly by summing over the unnormalized weights. As presented thus far, the variable \mathcal{N} is then *discarded*, along with any information that it carries about the inference procedure. We note, however, that \mathcal{N} is in fact the SMC approximation of $\mathbb{E}_{\underline{x}}[\Pr(d|\underline{x})] = \Pr(d)$, the total likelihood of a datum d . If we interpret this as being conditioned on the particular class M_A of models that \underline{x} parameterizes, we thus gain approximations of $\Pr(d|M_A)$ at each SMC update. As seen in [Section 3.2.1](#), this quantity can be used to compare A to other models by considering the Bayes factor $\Pr(d|M_A) / \Pr(d|M_B)$ for that datum.

Thus, model selection is implicitly carried out by running *parallel* instances of SMC on the same data set, as the weight over the total posterior cloud encodes the relative likelihood of that model. By the same argument, we can also consider the *model averaged estimate* (MAE), that is, the expectation value over models of those parameters shared between the models being considered [36]. An immediate consequence of the model averaged estimate approach is that MAE results in an SMC posterior over shared

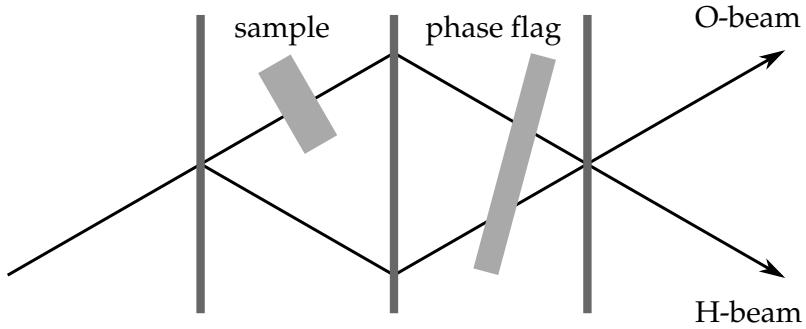


Figure 3.3: Schematic of a three-blade neutron interferometer with a controllable phase flag and a sample inserted in one arm. Detectors are placed at the ends of the two interferometric paths, and are called the ``O-beam'' and the ``H-beam'' by convention.

parameters, such that the arguments we have already developed for region estimation (Section 3.1), state-space tracking (Section 2.2.4) and experiment design (Section 2.1.2.3) with SMC proceed accordingly. For instance, we can formulate MAE-credible regions over shared parameters that account for uncertainty about models.

Later, in Section 6.2.3.3, we shall use Bayes factors as produced by QInfer to decide the validity of a truncated model.

3.2.3 Applications to State-Space Models

Model selection can also be combined with hyperparameter and state-space models (Section 2.2.4) to reason about how a stochastic system evolves in time. Here, we consider two distinct examples, in neutron interferometry and in nitrogen vacancy centers.

3.2.3.1 State-Space Neutron Interferometry

In neutron interferometry, an incoming beam of neutrons is split by Bragg scattering, and then later recombined to form a Mach-Zehnder interferometer [189; 190], as illustrated in Figure 3.3. By rotating a block of material such that the path length difference between the two arms is changed, the count statistics at the final detectors trace out a sinusoid that depends on the constant path-length difference between the two arms (Figure 3.4). Repeating this procedure both with and without a sample placed in one of the arms allows for a measurement of the path length through that sample.

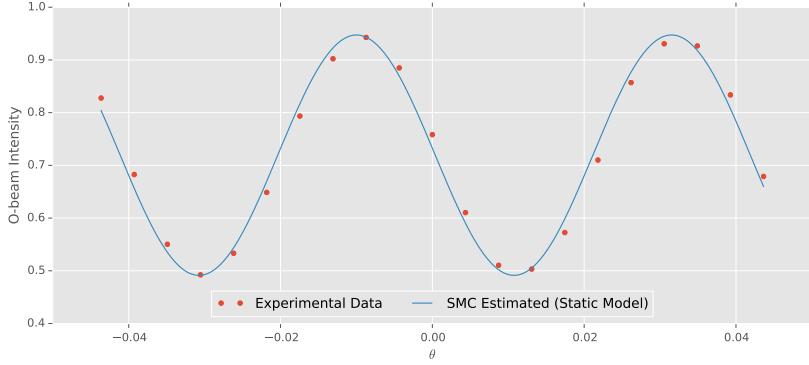


Figure 3.4: Example data collected from a three-blade neutron interferometer, shown with an SMC fit to the static model (3.10). Note that there is a slight phase difference between the static fit and the average data, due to the static model not accounting for stochasticity in the interferometer configuration.

This measurement procedure depends, however, on the path-length difference between the arms being *constant* throughout the experiment. Here, we apply model selection between a pure parameter-estimation model and a state-space model of the same neutron interferometer to decide whether this assumption holds, and to probe the degree to which the phase difference drifts with time. Similar techniques have been used to reason about the drift of measurement operators in a tomographic setting [181].

In particular, we will model neutron interferometry by a Bernoulli trial over which detector clicks, conditioned on a click happening at either detector. For the constant-parameter model, we consider a phenomenological likelihood function corresponding to the most general sinusoid that always yields a probability,

$$\Pr(\text{O-beam} | A, b, \alpha, \phi; \theta, \text{click}) = \frac{A}{2} \cos(\alpha\theta + \phi) + \frac{1+b}{2}, \quad (3.10)$$

where A and b represent the amplitude and the bias of the measurement, α the dependence of the phase difference on the setting θ of the controllable phase flag, and where ϕ represents the phase difference of interest.

For the case in which $\phi = \phi(t)$ is a stochastic function of time, we use the same likelihood function, but add a model parameter σ_ϕ such that

$$\phi(t_{k+1}) | \phi(t_k) \sim N(\phi(t_k), \sigma_\phi^2(t_{k+1} - t_k)^2). \quad (3.11)$$

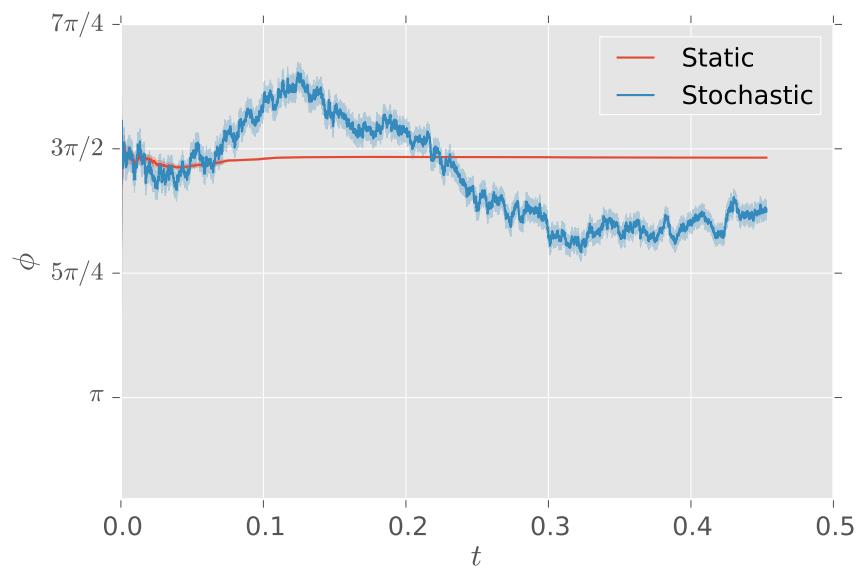


Figure 3.5: History of the estimator $\hat{\phi}$ of the phase difference between two arms of a three-blade interferometer as a function of the number of data points considered. (Left) No stochasticity assumed, (Right) $\phi = \phi(t)$. Shaded bands indicate the standard deviation of the posterior over ϕ .

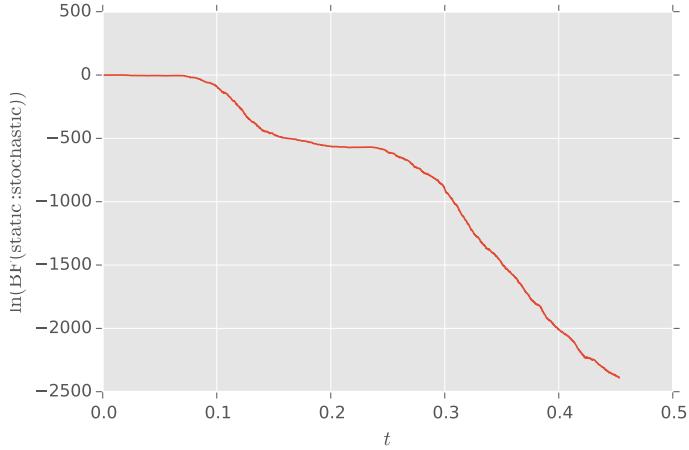


Figure 3.6: Logarithm of the Bayes factor between the static and stochastic models for the three-blade neutron interferometer. Negative values correspond to rejection of the static model in favor of the stochastic model.

In Figure 3.5, we show the results of estimating ϕ with each of these models in turn, using about 6000 data points consisting of approximately 40 neutrons each. In Figure 3.6, the Bayes factors between the two models is shown, illustrating that the stochastic model is to be preferred by a factor of approximately e^{2400} .

This application provides strong evidence that the data is not best explained by a constant phase difference, but by one that depends on time in a stochastic fashion. Future work will explore if a model of the relevant physics will allow predicting this stochasticity from measurements of secondary variables, such as the temperature. Using model selection in this way will allow us to decide the validity of any proposed physical model, and thus to recover measurements of the phase difference.

3.2.3.2 Data Referencing for Nitrogen Vacancy Centers

In Section 2.3.3, we discussed that data is *referenced* by secondary experiments that measure the visibility and dark counts before each primary experimental measurement. The bright and dark references, however, follow a random walk such that a state-space model can not only estimate references but *predict* them, in analogy to the use of state-space SMC in computer vision [133].

Let $\alpha = \alpha(t) = \Pr(\text{detect}|\text{emit}; t)$ be the bright referencing parameter and let $\beta = \beta(t) = \Pr(\text{detect}|\neg\text{emit}; t)$ be the dark referencing parameter. Then, using the state-space model

$$\alpha(t_{k+1})/\alpha(t_k) \sim N(\mu_\alpha, \sigma_\alpha^2) \quad (3.12a)$$

$$\beta(t_{k+1})/\beta(t_k) \sim N(\mu_\beta, \sigma_\beta^2), \quad (3.12b)$$

we can effectively predict references, as shown in [Figure 3.7](#). This comes at the cost, however, of introducing four additional model parameters. In practice, α and β are not independent, such that we introduce the possibility of over-fitting. That is, the model of [\(3.12\)](#) is effectively a free model for the state-space diffusion of α and β .

As an alternative model, we consider a five-parameter model that explicitly ties α and β using a dark-count parameter δ ,

$$\frac{\alpha(t_{k+1}) - \delta}{\alpha(t_k) - \delta} \sim N(\mu, \sigma^2) \quad (3.13a)$$

$$\frac{\beta(t_{k+1}) - \delta}{\beta(t_k) - \delta} \sim N(\mu, \sigma^2). \quad (3.13b)$$

Importantly, in order for [\(3.13\)](#) to be a better model than the free-drift model [\(3.12\)](#), it must not only outperform the free model at predicting the references, but also at learning parameters from experiments following those references.

To evaluate this, we revisit the analysis of [Section 2.3.3.3](#), using in parallel both state-space models as well as the pure parameter-estimation model discussed there. An example of estimating a single trajectory using each model is shown in [Figure 3.7](#). We then compare the Bayes factors for each of the three, as shown in [Figure 3.8](#), to conclude that the dark count drift model is a better model not only for the reference data, but also improves the model for the underlying Hamiltonian. In particular, the dark count drift model is to be preferred over the maximum-likelihood estimator for the reference trajectories by approximately five orders of magnitude. This is especially important in the example of hyperfine estimation, as some model parameters are primarily exposed as effective visibilities, such that uncertainty in the visibility becomes correlated with that in the parameters of interest.

In this way, we can not only verify the quantum dynamics of our systems, but also the assumptions that are made about classical experimental models through which we view our quantum systems. Taken in combination with techniques for controlling systems by using models of classical distortions ([Section 5.2](#)), verification via model selection and model-averaged estimation thus provide us with a powerful tool for quantum control.

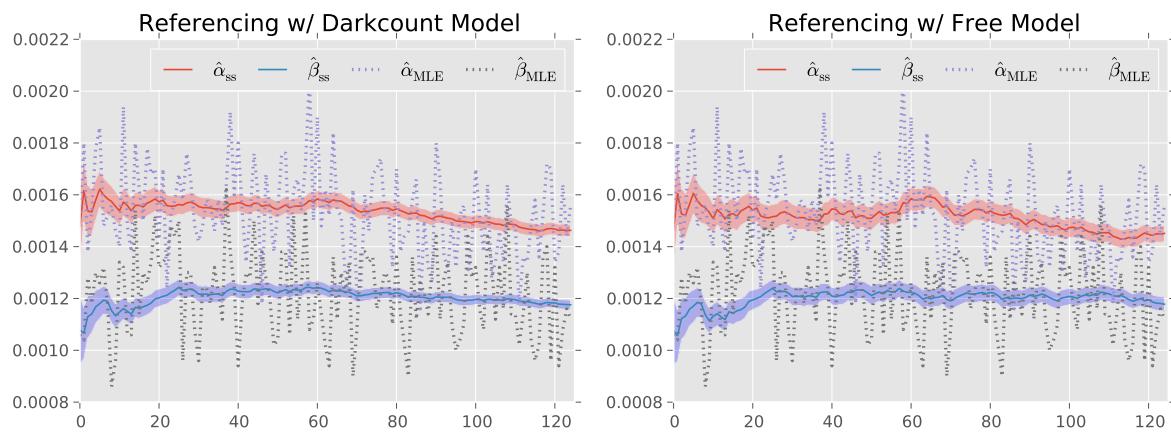


Figure 3.7: Example of reference trajectories as estimated by parameter-space maximum-likelihood estimator and the state-space models of (3.12) and (3.13).

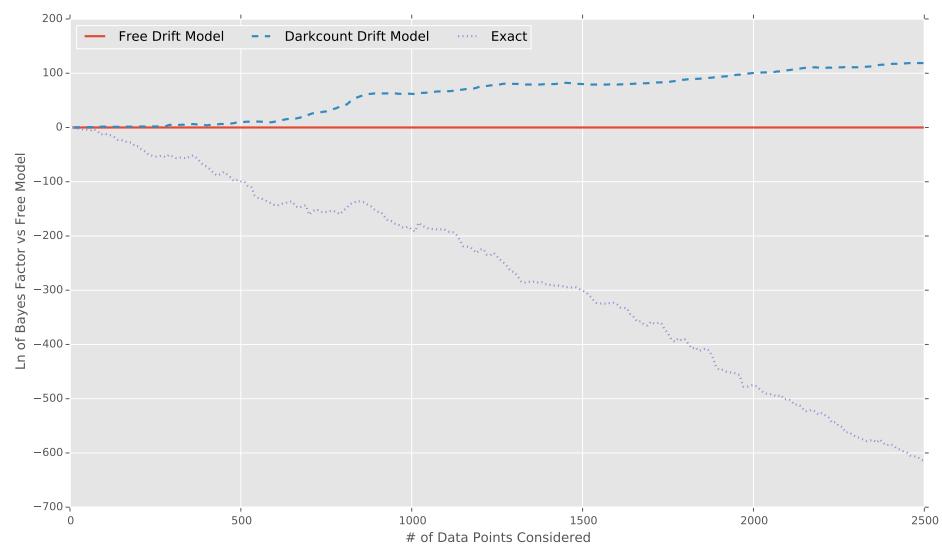


Figure 3.8: Total likelihoods for the nitrogen-hyperfine coupling experiments described in Section 2.3.3.3, using each of several different referencing models and the same physical model, compared to the total likelihood for the free model. Using the dark count model is thus preferred to the free model, while the traditional ("exact") reference model is dispreferred.

4 Honest Approximation

When evaluating candidate quantum information processing systems, a critical question is how well they will perform in the context of a proposed fault-tolerance and quantum error correction protocol. If a quantum system cannot be used fault-tolerantly, then that precludes using the system in a scalable fashion.

In order to answer this, we must either have an adversarial and analytically-proven threshold [191; 149], or we must be able to simulate a numerical threshold using many copies of the characterized system. The latter, owing to the large sizes of the quantum systems used by proposed fault-tolerant protocols, cannot be evaluated using classical resources alone if one demands simulating using the full Hilbert space description of the proposed fault-tolerance. Instead, efficiently simulatable subtheories of quantum mechanics are often used to carry out the fault-tolerance evaluation. For instance, numerical thresholds can be estimated for planar and surface codes using randomly drawn Pauli channels to represent errors, and then propagating these errors using improved Gottesman-Knill simulation [63; 192].

Unlike in the case of adversarial thresholds, however, we have no reason to *a priori* expect that the numerical estimation of a threshold for Pauli channels should inform us of a threshold for more general channels. Thus, in order to evaluate the performance of a generic device as used in a proposed fault-tolerant context, we must have a way of approximately *reducing* a description of the action of that device to an efficiently simulatable channel. Moreover, we shall demand that this approximation be *honest*, in the sense that errors are not underestimated [109].

It is important to note here that an approximation or estimate which is not honest is not necessarily useless, but rather that care must be taken in its application. For instance, randomized benchmarking (Section 2.3.2) finds the fidelity of an approximate channel which is, by the following definition, not an honest representation of the gate or gateset of interest. In particular, randomized benchmarking estimates the strength of a depolarizing channel having the same fidelity with the identity as the actual channel. As such, randomized benchmarking alone does not honestly capture the worst-case errors that a channel could incur. This distinction can be of particular importance when evaluating error correcting codes, as we shall see in Section 4.3.

Despite this, the transformed channel measured by randomized benchmarking explicitly preserves the fidelity. Thus, if the fidelity is the quantity of interest, we should not be bothered by the fact that randomized benchmarking is not honest according to the definition below. Moreover, in the case of randomized benchmarking, the worst-case scenarios can be bounded [161; 193], such that we can have confidence in our use of fidelity, even in absentia a guarantee of an exactly honest representation of errors. Indeed, for these reasons, the fidelity can be useful as an optimization functional, as we will discuss in Chapter 5.

Formally, for a channel Λ representing an imperfect implementation of a unitary U , and for a set of efficiently simulatable channels S , we wish to find $\tilde{\Lambda}$ according to the optimization program [37; 164; 38]

$$\begin{aligned} \min \quad & \|\Lambda - \tilde{\Lambda}\|_{\diamond} \\ \text{subject to} \quad & \forall \text{ pure states } \rho : \|\Lambda[\rho] - U \bullet \rho\|_1 \leq \|\tilde{\Lambda}[\rho] - U \bullet \rho\|_1 \\ & \tilde{\Lambda} \in S, \end{aligned} \tag{4.1}$$

where $U \bullet \rho := U\rho U^\dagger$, $\|\cdot\|_1$ is the Schatten 1-norm on states and where $\|\cdot\|_{\diamond}$ is the diamond norm on channels [194].

In order to explicitly reason about worst-case performance, we choose to express the concept of honesty in terms of the Schatten 1-norm. Informally, if an approximation preserves or increases the 1-norm distance between a true channel and its ideal action, then we will say that it is honest, as it has not neglected the worst-case performance of the unapproximated channel.

The 1-norm offers an operational interpretation as the unregularized (that is, single-shot) distinguishability between two states [65]. That is, given a single copy of either $U \bullet \rho$ or $\Lambda[\rho]$, we can distinguish them with probability of error

$$\Pr(\text{error}) = \frac{1}{2} - \frac{1}{4}\|U \bullet \rho - \Lambda[\rho]\|_1, \tag{4.2}$$

assuming a uniform prior over the two possibilities [195]. In the special case that $U = \mathbb{1}$, the 1-norm then defines the *input-output distinguishability*

$$d_{\text{IO}}(\Lambda, \rho) := \|\Lambda(\rho) - \rho\|_1. \quad (4.3)$$

The honesty condition then becomes the partial ordering

$$\Lambda \preceq \tilde{\Lambda} \iff \forall \text{ pure states } \rho : d_{\text{IO}}(\Lambda, \rho) \leq d_{\text{IO}}(\tilde{\Lambda}, \rho), \quad (4.4)$$

giving

$$\begin{aligned} & \min \quad \|\Lambda - \tilde{\Lambda}\|_\diamond \\ & \text{subject to } \Lambda \preceq \tilde{\Lambda} \text{ and } \tilde{\Lambda} \in S \end{aligned} \quad (4.5)$$

as the optimization program.

As an implementation note, the diamond norm can be computed by means of a semidefinite program in the Stinespring dilation [196]; the QuantumUtils library for MATLAB [43] implements this program with CVX [197], allowing us to quickly compute $\|\Lambda - \tilde{\Lambda}\|_\diamond$ for channels Λ and $\tilde{\Lambda}$ acting on small systems.

4.1 Stabilizer Formalism as an Efficiently Simulatable Subtheory

Thus far, we have left unspecified our choice of efficiently simulatable set S . There are several different subtheories of quantum mechanics which are classically simulatable [198; 199; 63; 200; 201]. Here, however, we shall focus on the stabilizer formalism (Section 1.5.2), such that we want to design honest approximations using mixed unitary channels over the Pauli group \mathcal{P}_n and the Clifford group \mathcal{C}_n . Such mixed channels are known as Pauli channels and Clifford channels, respectively. These channels are important to the study of error correction and fault-tolerance, as they can be classically simulated in $\text{poly}(n)$ time using Gottesman-Knill simulation [62; 63; 149] together with Monte Carlo sampling (GK-MC) [192].

4.2 Implementation of Honesty Approximation

The honesty constraint on the optimization program (4.5) is difficult to evaluate in its given form, as it requires checking that a condition holds for all pure states. An alternative expression in terms of quadratic forms has been given by Puzzuoli [37; 164; 38],

however, that is manifestly independent of the state. In particular, if Λ acts on Bloch vectors as $\Lambda : \underline{r} \mapsto \underline{\underline{M}}_\Lambda \underline{r} + \underline{t}$ for a matrix $\underline{\underline{M}}_\Lambda$ and a vector \underline{t} , and if $\tilde{\Lambda} : \underline{r} \mapsto \underline{\underline{M}}_{\tilde{\Lambda}} \underline{r}$, then $\Lambda \prec \tilde{\Lambda}$ if and only if

$$\|\underline{r} - M_{\tilde{\Lambda}} \underline{r}\|_2 \geq \|\underline{r} - (\underline{\underline{M}}_\Lambda \underline{r} + \underline{t})\|_2. \quad (4.6)$$

This is in turn true if and only if $\underline{\underline{A}} \geq \underline{\underline{B}}$, where

$$\underline{\underline{A}} := (\mathbb{1} - \underline{\underline{M}}_{\tilde{\Lambda}})^T (\mathbb{1} - \underline{\underline{M}}_{\tilde{\Lambda}}) \quad (4.7a)$$

$$\underline{\underline{B}} := (\mathbb{1} - \underline{\underline{M}}_\Lambda)^T (\mathbb{1} - \underline{\underline{M}}_\Lambda) + (\|\underline{t}\|_2 + 2\|(\mathbb{1} - \underline{\underline{M}}_\Lambda)^T \underline{t}\|_2) \mathbb{1}. \quad (4.7b)$$

For any given channel $\tilde{\Lambda}$ in our approximation set, we can thus check for the positivity of $\underline{\underline{A}} - \underline{\underline{B}}$ to determine if $\tilde{\Lambda}$ is feasible for (4.5). In particular, we can implement this procedure using QuantumUtils for MATLAB [43] to express the feasibility as a nonlinear constraint function as expected by MATLAB's `MultiStart` global optimizer.

4.3 Application to Error Correction

Having developed a method for honestly approximating channels by channels drawn from an efficiently simulatable set, it remains to be shown that this approximation remains honest in a scenario of interest. Thus, we consider an implementation of the five-qubit perfect code [83] and syndrome readout, and the effects on this procedure of honestly approximating each gate in the gateset used to in the implementation.

More precisely, we wish to consider a round of error correction as a channel $\overline{\Lambda}_{\text{gadget}}$ that acts on the logical space of an error correcting code. To extract a representation of this channel from a simulation of the channel Λ_{gadget} acting on the physical qubits, we surround Λ_{gadget} with an ideal encoder and decoder, as shown in Figure 4.1, thus exploiting the Choi-Jamiołkowski isomorphism [65]. In this way, we can obtain a description of the logical channel $\overline{\Lambda}_{\text{gadget}}$ on an $[\![n, k, d]\!]$ code with a full and explicit simulation on $2n$ qubits. Since the gates that we consider each act nontrivially on at most 2 qubits at a time, we can use the subsystem application method of Criger ¹. Using this technique, we can readily simulate the 10 qubits necessary to analyze honest approximations substituted into a five-qubit perfect code, with one logical qubit and four ancilla qubits.

¹An implementation and description of the Criger subsystem application method can be found in the `subsystem_apply` function of QuTiP [1], available at https://github.com/qutip/qutip/blob/master/qutip/subsystem_apply.py

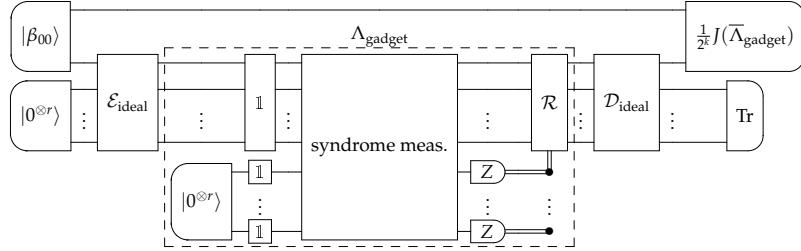


Figure 4.1: Circuit layout for extracting behavior of error-correction round $\bar{\Lambda}_{\text{gadget}}$.

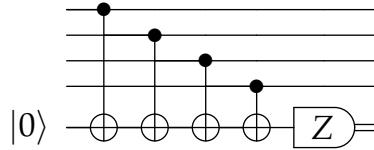


Figure 4.2: Fowler schedule for applying CNOT gates to measure XXXX syndromes in a surface or lattice code.

The circuit that we use for syndrome readout is given as [Figure 4.3](#), along with the calls to QuaEC used to derive the circuit. This circuit was chosen as it is not explicitly fault-tolerant, and resembles the Fowler schedule for CNOT measurements [\[192\]](#), as illustrated in [Figure 4.2](#).

For each gate set, we then simulate the circuit in [Figure 4.1](#) using the original gate set, the Pauli-twirl of each gate, and the Pauli and Clifford channel honest approximations to find $J(\bar{\Lambda}_{\text{gadget}})$ for each combination of gate set and approximation method. For brevity, we let $\bar{\Lambda} = \bar{\Lambda}_{\text{gadget}}$. We then tabulate several statistics on the relative error-correction gadget channels:

χ_{00} Average gate fidelity of $\bar{\Lambda}_{\text{gadget}}$ with the identity channel.

$\|\bar{\Lambda} - \Lambda_{\text{ideal}}\|_\diamond$ Distinguishability of the logical and ideal channels.

$\|\bar{\Lambda} - \bar{\Lambda}_{\text{original}}\|_\diamond$ Distinguishability of the logical channels obtained from the original and approximated gate sets.

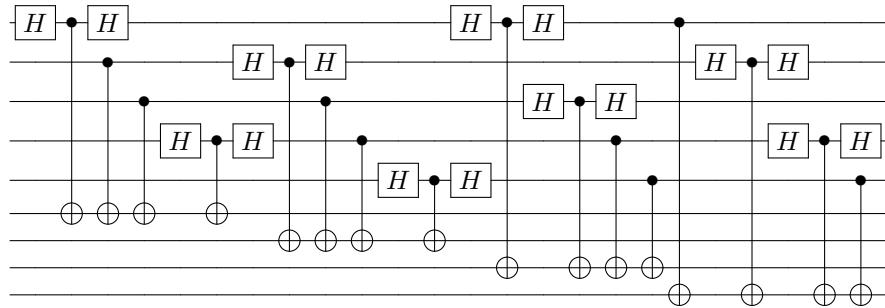
\bar{h} Average hedging/honesty parameter defined by

$$\begin{aligned}\bar{h} &:= \int d\psi h(\bar{\Lambda}, \bar{\Lambda}_{\text{original}}, |\psi\rangle\langle\psi|) \\ &= \int d\psi [d_{\text{IO}}(\bar{\Lambda}, |\psi\rangle\langle\psi|) - d_{\text{IO}}(\bar{\Lambda}_{\text{original}}, |\psi\rangle\langle\psi|)].\end{aligned}\tag{4.8}$$

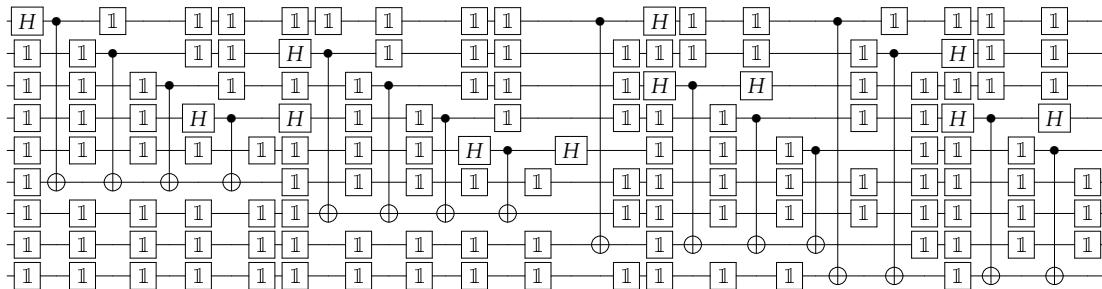
p_{viol} Probability over pure states ρ that the honesty condition is violated for ρ . (An approximation is honest if and only if p_{viol})

To approximate the integrals in the calculations of \bar{h} and p_{viol} using Monte Carlo sampling, we propagate a large number of pure states through the logical channel. The results of this analysis are shown in [Table 4.1](#) for each of three gate sets GS1, GS2 and GS3, each obtained from the cumulant expansion applied to a physical model [[164](#)]. GS1 describes a prototypical double quantum dot system, while GS2 and GS3 are simulated from a physical model of superconducting qubits using different parameters. In GS2, the gates are built from an XY4 sequence, while in GS1 and GS3, no refocusing is used.

We note that Pauli twirling of GS1 results in a gate set that violates the honesty criterion for *every* pure state sampled. By contrast, GS3 demonstrates a good regime for Pauli twirling, as explored in depth by Puzzuoli [[38](#)].



(a) Original circuit.



(b) Circuit with simplifications and with explicit wait locations.

Source for figure (b).

```

>>> import qecc as q
>>> stab = q.StabilizerCode.perfect_5q_code()
>>> print (
...     stab.syndrome_meas_circuit()
...     .cancel_selfinv_gates()
...     .pad_with_waits()
...     .as_qcircuit(0.2, 0.2)
... )

```

Figure 4.3: Syndrome measurement with and without explicit wait locations.

Table 4.1: Statistics for $\bar{\Lambda}_{\text{gadget}}$ using $N = 10^6$ randomly sampled pure states.

Set/Statistics	Original	Pauli twirled	Pauli	Clifford
GS ₁	χ_{00}	0.999964	0.999964	0.985820
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{ideal}}\ _\diamond$	4.76×10^{-3}	7.28×10^{-5}	2.84×10^{-2}
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{original}}\ _\diamond$		4.76×10^{-3}	2.87×10^{-2}
	\bar{h}		-3.69×10^{-3}	1.85×10^{-2}
GS ₂	p_{viol}	1.	0.	0.
	χ_{00}	0.991372	0.991372	0.991355
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{ideal}}\ _\diamond$	1.73×10^{-2}	1.73×10^{-2}	1.73×10^{-2}
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{original}}\ _\diamond$		2.45×10^{-5}	4.29×10^{-5}
GS ₃	\bar{h}		-1.63×10^{-8}	2.24×10^{-5}
	p_{viol}	0.55566	0.	0.
	χ_{00}	0.992495	0.987594	0.969499
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{ideal}}\ _\diamond$	1.51×10^{-2}	2.48×10^{-2}	6.10×10^{-2}
	$\ \bar{\Lambda} - \bar{\Lambda}_{\text{original}}\ _\diamond$		1.03×10^{-2}	4.60×10^{-2}
	\bar{h}		6.36×10^{-3}	3.04×10^{-2}
	p_{viol}	0.	0.	0.

5 Control Design and Optimization

The development of high-fidelity quantum control sequences is critical to producing useful quantum information processors. Indeed, fault-tolerance requires reaching demanding thresholds [62; 192], such that high-fidelity control is essential to protect quantum information during a computation. One approach that is often used in developing such control is that of *optimal control theory* (OCT), in which a performance functional of a control sequence, such as the fidelity, is treated as an *objective function*, and is maximized over a set of feasible sequences. OCT has been used in a wide range of applications, including the development of robustness to variations in Hamiltonian dynamics [202], universal control through actuators [203], the preparation of highly polarized nuclear spin systems [136], for the development of novel quantum metrology experiments [204; 205].

As with many optimization problems, gradient information can be used to aid in maximizing the fidelity of control pulses. In particular, the well-known gradient ascent pulse engineering (GRAPE) algorithm [84] has been widely applied [202; 206; 207; 208] to control design problems in quantum information processing. In this Chapter, we start by introducing GRAPE algorithm and showing how it may be extended to include descriptions of classical control electronics and augmented with heuristics for efficient energy removal [209; 210; 39]. Then, we show how OCT-designed pulses can be used in composite sequences to implement useful quantum computing architectures that natu-

rally map onto proposed quantum error correction geometries. Finally, we note the inherent limitations of gradient methods and discuss novel methods for mitigating these limitations using memetic algorithms--- that is, genetic algorithms that improve individuals between selection steps. This will then pave the way for the inclusion of quantum simulation resources in [Chapter 6](#).

5.1 GRAPE and Conjugate-Gradient Methods

In order to affect the highest-fidelity control possible, we want to use an accurate model of the system when designing control pulses. That is, we require that we can find a pulse that operates correctly in experimentally-reasonable conditions, using classical electronics models that exhibit finite bandwidths, power limitations, and even nonlinear circuit elements. This latter requirement is especially important with the development of novel superconducting resonator designs [211; 212; 213] that admit strong nonlinear inductances.

To develop a procedure to design pulses for such nonlinear systems, we first start with the ideal case wherein, given a unitary target U_{targ} , our task is to find \underline{p} for the optimization program,

$$\begin{aligned} &\text{given } U_{\text{targ}} \\ &\max o(\underline{p}) := \Phi(U_{\text{targ}}, U[\underline{p}]), \end{aligned} \tag{5.1}$$

where

$$U[\underline{p}; \underline{x}] = \prod_{n=\text{len}\underline{p}}^1 \exp(-i[t_n - t_{n-1}][H_{\text{int}} + p_n H_c]). \tag{5.2}$$

We can also consider multiple control fields, in which case \underline{p} is replaced by a matrix $\underline{\underline{p}} = [p_{n,k}]$ whose first index is a timestep and whose second index is a control field.

In order to solve the optimization program (5.1) using a classical computer, we must simulate $U[\underline{p}; \underline{x}]$ for a large number of candidate pulses \underline{p} . Since the control field is defined to be piecewise constant, this involves calculating each of the unitary timesteps in (5.2). Khaneja et al [84] showed that this information can be used to extract not only the objective function o evaluated at \underline{p} , but also the gradient $\nabla_{\underline{p}} o$. In particular, let $U_i := \exp(-i[t_n - t_{n-1}][H_{\text{int}} + p_n \cdot \underline{H}_c])$ for a control Hamiltonian vector $\underline{H}_c := (H_{c,1}, \dots, H_{c,K})$,

such that $U[\underline{p}; \underline{x}] = \prod_{n=\text{len } \underline{p}}^1 U_n$. Then,

$$\Phi(U_{\text{targ}}, U[\underline{p}; \underline{x}]) := \left| \langle\langle U_{\text{targ}} | U[\underline{p}; \underline{x}] \rangle\rangle \right|^2 \quad (5.3a)$$

$$\frac{\partial \Phi}{\partial p_{n,k}} = -2 \operatorname{Re} [\langle\langle P_n | i(t_n - t_{n-1}) H_{c,k} X_n \rangle\rangle \langle\langle X_n | P_n \rangle\rangle], \quad (5.3b)$$

where

$$X_n := \prod_{i=n}^1 U_i \quad \text{and} \quad P_n := \left(\prod_{i=n+1}^{\text{len } \underline{p}} U_i^\dagger \right) U_{\text{targ}} \quad (5.3c)$$

are the parts of UU_{targ}^\dagger arising from evolution before and after the timestep n , respectively. Importantly, this expansion of $\nabla_{\underline{p}} o$ requires calculating only a few more matrix products than calculating $o(\underline{p})$ itself, such that it does not contribute to the dominant simulation cost incurred by evaluating the objective function.

The use of gradient information is immensely helpful in many different contexts; in our case, we can integrate $\nabla_{\underline{p}} o$ into our optimization to dramatically reduce the number of simulation calls we will need to make, using a straightforward conjugate gradient ascent method detailed in [Algorithm 4](#), known as the gradient ascent pulse engineering (GRAPE) algorithm [84]. Implementations are available in several software libraries, including DYNAMO [214] and QuTiP [1]. Here, we are concerned with the implementation provided by QuantumUtils for Mathematica [42].

In addition to stopping when the target fidelity is reached, several different criteria can be used to detect a lack of improvement, such that a new initial guess can be tried. Such criteria include a maximum number of iterations or conjugate-gradient direction resets, or minimum allowable step size [42]. In the examples in this chapter, we perform line searches by minimizing a quadratic fit to the objective function evaluated at three points along the conjugate-gradient search direction.

5.2 Optimal Control with Realistic Circuit Models

5.2.1 Distortions

In practice, we do not actually have the ability to apply arbitrary pulses p . Rather, quantum systems are controlled by classical systems with finite power per bandwidth and

Algorithm 4 GRAPE: Conjugate Gradient optimization with Khaneja gradient calculation.

```

Choose initial pulse  $\underline{p}$ .
 $\beta \leftarrow 0, \underline{s} \leftarrow \underline{0}$ .                                ▷ Initialize conjugate direction calculations.
 $\underline{s}' \leftarrow \underline{0}, \Delta' \underline{p} \leftarrow \underline{0}$ .          ▷ Remember directions from the prev. step.
 $i \leftarrow 0$ .                                              ▷ Count the number of iterations.

while utility goal for  $o(\underline{p})$  is not yet met do
    Calculate  $o(\underline{p})$  and  $\Delta \underline{p} \leftarrow \nabla_{\underline{p}} o$  using (5.3).

     $\beta \leftarrow \max(0, \Delta \underline{p} \cdot (\Delta \underline{p} - \Delta' \underline{p}) / \Delta' \underline{p} \cdot \Delta' \underline{p})$ .
     $\underline{s} \leftarrow \beta \underline{s} - \Delta \underline{p}$ .
     $\alpha \leftarrow \text{LINESEARCH}(\lambda \alpha. o(\underline{p} + \alpha \underline{s}))$ .
     $\underline{p} \leftarrow \underline{p} + \alpha \underline{s}$ 

     $\Delta' \underline{p} \leftarrow \Delta \underline{p}, \underline{s}' \leftarrow \underline{s}$ .          ▷ Save the directions from this step.
     $i++$ .                                         ▷ Increment the iteration count.

end while
return  $\underline{p}$ 

```

noise figures, and thus introduce noise and distortion into our controls. Here, we are concerned mainly with the latter case, in which our controls are distorted by a deterministic operator $g : \mathbb{R}^{N \times K} \rightarrow \mathbb{R}^{M \times L}$. At this point, we assume only that g can be simulated. In [Figure 5.1](#), we show an example of a pulse distorted by a simple convolution operator

$$g_\star[\underline{p}; \underline{x}] := \underline{p} \star \underline{x}, \quad (5.4)$$

where \underline{x} is taken to be an exponentially-decaying kernel discretizing

$$k(t) = \begin{cases} \frac{1}{t_c} e^{-t/t_c} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (5.5)$$

for a time constant t_c . An exponential kernel of this form captures finite rise and ring-down times described by finite quality factors Q , and have previously been included into GRAPE to find pulses for high- Q systems [[210](#)].

For more general convolution kernels, such as those measured by the use of a pick-up or spy coil, we have that g_\star can be written as contraction with a four-index tensor $k_{nk,ml}$

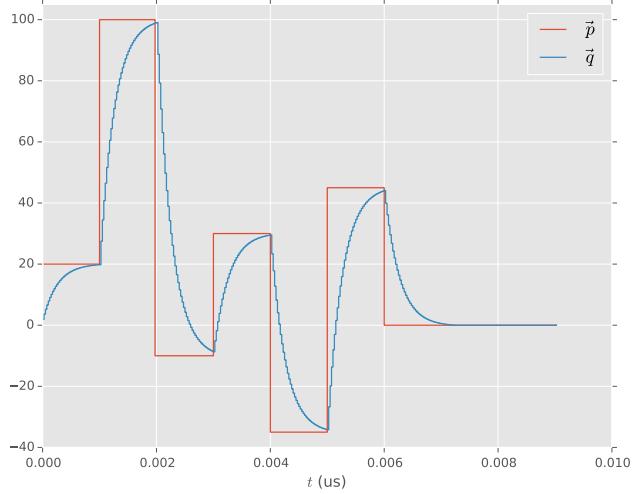


Figure 5.1: Example of an exponential convolution kernel $\underline{g}_*[p] = e^{-t/t_c}/t_c \star \underline{p} = \underline{q}$ applied as a distortion operator, where $t_c = 500$ ps.

of the nk indices, where

$$k_{nk,ml} := \int_{t_{n-1}}^{t_n} k_{l,k} \left(\frac{\tau_m + \tau_{m-1}}{2} - s \right) ds. \quad (5.6)$$

Here, $\{t_n\}$ are the time points at which \underline{p} is sampled, while $\{\tau_m\}$ are the time points at which \underline{q} is sampled.

More generally, we can also consider distortion operators that act in a more complicated manner than the linear kernels considered thus far. If we are able to compute $\underline{g}[p; \underline{x}]$ and the Jacobian $J_p(g)[p; \underline{x}]$ of g with respect to the input for an arbitrary pulse, then we can include g into gradient-based OCT algorithms by replacing u with $u_g := u \circ g$. The gradient of u_g is then given by

$$\nabla_{\underline{p}} u_g = \nabla_{\underline{q}} u[\underline{q}] \cdot J_p(g)[p; \underline{x}]. \quad (5.7)$$

In the special case of a linear map, such as is formed by multiplying \underline{p} by the right-hand indices of an operator \underline{k} , $\underline{g}_*[p] = \underline{k} \star \underline{p}$, the Jacobian is simply \underline{k} itself. More generally, we can approximate the Jacobian by taking a perturbative approach, as we

will do in [Section 5.2.2](#). Once we have a distorted objective and gradient-objective, we can then substitute each into [Algorithm 4](#) to find pulses that affect the desired action in the presence of this distortion.

5.2.2 Modeling Nonlinear Circuits

We are particularly interested in modeling the effect on pulses that is introduced by transmission through a superconducting resonator such as that detailed in [\[212; 213\]](#). Superconducting resonators do not respond linearly when driven strongly, as the inductance and resistance of superconducting circuits can, in general, depend on the incident power due to surface resistance and kinetic inductance [\[215; 216\]](#). In particular,

$$L(I) = L_0 \left(1 + \alpha_L |I|^2 \right) \quad (5.8a)$$

$$R(I) = R_0 (1 + \alpha_R |I|^\eta), \quad (5.8b)$$

where α_L , α_R and η are constants describing the nonlinear behavior of the circuit, and are often found by fitting the output of test pulses. If $\alpha_L |I|^2, \alpha_R |I|^\eta \ll 1$, then the device can be well-approximated by a linear resonator. This limits us to using much smaller powers, however. Instead, we would like to explicitly include the nonlinear behavior as a distortion, such that gradient-based optimization methods will find pulses that affect the desired action, even well away from the linear regime.

We model nonlinear resonators by a one-port effective circuit of the form shown in [Figure 5.2](#), where L and R are functions of the current I_L through the inductor, as given by [\(5.8\)](#). The voltage source $V_s(t)$ represents the driving of the circuit by an external signal generator, such as an FPGA with a DAC daughterboard [\[123\]](#). The Hamiltonian that spins sitting in the field of the inductor are subject to is then given by $H(t) = \gamma\kappa\phi(t)$, where $\phi(t)$ is the flux through the inductor, found by applying the constitutive law for the current-dependent inductance, and where κ is a *geometric conversion factor* that relates the flux to the field at the spins. Thus, if we know the current $I_L(t)$ through the inductor for a given driving function $V_s(t)$, we can simulate the action of that pulse on the spin ensemble.

Because the inductance and resistance are no longer constant, the resonance frequency ω_0 and quality factor Q are dynamic functions of the incident power, the tuning of the circuit is also dynamic, such that we must explicitly include the matching capacitance C_m in order to model the loss of coupling that occurs as the power is increased. In [Figure 5.3](#), we show the result of a square x -axis pulse distorted by such a nonlinear circuit, demonstrating the phase shift, change in resonance and loss of coupling.

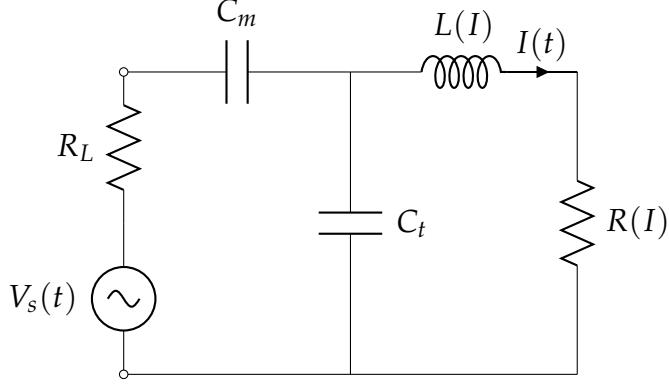


Figure 5.2: Parallel one-port RLC circuit with non-linear inductance and resistance and matching capacitance.

We note that a resonator of this kind can no longer be simulated by convolution of the driving function with some kernel, as convolution distortion operators are explicitly linear. Rather, we must integrate the system of differential equations describing the state variables for the circuit [216]. More efficient approximation techniques, such as harmonic balance [217], can be used to obtain information about the steady-state response, but as we are interested in the transient behavior, we use full time-domain integration. In particular, the state equations for the one-port model shown in Figure 5.2 are given by

$$\frac{d}{dt} \begin{bmatrix} I_L \\ V_{C_m} \\ V_{C_t} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 & \frac{1}{L} \\ 0 & \frac{-1}{R_L C_m} & \frac{1}{R_L C_m} \\ \frac{-1}{C_t} & \frac{1}{R_L C_t} & \frac{1}{R_L C_t} \end{bmatrix} \begin{bmatrix} I_L \\ V_{C_m} \\ V_{C_t} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{V_s(t)}{R_L C_m} \\ \frac{V_s(t)}{R_L C_t} \end{bmatrix}, \quad (5.9)$$

where L and R are given by (5.8). The solution to this system of differential equations for a given driving function $V_s(t)$, together with the map from inductor currents to fields,

$$B(t) = \kappa \phi(t) = \kappa I_L(t) L(I_L(t)), \quad (5.10)$$

gives us the distortion operator $g : V_s \rightarrow B$.

Since this distortion operator responds to driving pulses in a nonlinear fashion, such as in Figure 5.3, we must include its effects into our gradient calculations. In particular, we approximate J_g by expanding about the zero pulse $V_s(t) = 0$,

$$\left. \frac{\partial g_{m,l}}{\partial p_{n,k}} \right|_p \approx \left. \frac{\partial g_{m,l}}{\partial p_{n,k}} \right|_0 \approx \left[\frac{g(\epsilon e_{n,k})}{\epsilon} \right]_{m,l}, \quad (5.11)$$

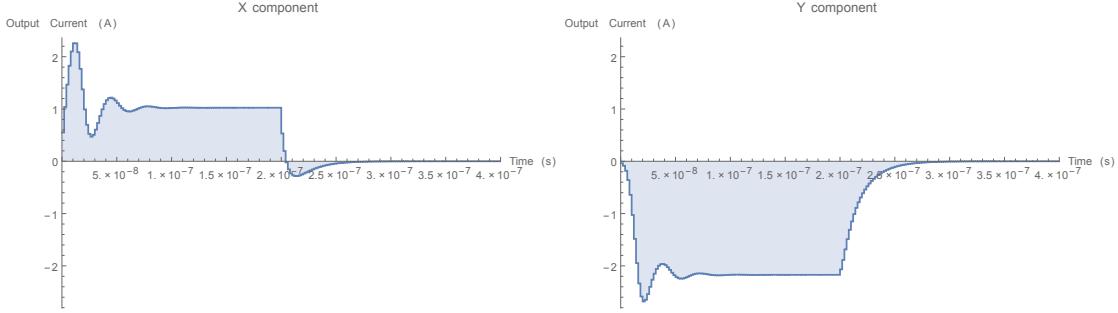


Figure 5.3: Example of a 200 ns, 7 V square pulse along x after being transmitted through a nonlinear resonator with $\alpha_I = 0.0006/\text{A}^2$ and $\alpha_R = 0$.

where ϵ is a small step size used in computing the gradient. While this is a very coarse approximation of the gradient, it captures enough dynamics for the gradient-based algorithms to pick a direction that is still uphill, and offers the advantage that we can precompute J_g using this approximation. That is, we can compute a good direction with no additional circuit simulation calls. In cases where the distortion computation is inexpensive compared to simulating the quantum evolution, a more sophisticated approximation may be used here to trade off between circuit and quantum system simulation calls.

Having thus implemented both $g[p]$ and $J_g(p)$ for the distortion operator $g : V_s \rightarrow B$ representing the nonlinear resonator, we can now include the resonator into our GRAPE implementation. We will consider internal and control Hamiltonians for a single qubit,

$$H_{\text{int}} = 2\pi\delta\omega\sigma_z \quad (5.12a)$$

$$H_c = (1 + \gamma)2\pi[\omega_x\sigma_x + \omega_y\sigma_y], \quad (5.12b)$$

where $\delta\omega$ is the static field offset, γ is the control field inhomogeneity, and where ω_x and ω_y are the control fields produced by the distortion of a given pulse.

In Figure 5.4, we show an example of finding a 99.99% fidelity $\pi/2)_x$ pulse for the nonlinear resonator described by the parameters in Table 5.1, and for the Hamiltonian above, taking $\gamma = \delta\omega = 0$. In the design of this pulse, we have included the ringdown compensation heuristic of Section 5.2.4.

Table 5.1: Values of parameters as defined by (5.9) for designing example pulses.

$L = 100 \text{ pH}$	$(1 + \alpha_L I_L ^2)$
$R = 0.01 \Omega$	$(1 + \alpha_R I_L ^{\eta_R})$
$R_L = 50 \Omega$	
$C_t = 2.49821 \text{ pF}$	
$C_m = 3.58224 \text{ fF}$	
$\alpha_L = 0.05 \text{ A}^{-2}$	
$\alpha_R = 0.001 \text{ A}^{-2}$	
$\eta_R = 0.7$	
$\omega_0 = 10.0622 \text{ GHz}$	

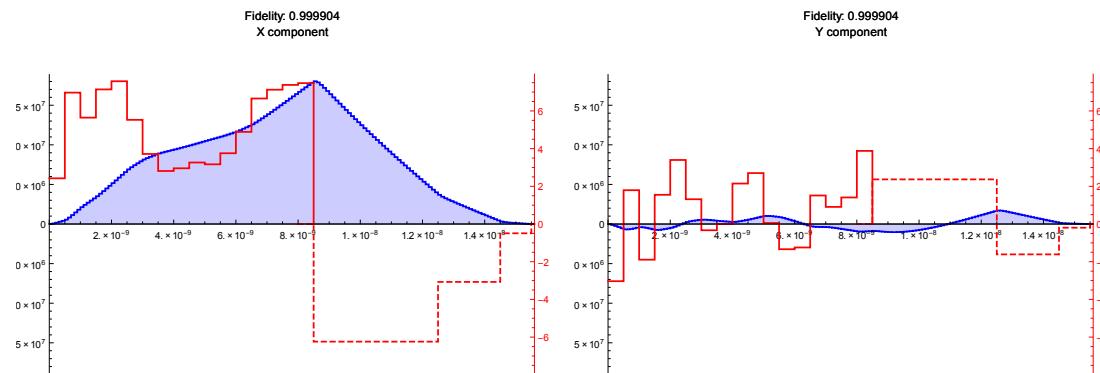


Figure 5.4: Example pulse implementing $\pi/4)_x$ for a nonlinear resonator. The driving pulse is shown in red, while the output pulse is shown in blue. The ringdown compensation steps are shown as dashed.

5.2.3 Robustness to Distributions

Importantly, we want to make pulses that not only account for classical control hardware, but that do so in a way that is *robust* to uncertainty either in the Hamiltonians that govern the system being controlled, or in the distortions that act on control sequences. Such uncertainty can arise either from a finite error in an earlier characterization procedure (Chapter 2), or due to an intrinsic property of the system being controlled. For instance, static and control field inhomogeneities in ensemble systems imply that each member of an ensemble experiences a different unitary in response to the same control sequence. In either case, we consider that the map $U(p)$ from control sequences to unitary operators is conditioned on a hypothesis, represented as a vector \underline{x} of model parameters. This is in direct analogy to the representation of posterior distributions that we adopted in considering the sequential Monte Carlo algorithm (Section 2.2).

In order to account for uncertainty in this hypothesis vector, we replace the objective function by the expectation of the *conditional objective* over the uncertainty in \underline{x} ,

$$o(\underline{p}) = \mathbb{E}_{\underline{x}}[o(\underline{p}|\underline{x})] = \mathbb{E}_{\underline{x}}[\Phi(U_{\text{targ}}, U[\underline{p}; \underline{x}])]. \quad (5.13)$$

As the expectation is a linear operator, we can thus evaluate the new definition of the marginalized objective $o(\underline{p})$ be computed by repeated evaluations of Φ for various hypotheses \underline{x} .

This technique has previously been used to find pulses that are robust to wide distributions arising from control and static field inhomogeneities [210; 218; 219]. Considering the marginalized objective is especially convenient for use with gradient methods such as GRAPE, as the gradient and expectation operators commute, such that gradients can similarly be computed by Monte Carlo expectations over the uncertainty being considered.

In Figure 5.5, we show an example of a pulse designed to be robust across a range of values of the inhomogeneity parameters $\delta\omega$ and γ . In designing the pulse, we started with a δ -distribution at $\delta\omega = \gamma = 0$ and then iteratively broadened the distribution in five steps, using the previous solution as an initial guess to each later iteration. Doing so, we obtain a pulse that is broad across 5% control inhomogeneity and 500 kHz static inhomogeneity.

We note, however, that there are fundamental limits to the applicability of marginalized objectives [220]. In practice, as the distributions being considered are made more broad, more initial guesses must be made as to pulse solutions, such that the computational cost of implementing marginalized-objective OCT methods increases accordingly.

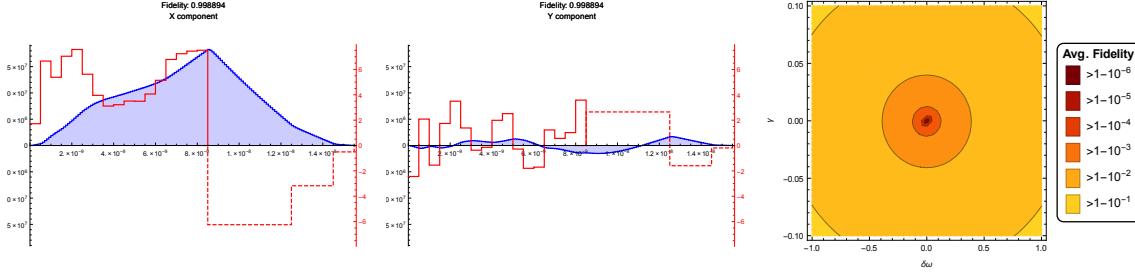


Figure 5.5: Example of a robust pulse for the nonlinear resonator. (Left and middle) Shape of the pulse, with input shown in red, ringdown compensation shown in red dashes and the distorted pulse shown in blue. (Right) Average fidelity contours over $\delta\omega/\text{MHz}$ and γ .

In [Section 6.1.3](#), we will use a genetic algorithm approach to formalize the use of many initial guesses into a *population*.

5.2.4 Heuristics for Pulse Desiderata

Using GRAPE, we can also account for additional heuristics within each iteration, such that further properties can be demanded of solution pulses. As an example of such heuristics, we consider the problem of ensuring that after driving a resonator with a large quality factor $Q \gg 1$, the energy stored in the resonator is effectively removed. That is, we want to compensate for the effect of *ringdown* seen in examples such as [Figure 5.4](#). Explicit optimization for ringdown compensation can be included as a secondary search within each GRAPE iteration if circuit simulation is relatively inexpensive [210], but here, we provide an explicit heuristic that iteratively and approximately solves ringdown compensation using significantly fewer circuit simulation calls.

In the case of a linear distortion $g = a_g e^{t/\tau_c} \star$ acting independently on each control knob, it is straightforward to compensate for ringdown by appending a new time step $K + 1$ of width δt and amplitude

$$p_{n,K+1} = -a_g \frac{g[p]_n}{e^{\delta t/\tau_c} - 1}. \quad (5.14)$$

This is frustrated, however, when g is nonlinear, as this implies that a_g and τ_c are no longer constant. Instead, we can generalize by demanding not that the compensation

step takes the distorted pulse amplitude to zero, but to some fraction r of its initial value. In this way, during each individual compensation step, treating the ringdown parameters as constant serves as a better approximation.

Using the example of the nonlinear resonator discussed in [Section 5.2.2](#), we wish to derive for a single iteration of ringdown compensation the complex-valued pulse amplitude \tilde{p}_{rd} that we should append to reduce the magnitude of the distorted pulse by a factor of $r \in [0, 1]$. Note that since we have been considering real-valued controls, this is effectively a two-control problem.

Formally, we write [\(5.9\)](#) as a matrix equation in the state parameters $\underline{y} = (I_L, V_{C_m}, V_{C_t})$,

$$\dot{\underline{y}} = \underline{\underline{A}}(\underline{y})\underline{y} + \alpha\underline{b}, \quad (5.15)$$

where $\alpha = \alpha(t)$ is a control proportional to $V_s(t)$, and where $\underline{\underline{A}}$ and \underline{b} are defined by [\(5.9\)](#). Then, finding \underline{y}_0 immediately before a ringdown compensation step from the full numerical integration of this differential equation, we can find the compensation pulse explicitly. In the limit $\tau_c \gg \delta t$, this objective is minimized by [\[39\]](#),

$$\tilde{p}_{\text{rd}} = \frac{\langle \underline{v}, \underline{w} \rangle}{\langle \underline{v}, \underline{v} \rangle}, \text{ where} \quad (5.16a)$$

$$\underline{v} = \underline{\underline{P}}(\mathbf{e}^{\underline{\underline{t}}\underline{\underline{A}}} - r\mathbb{1})\underline{y}_0 \quad (5.16b)$$

$$\underline{w} = -\underline{\underline{P}}\underline{\underline{A}}^{-1}(\mathbf{e}^{\underline{\underline{t}}\underline{\underline{A}}} - r\mathbb{1})\underline{b}, \quad (5.16c)$$

where $\underline{\underline{P}} \geq 0$ weights the importances of minimizing each state variable being considered.

5.3 Composite Control Design for Node-Based Architectures

Many different proposals have been advanced for partitioning quantum information processing devices into distinct *nodes*, for example, [\[221; 22; 222; 223; 224; 225; 166; 226\]](#). An architecture of this form allows for many distinct advantages, most notably when couplings between nodes can be turned on and off either directly, or by the application of decoupling sequences [\[227\]](#) such as CPMG [\[228\]](#) or WAHUHA [\[229\]](#). In this case, the predominance of control can be designed in a Hilbert space limited to the size of a single node, then applied in parallel *across* nodes.

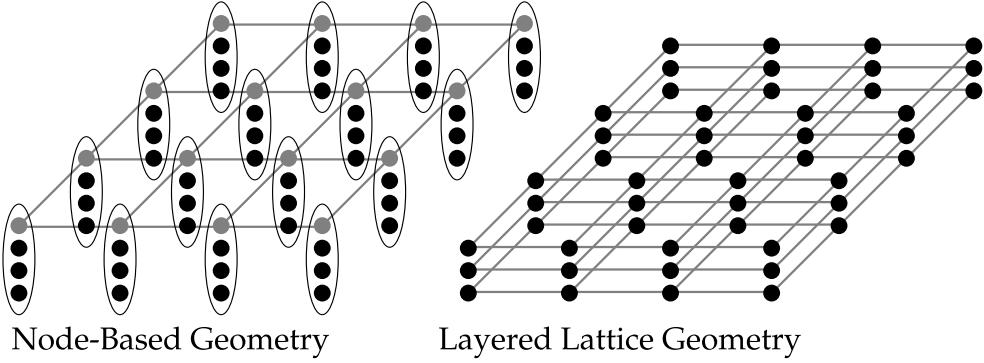


Figure 5.6: Geometry for layered lattice or sublattice codes, using node-based geometries. The intranodal index runs vertically, addressing the individual parts of each node, while the lattice is over the two spatial coordinates. The elements corresponding to a fixed index within each node form a single lattice or sublattice code, and transversal gates across “layers” can be implemented via parallel operations within each node. A composite pulse sequence affecting the interactions on the right for the electron/nuclear example is given in [Section 5.3.1](#).

Node-based architectures are thus especially appealing for fault-tolerant schemes where the geometry of the error correction code applied admits a redundant structure. For instance, error correction codes based on lattice surgery [230; 231] can then be implemented between nodes, rather than between spins, such that many copies of the code may be implemented at once [226]. Moreover, because sublattice surface codes such as those employed by lattice surgery admit a transversal CNOT gate, these independent sublattice codes can then be acted upon as the concatenation of a topological and a block code.

With this in mind, we are thus interested in finding unitary operations that act transversally between nodes, such that we can build the error-correction primitives for sublattice codes in a way that easily parallelizes over the logical index (that is, an index that addresses individual components of a node). Our approach will be to assume that each computation element of a node couples strongly to an *actuator*, such that only the actuators couple between different nodes. Here, an actuator is taken to be a component of a quantum system that couples both to the quantum register of interest and to the environment, such that control on the register can be implemented with control on the actuator alone [203]. [Figure 5.7](#) illustrates this for the example where the actuator for each node is an electron spin, and where each computational element is a nuclear spin;

the nuclear spins are each coupled to the neighboring electrons by a hyperfine coupling, while the different electrons are coupled by a dipolar coupling.

Our goal is thus twofold: first, we must engineer couplings directly between the corresponding computational elements. Second, we must then use these couplings to engineer couplings in parallel between the registers attached to each node. In the context of the example given in [Figure 5.7](#), we thus want to engineer 2-body couplings that act between $n_{1,i}$ and $n_{2,i}$ for each $i \in \{1, 2, 3\}$. To mitigate the costs of design in large Hilbert spaces, we will build these engineered couplings using *composite pulses* of gates that each act in a smaller space.

5.3.1 Wide Quantum Channels

As a motivating example, we will focus on the system demonstrated in [Figure 5.7](#), where the Hamiltonian is given by

$$H = H_D^{e-e} + H_{hf}^{e-n}, \quad (5.17a)$$

and where

$$H_D^{e-e} = \omega_D (2\sigma_z^{e_1}\sigma_z^{e_2} - \sigma_x^{e_1}\sigma_y^{e_2} - \sigma_x^{e_1}\sigma_y^{e_2}) \quad (5.17b)$$

$$H_{hf}^{e-n} = \sum_{k=1}^K \underline{A}_k \cdot (\sigma^{e_1}\underline{\sigma}^{n_{1,k}} + \sigma^{e_2}\underline{\sigma}^{n_{2,k}}), \quad (5.17c)$$

where each electron is assumed to address K nuclear spins by distinct hyperfine vectors \underline{A}_k .

Note that in our model, we have only kept the secular part of the electron-electron dipolar Hamiltonian, under the assumption of a large Zeeman interaction on each electron. This model admits universal control over the nuclear spins by pulses on the electron spins, provided each hyperfine vector \underline{A}_k is spectroscopically resolvable, anisotropic [[203](#)] and is well-characterized ([Section 2.3.3](#)).

The electron/nuclear example is an ideal prototype for more general actuator-based nodal architectures, as it admits a hierarchy between the strong intranodal couplings, the weaker internodal actuator couplings, and the negligible couplings between internodal computational elements.

In particular, since we have universal control via the argument of Hodges [[203](#)], an effective nuclear-nuclear coupling must exist in the Lie algebra of the controllable Hamiltonian terms. This then implies that we can find a commutator sequence that implements

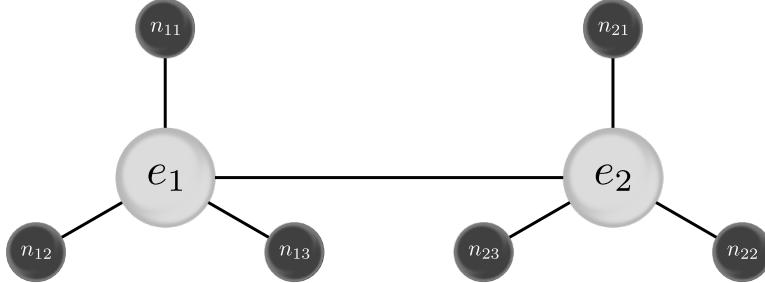


Figure 5.7: Example of two nodes in an e/n node-based architecture.

the desired interactions without explicitly storing quantum information in the state of the electron spins. Explicit algorithms have been provided that can simulate exponentials of arbitrary commutators, and with arbitrary orders of error cancellation [232], such that finding the commutator that produces the desired interaction is itself constructive for providing a composite pulse sequence. Though the aforementioned algorithm is optimal in the asymptotic sense, we will show a decomposition for our commutator that admits a better constant factor in this case.

As a guide to finding a commutator that produces the desired interaction, we consider the path through Hilbert space that an effective nuclear-nuclear interaction must take. The state manifold for this system in the example of a single nuclear spin per node is shown in Figure 5.8. Whereas the hyperfine states are coupled by the electron dipolar, we can affect a useful operation in the electron zero-quantum manifold by a combination of the hyperfine and dipolar couplings. We can also think of the desired zero-quantum manifold interaction as a continuous-time limit of an explicit swap protocol, such that we expect to find it at second order in $H_{\text{hf}}^{\text{e-n}}$ and first order in $H_D^{\text{e-e}}$.

Using this intuition, we look at second-order in the adjoint of $H_{\text{hf}}^{\text{e-n}}$, $\text{ad}(H_{\text{hf}}^{\text{e-n}}) = [H_{\text{hf}}^{\text{e-n}}, \cdot]$ for the desired interaction. We find that

$$[[H_D^{\text{e-e}}, H_{\text{hf}}^{\text{e-n}}], H_{\text{hf}}^{\text{e-n}}] \propto \omega_D (\sigma_+^{\text{e}_1} \sigma_-^{\text{e}_2} + \sigma_+^{\text{e}_1} \sigma_-^{\text{e}_2}) \otimes H_D^{\text{n-n}}, \quad (5.18)$$

where $H_D^{\text{n-n}}$ is a dipolar interaction on all edges of the complete (K, K) -bipartite graph, affected by the electron-electron interaction. We can then generate this Hamiltonian as the effective Hamiltonian for a composite sequence by recursively applying that $e^X e^Y = \exp(X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}[X, [X, Y]] + \dots)$ to obtain the identity

$$e^X e^Y e^{-X} e^{-Y} e^{-X} e^Y e^X e^{-Y} = e^{[X, [X, Y]] + \dots}. \quad (5.19)$$

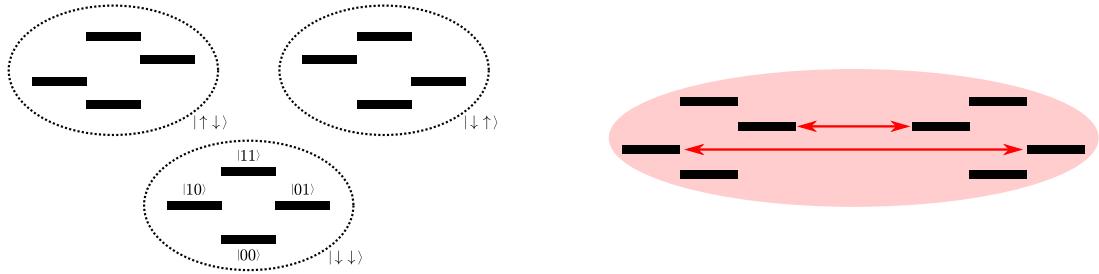


Figure 5.8: (Left) Labeling of electron (\uparrow/\downarrow) and nuclear (0/1) spin states, shifted by hyperfine couplings. (Right) Mixing of hyperfine levels in the electron zero-quantum manifold $\text{span}(|\uparrow\downarrow\rangle, |\downarrow,\uparrow\rangle)$ by dipolar coupling.

Letting $X = -iH_{hf}^{e-n} \delta t$ and $Y = -iH_D^{e-e} \delta t$, we identify this sequence as that which produces the wide quantum channel Hamiltonian (5.18) for the electron-nuclear node-based architecture. Here, the truncation is to lowest-order; generalizations for arbitrary order in δt and a_d are given by [232], such that more elaborate wide quantum channel Hamiltonians can be implemented.

To isolate the internodal couplings between correspondingly labeled nuclear spins, instead of the complete-bipartite graph, we require one more interaction frame in our composite pulse sequence. As illustrated in Figure 5.10, we want to suppress couplings between $n_{1,k}$ and $n_{2,k'}$ for $k \neq k'$, to affect control useful for such applications as transversal gates between layers of a lattice code. This is achieved by using an interaction frame defined by the Hadamard-Walsh basis [233], such that the sign of A_k is orthogonal to $A_{k'}$ unless $k = k'$. The Hadamard-Walsh basis, illustrated in Figure 5.9, is a common digital signal processing technique that has been previously used to characterize temporal profiles of magnetic fields in quantum systems [169].

Our technique thus provides a way of affecting useful transversal intermodal couplings in parallel, in turn enabling novel new architectures to be explored. By explicitly providing a set of composite pulse sequences that realizes these interactions, we also enable the design of control in smaller Hilbert spaces, namely that each gate operates either in parallel across nodes, or acts locally within each node.

In combination with optimal control methods for designing control in each of these smaller Hilbert spaces, the wide quantum channel construction thus provides a very important step towards developing control in large systems. In particular, these techniques together are especially useful in node-based actuator-controlled architectures, providing a natural path to implementing quantum error correction using lattice codes in parallel

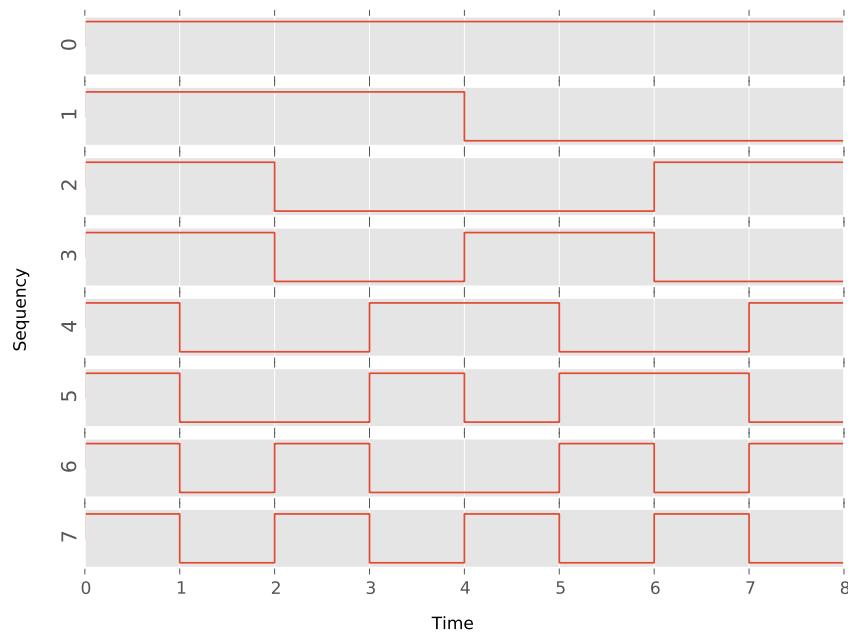


Figure 5.9: Walsh-Hadamard functions on 8 dimensions, ranked in increasing ``seqency'' order. Source code provided in [Listing C.7](#). Interpreted as an interaction frame for wide quantum channel internodal decoupling, these sequences eliminate crosstalk between mismatched nuclear labels at zeroth-order in Magnus for up to eight nuclear spins.

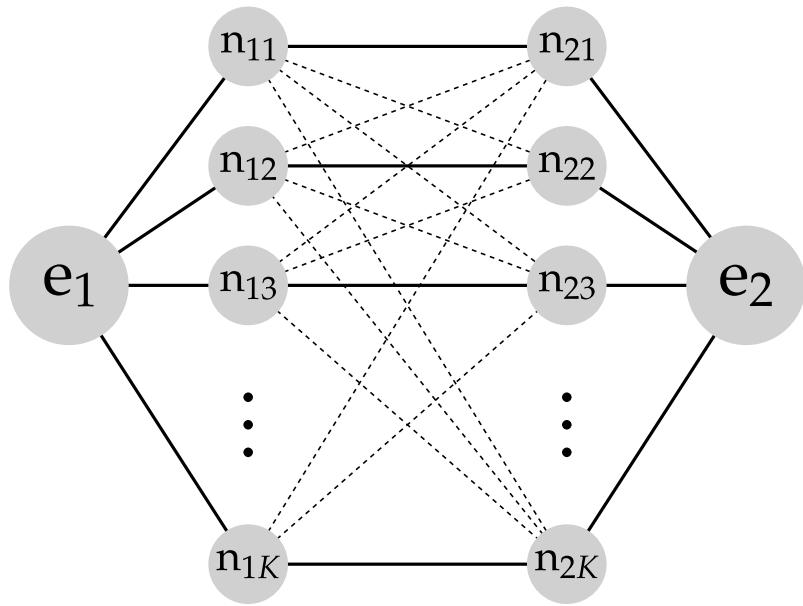


Figure 5.10: Suppression of cross couplings in a wide quantum channel.

across intranodal labels.

6 Quantum Coprocessors and Semiquantum Algorithms

Many of the approaches discussed thus far require simulating the dynamics of quantum systems. Though we have proposed doing so using classical resources, this places a severe constraint on the dimensionality of the systems that we can study using simulation-based techniques such as the SMC algorithm (Section 2.2) or optimal control theory (Chapter 5) unless long-standing conjectures about the difficulty of quantum systems are shown to be false. In order to build our way towards large quantum systems, then, we must do something different, and introduce a resource other than classical computing alone. Due to the difficulty of implementing quantum algorithms using existing devices and those proposed for the near future, however, we turn our attention instead to augmenting classical computing with access to *quantum coprocessors*; that is, small quantum systems capable of simulating the behavior of other quantum systems. This can proceed either by using quantum simulation algorithms [234; 235], or by engineering a system to emulate the behavior of another quantum system [236; 237; 238; 239; 89]. Here, we shall suppose no more capability than the latter, namely that we can prepare copies of a state $|\psi\rangle$, evolve under a Hamiltonian $H(\underline{x})$ drawn from an accessible family, and then measure in some basis that includes $|\psi\rangle$.

We shall consider this additional resource in two primary contexts, first in the design of quantum control, then in the characterization of quantum systems. In the case of characterization, we then show how information locality can be used to extend our methods

to the analysis of quantum systems that are strictly larger than our quantum coprocessors. This capability in turn allows us to calibrate the controls needed to produce larger coprocessors and bootstrap our way to useful quantum information processing devices.

6.1 Control Design with Quantum Coprocessors

6.1.1 In-Place and Parallel Control Optimization Models

As an alternative to using classical resources to design quantum control sequences, we can also consider using quantum coprocessors as simulators to design control. Several methods have been introduced over the past several years that address this idea in different ways, mostly by designing control *in-place* using the ORBIT [85], Ad-HOC [86] or ACRONYM [25] algorithms. That is, a quantum device is used as a high-fidelity simulator for *itself*, such that pulses can be designed for a device of interest by observing the output of characterization experiments on that device. This provides not only the advantage of being applicable in higher-dimensional systems beyond the range of algorithms such as GRAPE (Section 5.1), but can also mitigate at least some of the need to characterize systems before designing control, such that the exact control distortions need not be known in order to design and apply control.

Conceptually, in-place OCT is similar to quantum feedback schemes [240], in which control is dynamically recalibrated to account for changes in an experimental system. In-place quantum pulse design differs, however, in that the characterization steps are treated as an oracle that is called between or before experiments of interest. In this way, existing algorithms sit somewhere between open-loop classical OCT and closed-loop control feedback, allowing for OCT to be carried out with minimal quantum space requirements.

Each of these algorithms treats evaluation of the objective function as an *oracle*, that is, a subroutine that can be called to query the fidelity objective for a given pulse. In all three of these algorithms, the objective oracle is implemented by means of randomized benchmarking [145] (Section 2.3.2), such that the costs grow quadratically with the infidelity of initial guesses [161]. The quadratic growth of the uncertainty in the objective presents an obstacle to using existing approaches in the case where no good initial guesses can be drawn without the use of quantum resources.

Finally, by restricting ourselves to in-place design algorithms, we preclude the use of *parallel* quantum resources. That is, once we take the leap to designing control sequences

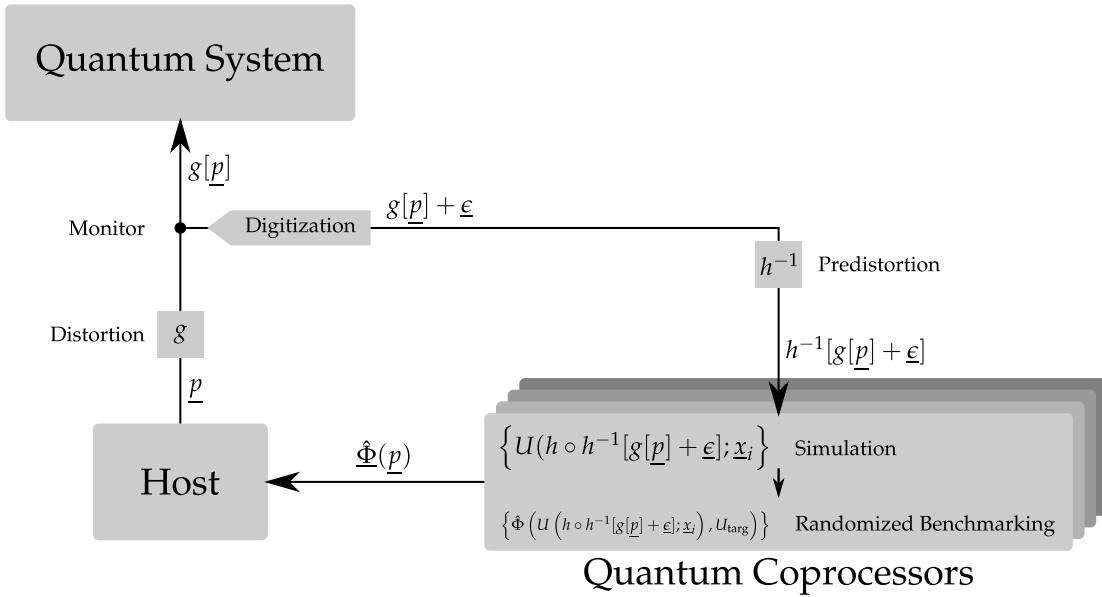


Figure 6.1: Use of parallel quantum coprocessors to evaluate pulse fidelities. A pulse p is prepared by a classical host, and then sent through a classical control system that introduces a distortion g . A monitor (such as a spy coil) is used to measure the output of this distortion, giving $g[\underline{p}]$, producing a pulse $g[\underline{p}] + \epsilon$ for some noise ϵ . The measured pulse is then pre-distorted by the inverse distortion h^{-1} for a well-characterized array of quantum coprocessors, and is simulated for each of a variety of hypotheses \underline{x}_i . Using accelerated randomized benchmarking to evaluate fidelities, we then get an estimate of the fidelity vector $\hat{\Phi}$ for the original pulse.

using quantum resources that are *external* to the system of interest, this immediately suggests the use of many small quantum coprocessors, each communicating classically with a single classical host. This configuration is illustrated for evaluating a single pulse across an ensemble of hypothesis in [Figure 6.1](#).

In light of these challenges, we therefore seek an algorithm that utilizes small quantum coprocessors in parallel to design robust pulses in a manner that is insensitive to errors introduced by finite sampling of randomized benchmarking oracles and that does not require simulation in order to design initial guesses.

6.1.2 Robustness and Multi-Objective Optimization

Before we develop our algorithm, however, we first consider the challenge posed to quantum coprocessor-assisted control optimization by the geometry of the optimal control theory problem. For a control pulse $\underline{\underline{p}}$ to be robust, we demand that it admits a high fidelity for each hypothesis \underline{x} about the quantum system of interest. We can model this by considering the fidelity as a vector-valued objective function

$$\underline{o}(\underline{\underline{p}}) : i \mapsto \Phi(U_{\text{targ}}, U[\underline{\underline{p}}; \underline{x}_i]) \quad (6.1)$$

that specifies the fidelities at each individual hypothesis. Similar approaches have been used to extend optimal control theory for quantum states to the multiobservable case [241; 242], and to quantum system discrimination [243]. By contrast, we are concerned here with the multihypothesis case for designing quantum operations. In considering the objective function as vector-valued, we are making a different demand than a scalar-valued objective function formed by taking an expectation over hypotheses, as we used in [Section 5.2.3](#).

It is not now immediately clear what it means for a given feasible solution to be optimal, as a tradeoff between the different components of \underline{o} cannot be made unambiguously. Instead of maximizing according to the total ordering \geq , then, in the multiobjective case, it is common to use instead the *partial* ordering given by Pareto optimality [242]. In particular, we say that $\underline{\underline{p}}$ **dominates** $\underline{\underline{q}}$ if $o_i(\underline{\underline{p}}) \geq o_i(\underline{\underline{q}})$ for all i . Having thus defined the domination relation, we can now define a Pareto optimal pulse to be one that is not dominated by any other pulse, such that it is extremal according to the partial ordering given by domination. Intuitively, a pulse is non-dominated if it cannot be improved in any component except at the cost of another component. Writing $\underline{\underline{q}} \blacktriangleright \underline{\underline{p}}$ to mean $\underline{\underline{q}}$ dominates $\underline{\underline{p}}$, our optimal control theory problem can now be restated in the multi-objective case as

$$\begin{aligned} &\text{given } U_{\text{targ}} \\ &\text{find } \underline{\underline{p}} \\ &\text{s.t. } \nexists \underline{\underline{q}} : \underline{\underline{q}} \blacktriangleright \underline{\underline{p}}, \forall i : o_i(\underline{\underline{p}}) \geq o_{\text{targ}}, \end{aligned} \quad (6.2)$$

where $o_i(\underline{\underline{p}})$ are the components of the vector objective $\underline{o}(\underline{\underline{p}})$.

We can consider the *surface* of Pareto optimal solutions in the space of objective function values. For instance, an octant of the 3-sphere is the Pareto surface for a trivial objective function on vectors in \mathbb{R}^3 , $\underline{o}(\underline{r}) = \underline{r}$, under the constraint that $\|\underline{r}\| \leq 1$. We can

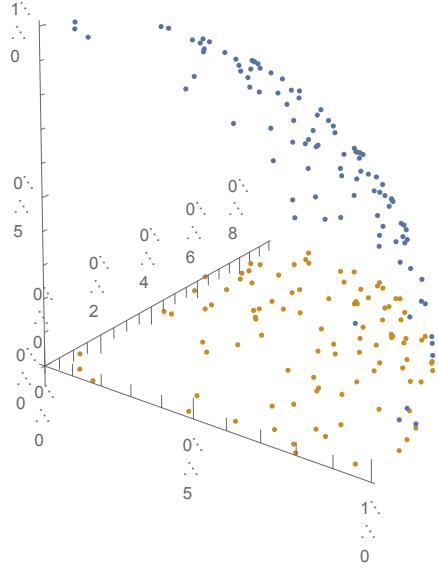


Figure 6.2: Projection onto the plane of the Pareto optimal solutions to the trivial objective $\underline{\varrho}(\underline{r}) = \underline{r}$ under the constraint $\|\underline{r}\| \leq 1$.

clearly not improve on $\underline{r} = (0, 0, 1)$ except by trading off the z component for gains in x and y . Importantly, if we project this Pareto surface onto two dimensions, we obtain solutions that appear to be interior, as they offer advantages in components not considered in that projection. This is illustrated in Figure 6.2.

Pareto optimality is a more demanding condition than that considered in Section 5.2.3, where $\underline{o} := \mathbb{E}_{\underline{x}}[\Phi]$. In particular, as we broaden the distribution over \underline{x} in this definition, we will in general introduce cusps into the control landscape, such that Pareto optimal solutions cannot be found by expectation values alone [220]. This limitation can arise from nonconvexity in the Pareto optimal surface, such that expectation values alone will miss Pareto optimal solutions. In order to explore the set of nondominated solutions, we must therefore move to a multi-objective control design algorithm that directly solves (6.2).

Importantly, however, we cannot naïvely use a subgradient descent method (SGD) for this problem, as $\underline{\varrho}$ is not convex in the case of the fidelity objective $\underline{\varrho} = \Phi$. Algorithms such as the goal-attainment method can solve this by reducing a multiobjective problem to a single objective problem with a minimization over a slack variable [244].

Moreover, generalizations of SGD methods for nonconvex, nondifferentiable functions such as $o_{\min} := \min_i o_i$ have been demonstrated [245], allowing the direct use of gradient information. Here, however, we take a different approach by instead focusing on multi-objective evolutionary algorithms (MOEAs).

6.1.3 Multi-Objective Memetic Algorithms

Genetic and evolutionary algorithms offer a promising paradigm for solving the problems that we have considered thus far for several reasons. First, by using *tournaments* that randomly select between components of a multiobjective fitness, evolutionary algorithms (EAs) can be made manifestly multiobjective [246]. That is, we can design multi-objective EAs (MOEAs) by selecting individuals for reproduction that are non-dominated [247]. Second, EAs need not include explicit gradient information, relying instead on parallel evaluation of the fitness vectors for each member of a large population. Our approach thus does not break when considering a nonconvex objective vector such as \underline{o} . Third, we can readily extend EAs to include *local improvement*, forming multi-objective *memetic* algorithms (MOMAs) instead [248]. Here, we use the SPSA algorithm [249] employed by ACRONYM [25] to perform local improvements on individual components of randomly selected individuals [250], such that we require gradient estimation only of a single- objective function of the underlying multi-objective fidelity function. The use of a local improvement step allows us to rapidly expand the set of pulses that are not dominated by any other pulse yet discovered, known as the *Pareto front*. In particular, by focusing on those components that restrict the Pareto front, local improvement can dramatically reduce the number of generations required to find solutions that are approximately Pareto optimal.

Putting these together, we form our memtic optimal quantum-accelerated control algorithm (MOQCA) for QCP-assisted pulse design. In this Section, we describe MOQCA, detail an implementation in Python based on the DEAP library [251], and then show a simple example of applying MOQCA to design pulses that account for ringdown and static field inhomogeneities. The inclusion of randomized benchmarking sampling errors is left to future work.

6.1.3.1 Genetic Algorithm Definitions

In formulating and specifying MOQCA, it is helpful to first build some terminology connecting our problem of interest to that used in genetic algorithms. In the interest of generality, we do not give specific values for any parameters defined in this Section,

but instead lay out how MOQCA can be tuned and configured. We will later present an example, using the specific values given in [Table 6.1](#).

Firstly, we define an *individual* to be a sequence of pulse *attributes* together with a *strategy*. Each such attribute $(\Delta t_k, p_k)$ specifies the width of a pulse timestep and the values of each control field at that timestep. The strategy $\underline{s} = (\sigma_p, a, b, s, t)$ encodes the pulse amplitude mutation strength σ_p and the SPSA parameters a, b, s, t described in [Algorithm 5](#), such that the strategy can coevolve with the pulses.

Algorithm 5 SPSA algorithm for local improvement.

Input: Pulse \underline{p} , fitness function \underline{f} , key function k , real parameters a, b, s, t , number of iterations n .

Output: Optimized pulse \underline{q} .

```

function SPSAIMPROVE( $\underline{p}, \underline{f}, k, a, b, s, t, n$ )
    for  $i \in 1 \rightarrow n$  do
         $\alpha \leftarrow a/(i+1)^s$ 
         $\beta \leftarrow b/(i+1)^t$ 
         $\underline{\Delta} \leftarrow$  random matrix of  $\pm 1$  entries, same size as  $\underline{p}$ 
         $\underline{g} \leftarrow [k(\underline{f}(\underline{g} + \beta \underline{\Delta}) - k(\underline{f}(\underline{g} - \beta \underline{\Delta}))]/2\beta$ 
         $\underline{q} \leftarrow \underline{p} + \alpha \underline{g} \underline{\Delta}$ 
        if  $\underline{q}$  is worse than  $\underline{p}$  under  $k \circ \underline{f}$  then
            return  $\underline{p}$ 
        else
             $\underline{p} \leftarrow \underline{q}$ 
        end if
    end for
    return  $\underline{p}$ 
end function
```

Individuals are each assigned a *fitness vector* \underline{f} by interpreting the fidelity objective vector $\underline{o}(p)$ as an evolutionary fitness. The fitness of each individual is evaluated by performing interleaved randomized benchmarking with the pulse associated with each individual, and for each of a list of hypotheses about the system of interest. This introduces a sampling error described either as in [Section 2.3.2](#), for strong measurement, or as by Wallman and Flammia [\[161\]](#) for ensemble measurement. Here, we do not include this error explicitly, leaving this to future work. In evaluating each pulse, we also

apply distortions as discussed in [Section 5.2.1](#), with additional Gaussian noise added at each timestep to represent imperfect measurement of the pulse produced. Notably, simulation-free heuristics such as ringdown compensation (see [Section 5.2.4](#)) can be included as distortions, such that MOQCA need not directly integrate knowledge of how energy in a resonator is removed at the end of a pulse.

At the start of each iteration, a random subset of individuals is chosen for memetic improvement using SPSA. Since SPSA is a single-objective algorithm, for each individual undergoing improvement, we first select a scalar function $k(\underline{f}(I))$ of the individual fitness vectors $\underline{f}(I)$ to be improved. Here, we consider three such *key functions*, $k_{\min}(\underline{f}) = \min \underline{f}$, $k_{\mathbb{E}}(\underline{f}) = \sum \underline{f} / \dim \underline{f}$ and $k_i(\underline{f}) = f_i$ for a random component i . For each individual, we draw k from $\{\bar{k}_{\mathbb{E}}, k_1, \dots, k_{\dim \underline{f}}\}$ and perform a predetermined number of SPSA iterations, using the parameters given by the individual's strategy. The probabilities of each key function are varied as a function of how many generations have been performed so far, such that more aggressive improvement can be used later in the algorithm's execution.

Following improvement, another random subset of individuals is chosen for mutation. Once chosen, a random subset of an individual's attributes are then chosen to be perturbed by a normal random variate drawn with zero mean and standard deviation given by the strategy parameter σ_p . With small probabilities, an individual undergoing mutation may have a new attribute inserted at a random index, or may have an attribute removed. Finally, each strategy parameter is redrawn from a log-normal distribution centered on the original value, and with scale given by the *learning rate* $\sigma = 1/\sqrt{\dim p}$. The SPSA a and b parameters are then restricted to lie within predetermined bounds.

Once improvement and mutation have been completed, individuals are selected for reproduction by a sequence of *tournaments*. In each tournament, three individuals are chosen at random, of which one is added to the mating pool. A number of tournaments equal to the previous population size is performed, such that the number of individuals remains constant. Selection is always performed using k_{\min} , such that the population at the next generation will consist of those pulses whose worst-case performance is best.

Reproduction proceeds by selecting pairs from the mating pool and performing a crossover of their respective attributes with a given probability. In MOQCA, we use the two-point crossover illustrated in [Figure 6.3](#), padding the shorter pulse by zeros if needed. After crossover, the resulting pulses form the new population that proceeds into the next iteration.

At this point, the fidelities of the new generation are evaluated and compared to

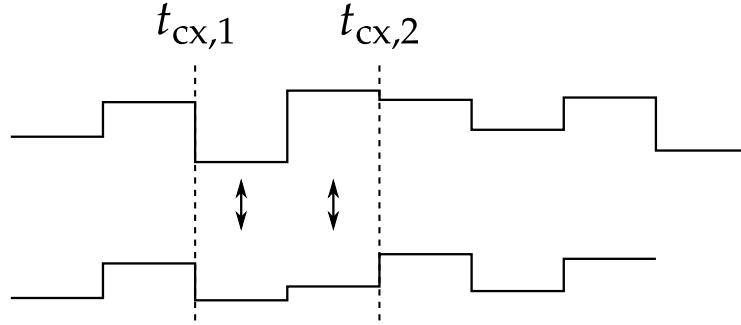


Figure 6.3: Two-point crossover operator between two distinct pulse individuals, with randomly chosen cut points $t_{\text{cx},1}$ and $t_{\text{cx},2}$. All pulse steps between the cut points are swapped between the two individuals.

previously recorded fidelities to form an online estimate of the Pareto front. Previous members of the Pareto front are checked to see if any members of the new population now dominate them, and if so, are removed from the Pareto front. Once the new generation has been evaluated, no further simulation is needed to compare to the recorded Pareto front. Thus, examination of the current non-dominated individuals can be used as a stopping criterion.

6.1.4 Single-Qubit Example of MOQCA

In this Section, we detail an example of applying MOQCA to design a single-qubit gate for a linear resonator with a significant quality factor. As discussed above, we do not assume that the distortion g acting on each individual is accurately measured, but rather that we incur additional white noise, such that the objective estimates are $\hat{\varrho}(p) = \hat{\Phi}(g[p] + \underline{\epsilon})$ for some noise $\underline{\epsilon}$. Our implementation is thus severely hampered in an experimentally-relevant way; though the ideal case is straightforward for existing algorithms, we show here that MOQCA continues to learn good pulses even with significant additive white noise. Additionally, we consider five different hypothesis about the static field inhomogeneity of the internal Hamiltonian, such that we obtain a five-dimensional objective space.

Using the parameters in [Table 6.1](#), we optimized a population of 200 individuals. At each step, we simulated objectives without any additional error to compare the estimated performance of the algorithm to what would be observed in practice. In par-

Table 6.1: Parameters used in single-qubit MOQCA demonstration.

Parameter	Value
Target	$U_{\text{targ}} = \sigma_x$
Internal Hamiltonian	$H_0(\delta\omega) = \delta\omega\sigma_z, \delta\omega \in \{0, \pm 50\text{kHz}, \pm 100\text{kHz}\}$
Control Hamiltonians	$\underline{H}_c = (\sigma_x, \sigma_y)$
Control Distortion	Exponential kernel w/ time constant $\tau = 2\text{ns}$, control rotation by $\pi/12$
Pulse Measurement Error	Gaussian white noise, amplitude $2\pi \cdot 100\text{kHz}$ per step
Ringdown Compensation	One step, 2 ns
Pulse Step Width	10 ns
Initial Pulse Lengths	70 to 100 ns
Allowable Control Range	$[-7, +7] \times 2\pi \cdot 10\text{MHz}$
Generations	200
Population Size	140
Crossover Probability	50%
Mutation Probability	20%
Attribute Mutation	20% probability each attribute, std. dev. σ_p
Attribute Creation	0.2% probability
Attribute Deletion	0.15% probability
Improvement Probability	10%
SPSA Iteration Count	2 iterations / improvement $r^2 + 10\%(1 - r^2)$ random component,
Improvement Criterion Probabilities	$90\%(1 - r^2)$ worst fidelity
Ringdown Compensation	1 ns
Initial Strategy	$\sigma_p = 2\pi \cdot 100\text{kHz}, a = 1, b = 0.05, s = 0.602, t = 0.101$
Strategy Parameter Bounds	$a \in (0, 2], b \in (0, 1]$

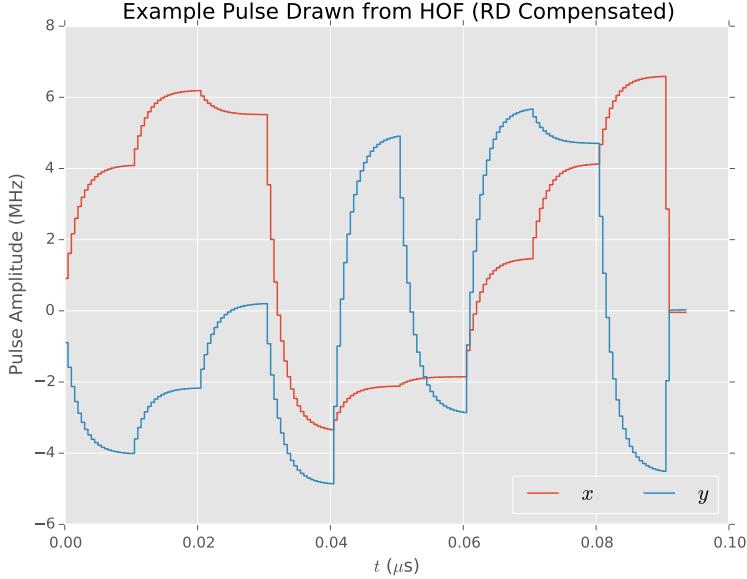


Figure 6.4: Example pulse generated MOCQA with 200 generations, a population of 140 pulses each no more than 100 ns in length, and using the MOQCA parameters in [Table 6.1](#).

ticular, we used estimated (noisily-evaluated) objectives to build a *hall of fame* (HOF) of non-dominated individuals at each step, then exactly analyzed the performance of the HOF to gather performance statistics. An example pulse from the final hall of fame, fed through the noiselessly-observed distortion, is shown in [Figure 6.4](#). In [Figure 6.5](#), we show the size of the HOF at each generation; drops in the number of nondominated individuals correspond to the discovery of newly dominating individuals, such that existing individuals are purged from the HOF.

In [Figure 6.6](#), we track $\mathbb{E}_{\text{pop}}[\hat{\varrho}]$, the mean over the population of the estimated infidelities, with a 99% goal shown for comparison. We show in [Figure 6.7](#) that at the end of the 200 MOQCA generations, we are left with a HOF whose *true* infidelities are well below the goal, such that randomly sampling a nondominated individual from the final Pareto front is highly likely to give an acceptable solution.

Finally, in [Figure 6.8](#), we consider the shape of the final Pareto front by projecting the 5-dimensional objective space into 10 different two-dimensional subspaces, each cor-

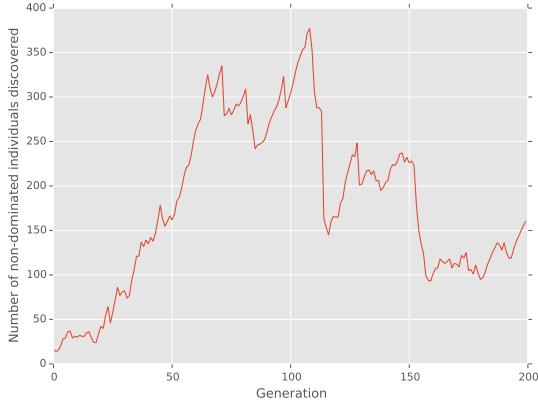


Figure 6.5: Number of non-dominated individuals found by the MOQCA algorithm as a function of generations.

responding to a pair of hypotheses about the static field inhomogeneity. These plots are similar in style to those used by [248] to visualize 3-dimensional objective space. Notably, because of the geometry of the Pareto front in the full 5-dimensional objective space, extremal points can appear to be interior when projected onto 2-dimensional subspaces. Those Pareto front individuals meeting the 99% goal are highlighted. For comparison, the *estimated* infidelities of the final population are superimposed.

In total, this example required approximately 100,000 calls to the fidelity oracle to reach the 99% goal across the ensemble of hypotheses. Though this is obviously not yet practical, we discuss improvements in [Section 6.1.5](#) that can address this gap.

6.1.5 Future Improvements

In this Section, we have introduced a new memetic algorithm, MOQCA, for the design of quantum control sequences using quantum simulation resources in parallel to evaluate objectives. Our algorithm addresses the limits of single-objective optimization by using multi-objective tournaments to find Pareto optimal pulse sequences. This effort currently suffers, however, from being expensive in terms of the quantum simulation calls required. Here, we address a few ways that future work can improve upon the algorithm as we have implemented it thus far, and in doing so render it more manifestly practical.

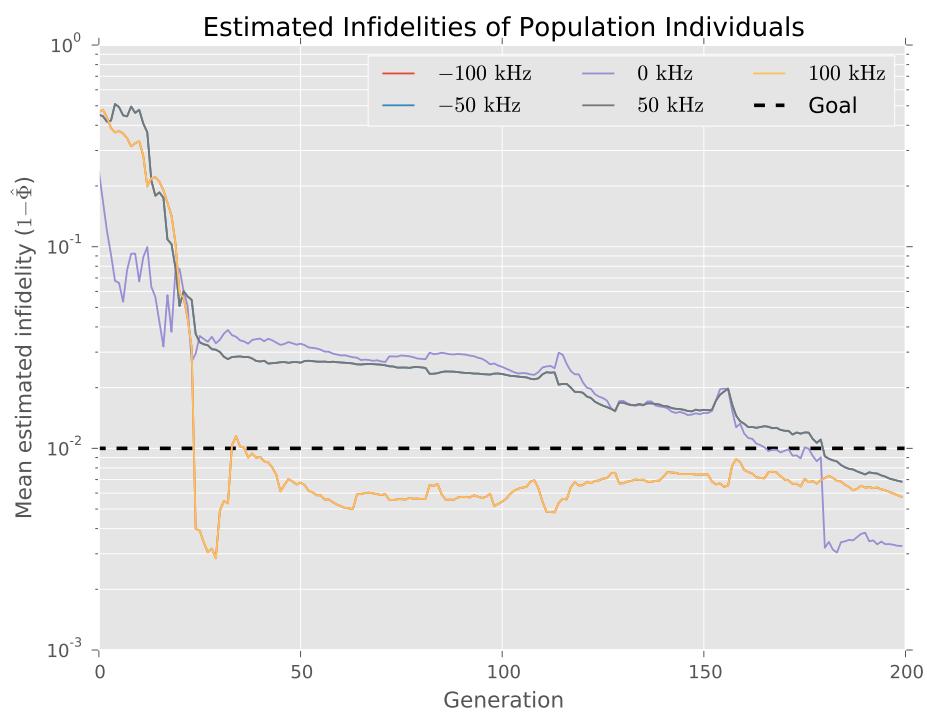


Figure 6.6: Estimated infidelities $1 - \hat{\Phi}$ for each hypothesis about H_{int} as a function of the number of MOQCA generations.

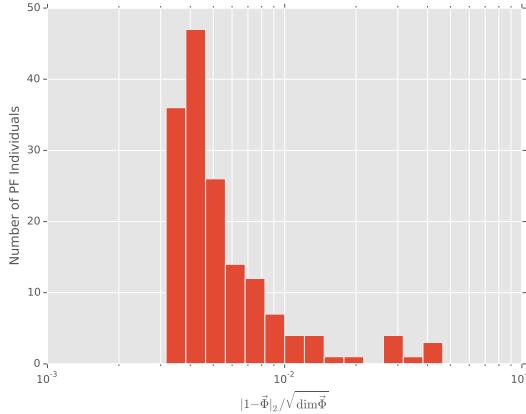


Figure 6.7: Histogram of 2-norm infidelities across the Pareto front after 60 MOQCA generations.

First and foremost, we did not account here for *lazy* evaluation of individual pulse fidelities; oracle calls are only actually required when comparing to Pareto optimality, in selection tournaments and in improvement steps. The former is the only one of these steps that requires fully evaluating the objective vector, and thus can be treated as a bottleneck of the algorithm. By contrast, selection tournaments and improvement steps comparing random components can be readily implemented without feeding through all other components. Similarly, expected performances can be calculated directly, using each hypothesis in turn in a randomized benchmarking experiment. In this way, MOQCA can be readily extended with low-level optimizations that intelligently reuse and defer fidelity evaluations.

Second, in the randomized benchmarking implementations of fidelity oracles, the prior can be made very tight around by considering the history of an individual, together with well-known bounds on how much a mutation or SPSA step can affect the fidelity. By using an inference scheme that robustly includes prior information (Section 2.3.2), the randomized benchmarking oracle can depend on this bound to dramatically reduce the number of sequences that must be implemented.

Third and finally, we note that our current MOQCA implementation uses a naïve genetic algorithm at its core. By using a more sophisticated GA after all memetic steps, MOQCA can possibly be extended to better use the geometry of the current population's objective vectors. Techniques such as domination ranking and crowding avoidance,

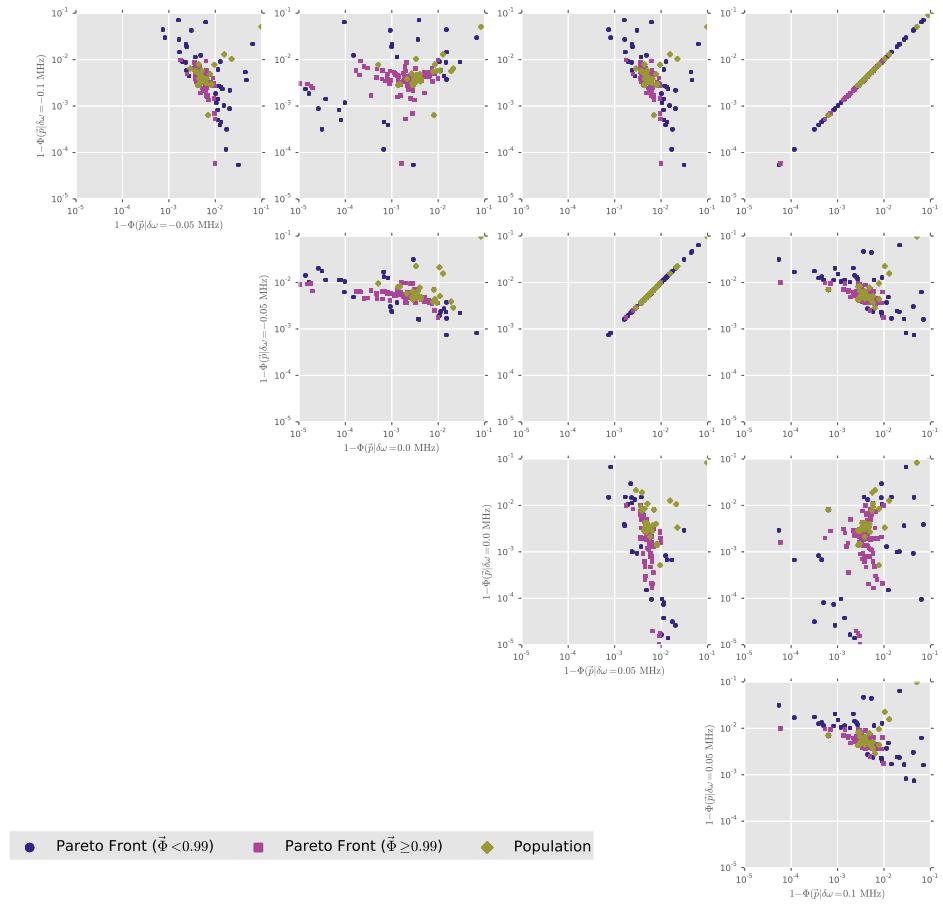


Figure 6.8: Projection of final population and Pareto front infidelities after 60 MOQCA generations onto 10 planes through infidelity space (\mathbb{R}^5). The lower-left corner of each projection is the ideal target pulse.

for instance, can possibly better distribute the genetic diversity of the MOQCA population across the Pareto front such that improvement steps in distinct individuals are less redundant with each other.

6.2 Hamiltonian Learning with Quantum Resources

In the previous Chapters, we have reduced the problems of characterization ([Chapter 2](#)) and verification ([Chapter 3](#)) to classical algorithms whose cost is dominated by the simulation of quantum systems. Since this is in general difficult under the assumption that $\text{BQP} \not\subseteq \text{P}$, these algorithms cannot thus scale with classical resources alone. On the other hand, since these classical algorithms essentially treat the quantum system being characterized, verified or controlled as a ``black box," they can be accelerated by the introduction of quantum resources.

As we have seen via the MOQCA algorithm ([Section 6.1.3](#)), this strategy is useful for addressing optimal control of quantum systems. In this Section, show how this advantage can be leveraged in the case of characterization as well, to help enable the development and verification of useful quantum information processing devices. In particular, we will show in [Section 6.4](#) that we can characterize and identify control distortions in quantum systems as large as 50 qubits using strictly smaller quantum devices as a resource, thus enabling a *bootstrapping* procedure from small quantum resources to larger devices. As quantum devices increase in scale, we therefore have a critically enabling tool for pushing to larger scales still.

Our strategy in introducing quantum resources into characterization algorithms will be to again focus on the sequential Monte Carlo algorithm ([Section 2.2](#)), building further on the insight that Bayes' rule connects *simulation* to *learning*. Since SMC already treats simulation as a black box, the algorithm is ideal for augmentation with quantum simulation.

More formally, in our description of the sequential Monte Carlo algorithm thus far, we have reduced characterization to the classical simulation of quantum systems. SMC takes as input a description of the quantum system under study, represented as a *function* $\Pr(d|\underline{x}; \underline{\varrho})$, where \underline{x} are the parameters of the Hamiltonian $H = H(\underline{x})$ we would like to learn, and where $\underline{\varrho}$ is a description of the experiment to be performed. No demand is placed by SMC on *how* evaluations of this function are implemented, aside from that we must be able to pass in a list of different hypotheses $\{\underline{x}_i\}$. Quantum simulation does not in general give us the ability to precisely calculate the likelihood function at a given

hypothesis, but rather gives us the ability to sample measurements according to that likelihood. As a result, we must generalize SMC to utilize *weak simulation*, in which we instead have access to *samples* drawn from the same distribution as for a particular hypothesis. We show such a generalization in [Appendix A](#), and show how this can be applied in the Hamiltonian learning case in [Section 6.2.1](#).

Throughout, we use QInfer [44] to demonstrate Hamiltonian learning. In particular, to demonstrate learning in Ising models, we implement likelihood evaluations classically using the F2PY and fht libraries [252; 253].

6.2.1 Design of Quantum Hamiltonian Learning Experiments

Using a variation of sequential Monte Carlo based only on samples from a simulator rather than full simulations of a likelihood function (described in detail in [Appendix A](#)), we can substitute in quantum resources for classical simulators. It thus remains to design an experiment that can well-utilize this algorithm to learn Hamiltonians. We will suppose that we are interested in learning $H = H(\underline{x})$; later, we shall consider the example of a specific family of Hamiltonians, but at this stage we consider Hamiltonian families generically. Below, we consider two distinct experimental designs for learning \underline{x} .

6.2.1.1 Classical and Quantum Likelihood Evaluation

In the first experimental design that we consider, we make no additional use of quantum resources beyond performing the simulation required to compute the Bayes update. In this way, the experiment design is very similar to that considered throughout most of [Chapter 2](#): we pick an initial state $|\psi\rangle$, evolve under the true Hamiltonian $H(\underline{x}_0)$ and measure in a basis that includes $|\psi\rangle$, such that we look for a return to the initial state. This experiment, shown in [Figure 6.10](#) (left), is generic for a wide range of Hamiltonian families.

We encounter a challenge, however, in that the Loschmidt echo famously shows that in complex quantum systems, two nearly identical Hamiltonians will typically generate evolutions that diverge exponentially after a short time, before saturating at an exponentially small overlap [254]. For example, if a Hamiltonian $H_b \in \text{Herm}(\mathcal{H})$ differs from a Hamiltonian H_a by an amount that is large compared to the characteristic scale of its eigenvalue gaps and H_a has complexity that is characteristic of canonical random matrix

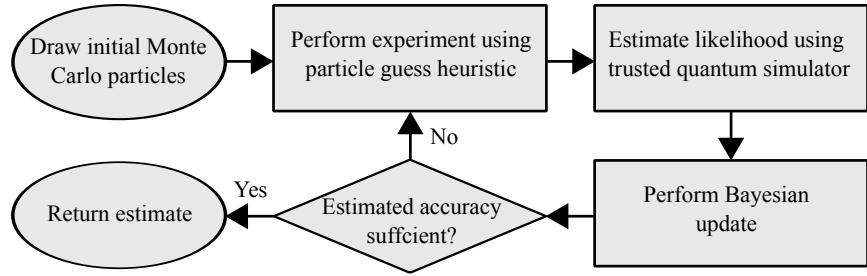


Figure 6.9: Flow chart for quantum Hamiltonian learning with interactive likelihood evaluation and particle guess heuristic.

ensembles then there exists times t_1 and t_{eq} such that [255; 256]

$$|\langle \psi | e^{iH_a t} e^{-iH_b t} |\psi \rangle|^2 \sim \begin{cases} 1 - O(t^2) & \text{if } t < t_1 \\ e^{-O(t)} & \text{if } t \in [t_1, t_{\text{eq}}] \\ 1 / \dim \mathcal{H} & \text{for most } t \geq t_{\text{eq}} \end{cases} \quad (6.3)$$

Hence, the simulated evolution and actual evolutions will vary substantially in typical QLE experiments if t is not short. This frustrates learning in two distinct ways. First, experiments with small t are uninformative as the Fisher information for any QLE experiment scales as $O(t^2 \|H\|^2)$ [34; 10], rendering short-time experiments uninformative. Second, the exponentially small likelihoods that occur for $t > t_{\text{eq}}$ cannot be efficiently estimated from frequencies on a trusted simulator, as this would require exponentially many samples to be drawn for each hypothesis.

We thus need to consider a more informative set of experiment designs in order to probe the long evolution times necessary for maintaining exponential scaling in the decay of errors.

6.2.1.2 Interactive Quantum Likelihood Evaluation

We resolve the challenges discussed in the previous section by using the trusted quantum device not just to simulate the observed data, but also to invert the evolution on the untrusted device by a hypothesis. This procedure, interactive quantum likelihood evaluation (IQLE), is reminiscent of the Hahn echo experiments commonly used in magnetic resonance and experimental quantum information processing [257]. As detailed in Figure 6.10, an IQLE experiment uses a SWAP gate to transfer a state $e^{-iH(\underline{x}_0)} |\psi\rangle$ from the untrusted device to the simulator, which then inverts to obtain the final state $e^{iH(\underline{x}_-)} e^{-iH(\underline{x}_0)} |\psi\rangle$

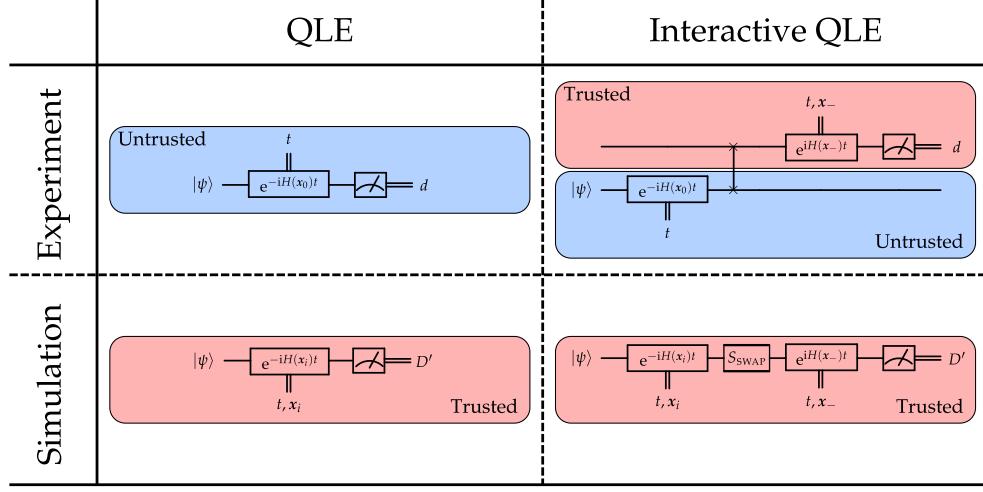


Figure 6.10: Experiment and simulator design for non-interactive and interactive quantum likelihood evaluation.

for a hypothesis \underline{x}_- about the true evolution. We assume that the measurement is in an orthonormal basis including $|\psi\rangle$, such that IQLE is seen to effectively detect the Loschmidt echo back to the original input. That is, while the Loschmidt echo may also seem to be problematic for IQLE experiments, we can take advantage of this echo to find highly informative experiments.

6.2.1.3 Particle Guess Heuristic

To design experiments for interactive quantum likelihood evaluation that take advantage of the Loschmidt echo, we use the ``particle guess heuristic'' (PGH), a simulation-free heuristic for selecting inversion hypotheses [41]. In particular, we choose the inversion Hamiltonian $H_- := H(\underline{x}_-)$ by sampling \underline{x}_- from the prior $\Pr(\underline{x})$, which describes our current knowledge of the Hamiltonian parameters. The corresponding evolution time t is chosen by drawing a second particle $\underline{x}'_- \neq \underline{x}_-$ and setting $t = 1/\|H(\underline{x}'_-) - H(\underline{x}_-)\|_2$. For many models of interest, $\|H(\underline{x}'_-) - H(\underline{x}_-)\| \propto \|\underline{x}'_- - \underline{x}_-\|$, such that the PGH can be evaluated without explicitly considering large matrices. As the uncertainty in the estimated parameter shrinks, the PGH will tend to pick longer times to ensure that informative experiments continue to be chosen as certainty about the unknown parameters increases. In practice, this means that t grows as $1/\epsilon$ similar to phase estimation. The Loschmidt echo therefore guarantees that if $H_- \not\approx H$ then $\Pr(\psi|\underline{x})$ is small, but

if $H_- \approx H$ then $\Pr(\psi|\underline{x}) \approx 1$. This provides a sensitive test for distinguishing between good and poor hypotheses that works extremely well in practice and is provably optimal in certain cases [41].

6.2.2 Ideal Performance and Scaling with Dimension

We now consider the costs incurred by our procedure by breaking the cost down into two components,

$$\text{Cost}(\delta) := N_{\text{updates}}(\delta) \times \text{Cost}(\text{update}; \epsilon) \quad (6.4)$$

Formally, the cost of each Bayes update performed scales as [41]

$$\text{Cost}(\text{update}; \epsilon) \propto \frac{n}{\epsilon^2} \left(\mathbb{E}_{D|\underline{x}} \left[\frac{\max_i \Pr(D|\underline{x}_i)(1 - \Pr(D|\underline{x}_i))}{\Pr^2(D)} \right] \right), \quad (6.5)$$

where n is the number of SMC particles, ϵ is the adaptive likelihood estimation tolerance, and where D is the data record used in estimating \underline{x} . This is efficient only if we require no more than a polynomial number of particles in the number of qubits, and if the likelihood function is not too ``flat''. The former condition has been addressed for SMC algorithms more generally by Beskos et al [258], who showed that the number of particles required grows subexponentially in $\dim \underline{x}$. In practice, $\dim \underline{x}$ will often grow slowly with the dimension of the Hilbert space, such that we will require a small number of particles for a range of interesting applications. For example, the number of free parameters in any 2-local Hamiltonian model, such as the Ising model, grows at most quadratically with the number of sites being considered.

For the other condition, we can ensure that the likelihood functions are sufficiently far from flat by using the particle guess heuristic (Section 6.2.1.3). In particular, if at a given step, the posterior is approximately unimodal and has a mean that is within a fixed distance of the true parameter vector \underline{x} , then the PGH ensures that

$$\mathbb{E}_{\underline{x}_-} [\langle \psi | e^{iH(\underline{x}_-)t} e^{-iH(\underline{x}t)} | \psi \rangle] \in \Theta(1), \quad (6.6)$$

since $t \in \Theta(\|H(\underline{x}) - H(\underline{x}_-)\|^{-1})$ [41]. Thus, the total cost of interactive quantum likelihood evaluation is reduced superpolynomially by comparison with classical evaluation if the model dimension and the effective number of outcomes $\sum_j \Pr(d = j|\underline{x}_k)^{-2}$ are each modestly sized ($\underline{x} \in O(\text{polylog } \dim \mathcal{H})$).

Having thus analytically established the potential improvement offered by IQLE, we now consider a numerical example for a specific class of Hamiltonians. In order to enable

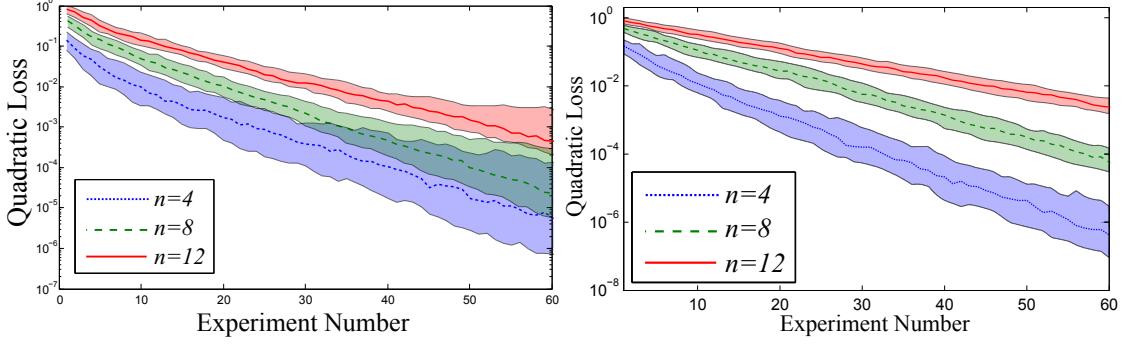


Figure 6.11: The median quadratic loss plotted as a function of the number of (left) non-interactive and (right) interactive QLE experiments, learning Ising models on the line graph for each of several numbers of qubits n . The shaded areas show a 50% confidence interval for the quadratic loss. 10 000, 10 000 and 20 000 particles were used in the $n = 4$, $n = 8$ and $n = 12$ cases respectively.

simulating the entire protocol on a classical computer, we will restrict our focus here to the Ising model

$$H(\underline{x}) = \sum_{(i,j) \in E(G)} x_{i,j} \sigma_z^{(i)} \sigma_z^{(j)} \quad (6.7)$$

for the edges $E(G)$ of an interaction graph G . Unless otherwise specified, we will take the prior and true distributions for each $x_{i,j}$ to be the uniform distribution $\text{Uni}(-1/2, 1/2)$, and will take the initial state to be $|\psi\rangle = |+\rangle^{\otimes n}$. The Ising model is of immediate relevance to current and recent experiments [236; 239], and allows us a model that is tractable to classical simulation such that we can numerically analyze performance. We need not coarse-grain in this case, as IQLE and the PGH together concentrate the measurement outcomes to a small number of effective outcomes.

The data for the linear and complete graphs (Figure 6.11 and Figure 6.12, respectively) show that the quadratic loss shrinks exponentially with the number of experiments performed, matching the observations made thus far in Hamiltonian learning. However, the rate at which the error decreases slows as the number of qubits n increases. This is expected, as $\dim \underline{x} = n(n - 1)/2$ for the case of a complete interaction graph, such that as the number of qubits increases, the same information is spread over more model parameters.

As we will discuss in Section 6.2.3.1, IQLE performs comparatively better when we in-

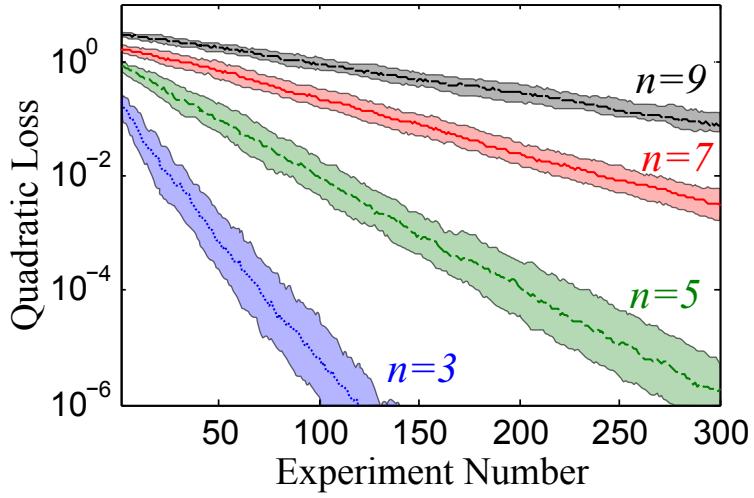


Figure 6.12: The median quadratic loss plotted as a function of the number of interactive QLE experiments, learning Ising models on the complete graph for each of several numbers of qubits n . The shaded areas show a 50% confidence interval for the quadratic loss.

clude the effects of finite sampling. In the linear case shown here, that we do not include sampling effects means that QLE can extract information effectively from a large number of outcomes. This is not feasible in practice, however, such that QLE demands more of our simulators than does IQLE. It is worth noting that only a few hundred IQLE experiments are needed to learn the Hamiltonian within a quadratic loss of 10^{-2} or smaller for even a nine-qubit system.

Critically, we note that the scaling parameter for quantum Hamiltonian learning is not the dimensionality of the Hamiltonian being learned, but rather, the dimensionality of the model parameter vector describing the Hamiltonian. In Figure 6.13, we illustrate this by considering the case where a single parameter in the prior starts off much less certain than the others, such that the error decay rate dynamically changes based on the dimensionality of the problem.

This observation allows us to reason about the scaling of our algorithm with respect to the number of unknown parameters directly. In particular, in Figure 6.14, we fit each of a large number of trials to $e^{-\gamma N}$, where N is the number of experiments performed, and where γ is the rate at which errors decay. We then plot the median decay for each of a number of different parameters in the linear and complete graphs, showing that

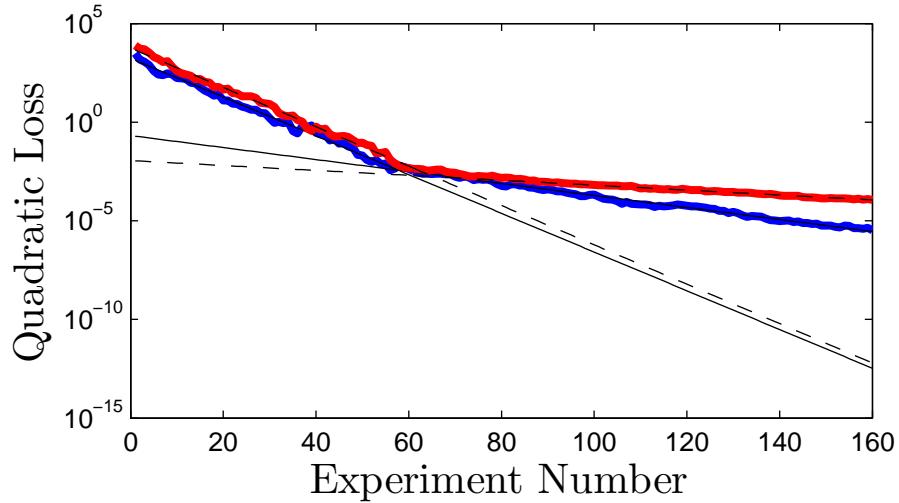


Figure 6.13: Scaling for four-qubit (red) and six-qubit (blue) spin chain models, where each model admits a single parameter that is much less certain than the others. The black lines illustrate that the exponential rate at which errors decay is dramatically different once the uncertain parameter ``catches up," such that the scaling switches to the $n = 4$ and $n = 6$ rates.

the falloff with the dimensionality of the model of the rate at which errors decay is not exponential. This in turn suggests that we can continue to learn exponentially quickly in larger and larger systems by exploiting prior information such as model reduction.

6.2.3 Robustness of QHL

In order to be a useful method for characterizing quantum systems, quantum Hamiltonian learning must be robust to realistic sources of error and noise. Here, we argue that QHL admits this robustness, using a combination of numerical results and analytic bounds. In particular, we will show that QHL continues to learn at an exponential rate in the presence of three distinct sources of error: that introduced by finite count statistics of measurements on the trusted simulator, noise introduced by imperfect implementations of the `SWAP` gate used to couple the two devices, and errors introduced by approximations in the Hamiltonian model under consideration.

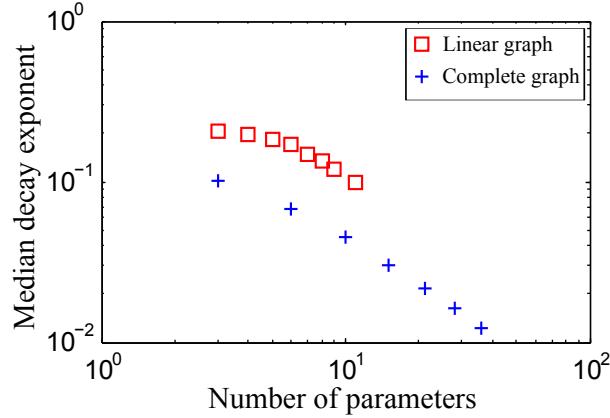


Figure 6.14: The median decay exponent for the quadratic loss as a function of the number of parameters in the Ising model d , and for each of the complete and linear interaction graphs.

6.2.3.1 Sampling Error

As discussed in [Section B.1](#), we can analytically bound the performance of quantum Hamiltonian learning under pessimistic assumptions, complementing known results on the stability of classical sequential Monte Carlo [258]. This bound above is overly pessimistic, however, as we have seen considerably better performance from our explorations of more simple likelihood-free models ([Appendix A](#)). Thus, we also consider sampling error numerically, using the `PoisonedModel` QInfer model to simulate finite adaptive likelihood estimation (ALE) tolerances. Doing so for a nine-qubit spin-chain Ising model, we observe in [Figure 6.15](#) that median performance of our algorithm still provides exponential decreases in uncertainty with the number of experimental data points collected, but that the slope of the exponential decay is reduced.

Moreover, if we tighten the ALE tolerance to with the number of qubits to represent the demand that the noise per qubit is held constant, then we obtain qualitatively similar results. in [Figure 6.16](#), we show that if the ALE tolerance is set to $\epsilon = 0.4/n$ where n is the number of qubits, then the median of the decay parameter γ obtained by fitting the loss for each trial to $Ae^{-\gamma N}$ scales in a manner comparable to the ideal case.

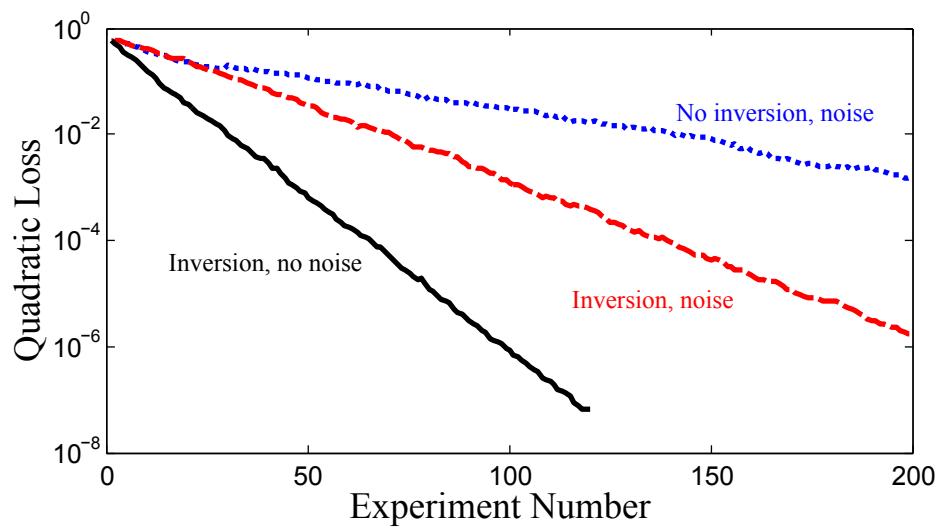


Figure 6.15: The median quadratic loss for a 9 qubit Ising model on the line for the IQLE and QLE cases with 10% ALE tolerance for estimating likelihoods, compared to the case of IQLE with perfect likelihood estimation. 10 000 particles were used for the learning algorithm in all cases.

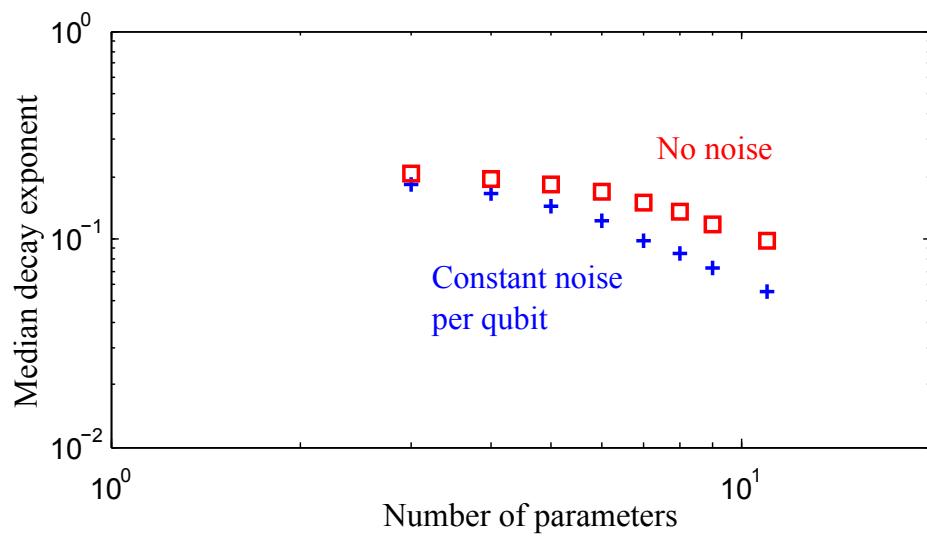


Figure 6.16: Median value of the error decay rate γ computed for IQLE experiments in which the interaction graph is a line, and where n ranges from 1 to 12 with ALE tolerance $\epsilon = 0.4/n$. For each experiment, learning was performed using 20 000 SMC particles. For comparison, the infinite-sampling limit $\epsilon = 0$ is also plotted.

6.2.3.2 Depolarizing Noise and Realistic SWAP Gates

Thus far, we have assumed that the SWAP gates coupling the trusted and untrusted devices are perfect, such that the state on the untrusted device is faithfully transferred to the trusted simulator. In practice, any implementation of SWAP will introduce decoherence into the system, such that QHL must be robust to this noise in order to successfully learn in practice. As such decoherence effectively reduces the visibility of measurements by a constant factor, by the argument given in [Section 2.1.3](#), we expect that if the noise is correctly modeled, then the exponential scalings observed thus far will be preserved, albeit at a slower rate.

In particular, we consider the disturbance to the prior introduced by a Bayes update conditioned on datum d drawn from a depolarized model with finite depolarizing probability p_{depol} . Let $p_{\text{ideal}} := \Pr(d|\underline{x}; p_{\text{depol}} = 0)$ be the ideal measurement probability, assuming no depolarizing noise, such that under depolarizing noise,

$$\Pr(d|\underline{x}) = p_{\text{ideal}}(1 - p_{\text{depol}}) + \frac{p_{\text{depol}}}{\dim \mathcal{H}}. \quad (6.8)$$

Moreover, let $p_{\text{ideal},j} := \Pr(d|x_j; p_{\text{depol}} = 0)$ represent the same for each individual hypothesis. Then, if we assume that $\min_j(1 - p_{\text{ideal},j}) \gg p_{\text{depol}}/(1 - p_{\text{depol}})$, we obtain that as the dimension of the Hamiltonian increases the disturbance $\mathbb{E}_d [\Pr(\underline{x}|d) / \Pr(\underline{x})]$ in the posterior asymptotically scales as [34],

$$(1 - p_{\text{depol}}) \left(\frac{p_{\text{ideal}}^2}{\sum_j p_{\text{ideal},j} \Pr(x_j)} + \frac{(1 - p_{\text{ideal}})^2}{\sum_j (1 - p_{\text{ideal},j} \Pr(x_j))} \right). \quad (6.9)$$

That is, the disturbance is reduced by the depolarizing strength to an effective learning rate $1 - p_{\text{depol}}$. As a particular consequence, the effect of depolarizing noise in the untrusted and trusted devices should not be expected to destabilize the estimates produced by QHL.

Here, we verify this numerically in two distinct ways: by simulating IQLE experiments with a depolarizing channel following the SWAP and by simulating IQLE using physically-realistic gates similar to those used in [Section 2.3.2](#) and [Chapter 4](#). For the first case, we let p_{ideal} be the strength of the depolarizing channel, and simulate the performance of IQLE on a four-qubit spin chain for varying \mathcal{N} . The results of this are shown in [Figure 6.17](#), in which we consider the median quadratic loss for a line-graph Ising model with varying settings of p_{depol} . Using depolarizing noise, we can also explicitly check the $(1 - p_{\text{depol}})$ scaling by considering the scaling of $\gamma/(1 - p_{\text{depol}})$ versus

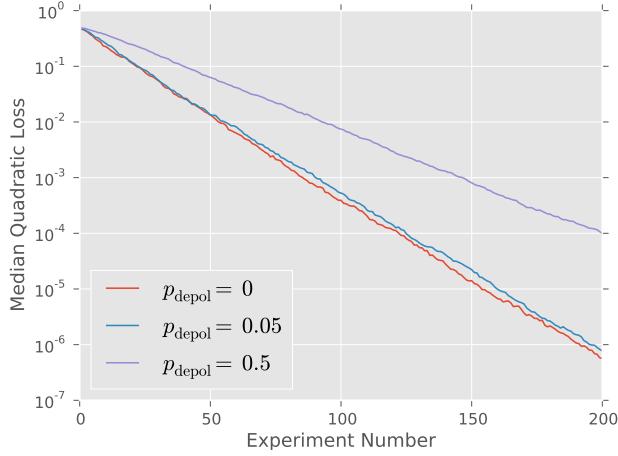


Figure 6.17: The median quadratic loss, plotting for 200 IQLE experiments with a line-graph Ising Hamiltonian on four qubits, and with varying levels of depolarizing noise p_{depol} .

model dimension, instead of γ versus $\dim \mathcal{X}$, as was considered in Figure 6.14. Using two different values of p_{depol} for each dimension, we show in Figure 6.18 evidence that the asymptotic scaling of (6.9) is a good approximation in cases of interest.

Generalizing from depolarizing noise, we again consider the action of cumulant-simulated SWAP gates in place of the ideal SWAP gate in Figure 6.10. We consider a quantum dots gate along with four gates of varying fidelity simulated using the superconducting qubit model. For each gate, a simulation using the method of [38] provides a superoperator description \hat{S}_{SWAP} . We then simulate quantum Hamiltonian learning on a two-qubit Ising model with each such gate. As shown in Figure 6.19, we are able to learn the J -coupling quite well, even in the presence of realistic noise models.

These two simulations thus show that quantum Hamiltonian learning can continue to operate well, even in the presence of realistic couplings to the trusted quantum resources. We require only that the coupling is well-characterized, such that the trusted simulator can include the effects of a realistic gate. This requirement is addressed by the classical methods of Chapter 2, however, such that we have a practical method for performing quantum Hamiltonian learning in experimentally-feasible scenarios.

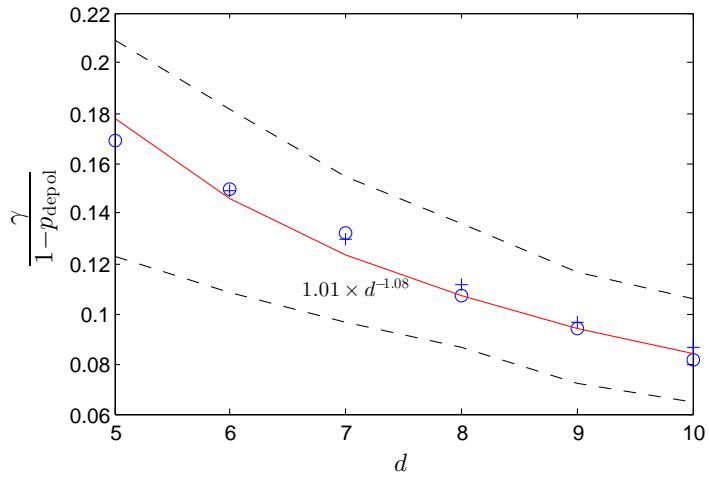


Figure 6.18: The median value of the error decay rate γ found by fitting the quadratic loss of 200 random line-graph Ising Hamiltonians to $Ae^{-\gamma N}$, rescaled by the asymptotic behavior predicted by (6.9), and where $d = \dim \underline{x}$ is the number of model parameters. The dashed lines represent a 50% confidence interval for the data, and the crosses and circles correspond to $p_{\text{depol}} = 0.75/d$ and $p_{\text{depol}} = 0.5/d$ respectively.

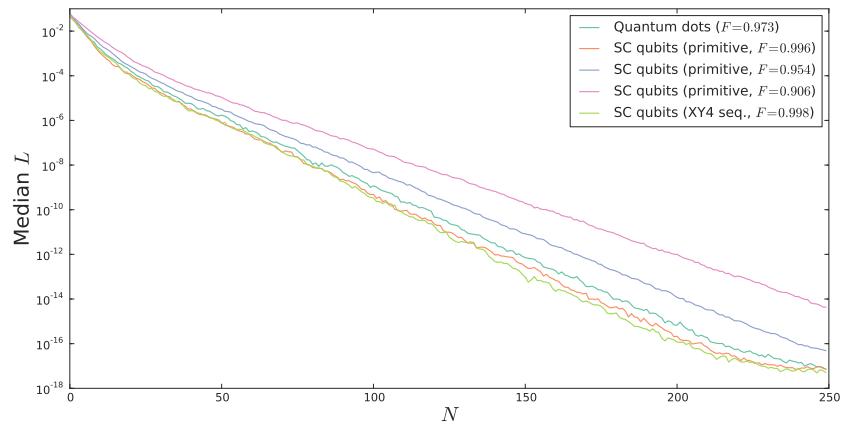


Figure 6.19: Median value of the loss L incurred by quantum Hamiltonian learning with IQLE in estimating the J -coupling between two qubits as a function of the number of measurements N performed, for each of the five physical models of the swap considered. The fidelities of each gate are shown in the inset. For the superconducting model, we also specify whether the gate was generated directly, or interleaved into an XY4 sequence.

6.2.3.3 Approximate and Truncated Models

Finally, we set the stage for the next algorithmic development by noting that quantum Hamiltonian learning can work with simulators implementing approximate models. That is, we relax the assumption that the family of Hamiltonians $H(\underline{x})$ accessible to the trusted device is not equal to the family of Hamiltonians \tilde{H} under which the device under study evolves. Though we consider the example of a truncated model here, the derivation holds for approximately-inverted Hamiltonians more generally [34].

As an example, we shall consider the case in which a large number of small terms in a Hamiltonian are neglected in implementing our trusted device. In particular, let \tilde{H} be an Ising model on the complete graph

$$\tilde{H} = \sum_{i=1}^{n-1} a_i \sigma_z^{(i)} \sigma_z^{(i+1)} + \sum_{i=1}^n \sum_{j=i+2}^n b_{i,j} \sigma_z^{(i)} \sigma_z^{(j)}, \quad (6.10)$$

where we assume that $a_i \sim \text{Uni}(-1/2, 1/2)$ and $b_{i,j} \sim N(0, (10^{-4})^2)$. We then consider inversion by a trusted simulator that only has access to the nearest-neighbor terms

$$H(\underline{x}) = \sum_{i=1}^{n-1} x_i \sigma_z^{(i)} \sigma_z^{(i+1)}. \quad (6.11)$$

The standard deviation of 10^{-4} was chosen to illustrate a regime in which the truncated terms are small by comparison to the included terms, as will be critical in our development of quantum bootstrapping (Section 6.4).

Using well-known bounds [47], we derive that the difference $\Delta \Pr(d|\underline{x})$ in the true likelihood for QHL experiments in this scenario and the likelihood function reconstructed using the trusted resources obeys [34]

$$\Delta \Pr(d|\underline{x}) \leq \|H(\underline{x}) - \tilde{H}\|^2 t^2. \quad (6.12)$$

The use of a truncated model is thus not problematic if the error in truncation is small compared to the difference in the forward and backwards evolution chosen by the PGH,

$$\|H(\underline{x}) - \tilde{H}(\underline{x})\| \ll \|H(\underline{x}) - H(\underline{x}_-)\|. \quad (6.13)$$

For truncation, $\|H(\underline{x}) - \tilde{H}(\underline{x})\| \leq R \max_k \|\tilde{H}_k\|$, where \tilde{H}_k are the Hamiltonian terms included in H but not \tilde{H} .

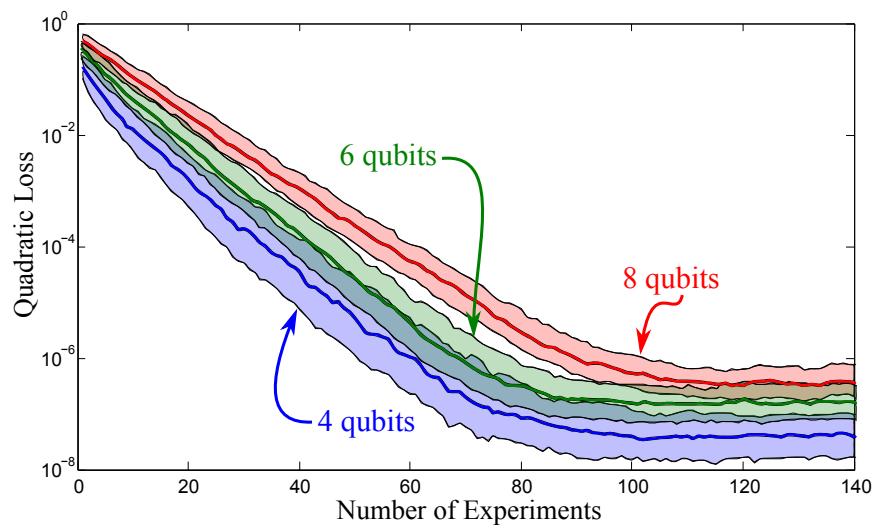


Figure 6.20: The performance of QHL for the case where the trusted simulator uses an Ising model on the line given that the true Hamiltonian is an Ising model on the complete graph with non-nearest neighbor interactions on the order of 10^{-4} and nearest neighbor interactions on the order of 0.5.

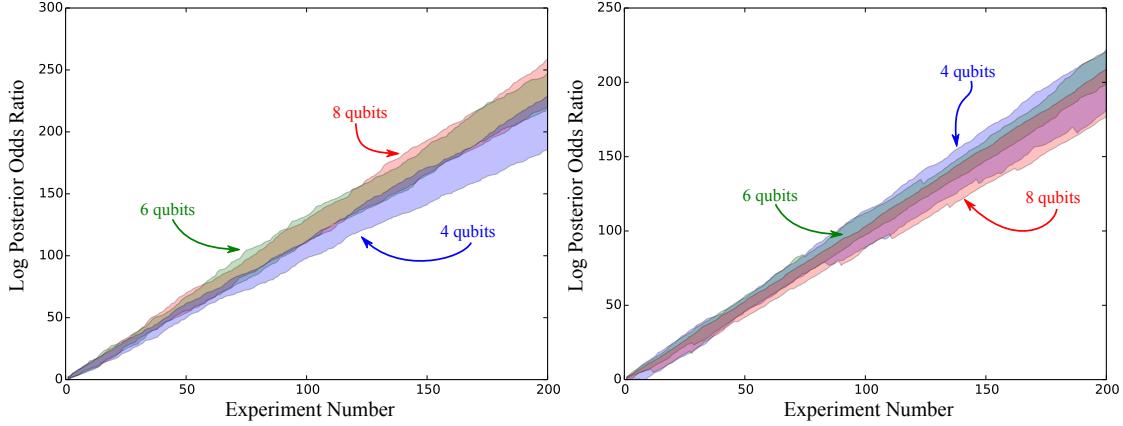


Figure 6.21: The logarithm of the posterior odds ratio of the true model to the reduced model, as analyzed by interactive quantum Hamiltonian learning. The shaded regions include all trials. (Left) The reduced model is a Hamiltonian with only nearest-neighbor couplings on a line while the true Hamiltonian contains small non--nearest neighbor couplings (order 10^{-4}). (Right) The true model contains only nearest-neighbor couplings, such that the complete model overfits.

In Figure 6.20, we show this numerically for \tilde{H} following (6.10), and for each of four, six and eight qubits. As predicted, interactive QHL continues to learn well until the excluded couplings are comparable to the uncertainty in the posterior. In particular, the plateaus in learning are reached at approximately $10^{-8} \times R$ for $R = \binom{n}{2} - n$, confirming (6.13).

Importantly, we can also use model selection (Section 3.2) to decide the validity of a truncated model by comparison to a more complete alternative. Comparing the posterior odds ratios for learning parameters of the complete- and line-graph Ising models from the same data record, for instance, provides evidence as to which model more accurately describes that data. In Figure 6.21, we show the results of this comparison for four-, six- and eight-qubit instances of each model. In all trials considered, we favor the correct model exponentially with the amount of data collected, further demonstrating that model selection is a powerful tool for deciding on the validity of an approximate model.

6.3 Quantum Hamiltonian Learning with Truncated Simulation

A major limitation of the quantum Hamiltonian learning algorithm as advanced thus far is that we have assumed that the quantum simulation resources used are of the same Hilbert space dimension as the system of interest. While this is useful in verification and validation procedures, and in using a quantum device to reduce the cost of producing further such devices, QHL is thus not immediately applicable to building a large quantum device in the first place. Here, we rectify this gap by using *small* quantum simulators to characterize and calibrate larger quantum devices.

We develop our algorithm by introducing *epistemic information locality*, whereby the causal support of observables is effectively reduced through repeated inversion by a hypothesis. This approach builds on the information- locality arguments of Da Silva et al [60], while avoiding the limitation of short-time evolutions. In particular, we show that information propagation in a quantum system can be effectively slowed down by using knowledge about the dynamics of that system, such that the particle guess heuristic can continue to explore long evolution times, even in a truncated simulation.

In this Section, we use epistemic information locality together with a novel extension to the sequential Monte Carlo algorithm to produce a characterization algorithm that can use quantum resources to characterize a larger quantum device. We formally bound the errors introduced by the compressed simulation steps, and then provide numerical examples of learning a 50-qubit Ising model with an 8-qubit Ising simulator.

6.3.1 Epistemic Information Locality

Lieb-Robinson bounds provide a formal analog of special relativity for local systems evolving under Hamiltonians that have rapidly decaying interactions [60; 259; 260; 261; 262]. This analogy gives an effective ``light cone'' around observables, as illustrated in Figure 6.22 for an observable A propagating inside the untrusted device before being inverted on the trusted simulator. Specifically, Lieb-Robinson bounds imply that $A(t)$ provides at most an exponentially small amount of information about systems that are farther than st away from the initial support of A , where s is the Lieb-Robinson velocity for the system. Thus, s plays the role of the speed of light in special relativity, providing a bound on the speed at which information can propagate within a system. The actual value of s depends only on the strength and geometry of interactions in the system. Thus,

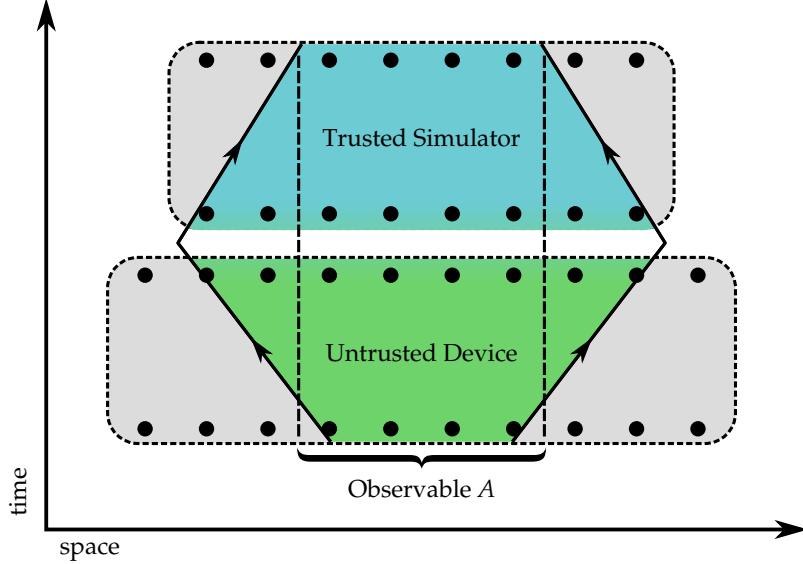


Figure 6.22: Light cones for $A(t)$ for a single step of an r step protocol. The green region is the light cone after the evolution in the untrusted device, and the blue region is after inversion in the trusted device. The dashed lines show the spread of $A(t)$ due to inexact inversion in the trusted simulator.

if st is bounded above by a constant and A has only small support, then measurements of A can be efficiently simulated.

We extend this using ideas from quantum Hamiltonian learning by transferring the state of a subregister of the larger system that we would like to characterize into a trusted quantum simulator, and then inverting the evolution by a hypothesis to reduce the norm of the effective Hamiltonian under which A evolves. Critically, this also reduces the effective Lieb-Robinson velocity describing the propagation of information from A , such that we use our hypothesis about the true evolution of the untrusted device to causally isolate A within the trusted simulator. *That is, the light cone about A represents an ‘‘epistemic’’ speed of light in the coupled systems that arises from the speed of information propagation depending more strongly on the uncertainty in the Hamiltonian than the Hamiltonian itself [10].*

In particular, by using the particle guess heuristic (PGH), our inversion hypothesis H_- approaches the true dynamics H_A of the restricted subregister we can access with our trusted device. The effective Hamiltonian $H_A - H_-$ thus approaches zero as the algorithm proceeds. Since the Lieb-Robinson velocity is linear in the norm of the largest

term in this effective Hamiltonian [260], under reasonable assumptions, the product $s|t|$ of the Lieb-Robinson velocity and the evolution time is approximately *constant*. Therefore, epistemic information locality allows us to use long evolution times, removing the major restriction of the Da Silva et al algorithm [60].

By the results of the previous Section, we know that quantum Hamiltonian learning is robust to approximations in the model used to estimate parameters. Here, we extend these results, using epistemic information locality to show that for systems with exponentially decaying interactions, compressed QHL incurs simulation errors that can be made negligibly small at costs that are small in both space and experimental data. We then show that the entire algorithm is efficient for such interactions, provided that each experiment yields a fraction of a bit of information about H .

In particular, we show in [Section B.2](#) that for non-commuting Hamiltonians that exhibit exponentially-decaying interactions, the error we incur scales based on our uncertainty in the true evolution of the system under study *within* the subregister that couples to our trusted simulator. This point critically forms the basis of our epistemic information locality argument, in that it shows that the Lieb-Robinson velocity for the truncated IQLE experiment is dependent *only* on the difference between the internal dynamics and the choice of inversion hypothesis. Using the particle guess heuristic to select our inversion hypotheses, we see Lieb-Robinson velocities decrease with our knowledge of the true truncated dynamics. In particular, in the limit of a well-characterized subregister, the Lieb-Robinson velocity vanishes entirely.

In the rest of this Section, we consider the important special case where all terms in the Hamiltonian under study mutually commute, such as in the case of Ising models without transverse fields. We then detail the modifications that we make to sequential Monte Carlo to enable reasoning about likelihood functions in which a small subset of parameters affect each individual datum. Finally, we demonstrate numerical examples in commuting-Hamiltonian case, using classical computation to simulate the performance of our algorithm when learning a 50-qubit Ising model with an 8-qubit trusted device.

6.3.2 Learning Commuting Hamiltonians

We first detail the operation of compressed quantum Hamiltonian learning in the important special case that all terms in the Hamiltonian under study are local and commute with each other. This is true, for instance, in the Ising models that we have studied thus far. In this special case, compressed QHL is particularly simple to analyze.

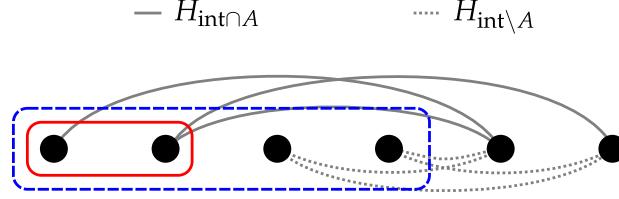


Figure 6.23: Separation of $H_{\text{int}} = H_{\text{int} \cap A} + H_{\text{int} \setminus A}$ where $H_{\text{int} \cap A}$ are interactions with qubits in the support of A (red solid box) and $H_{\text{int} \setminus A}$ interacts with qubits that are swapped into the trusted simulator but are outside A (blue dashed box).

Let an observable A be supported on a sites of an interaction graph. Then, we partition the Hamiltonian into terms based on their relation to the boundary of the trusted simulator and the support of A . As shown in Figure 6.23, we let

$$\begin{aligned} H &= H_{\text{int}} + H_{\text{in}} + H_{\text{out}} \\ &= H_{\text{int} \cap A} + H_{\text{int} \setminus A} + H_{\text{in}} + H_{\text{out}}, \end{aligned} \tag{6.14}$$

where H_{out} represents all terms that do not interact with the subsystems in the trusted simulator, $H_{\text{int}} = H_{\text{int} \cap A} + H_{\text{int} \setminus A}$ is the sum of all interactions between these subsystems and the simulated subsystems, and where H_{in} represents the internal dynamics common to both the simulator and the larger system.

If we work in the Heisenberg picture then it is easy to see from the assumption that the Hamiltonian terms commute with each other ¹ that $[H_{\text{int} \setminus A} + H_{\text{out}}, A(t)] = 0$. This implies that

$$A(t) = e^{iH_{\text{int}}t} e^{iH_{\text{int} \cap A}t} A e^{-iH_{\text{int}}t} e^{-iH_{\text{int} \cap A}t} \tag{6.15a}$$

$$\tilde{A}(t) = e^{iH_{\text{int}}t} A e^{-iH_{\text{int}}t}, \tag{6.15b}$$

where $\tilde{A}(t)$ is the simulated observable within the trusted simulator.

Using Hadamard's Lemma and the triangle inequality to bound the truncation error $\|\tilde{A}(t) - A(t)\|$, we obtain that

$$\|\tilde{A}(t) - A(t)\| \leq \|A\| (e^{2\|H_{\text{int} \cap A}\|t} - 1) \tag{6.16}$$

¹But not necessarily with A ; in fact, we require that $[A, H] \neq 0$ in order for the observable to carry information about H .

Thus, if we can tolerate an error of δ in the simulation then it suffices to choose experiments with evolution time at most

$$t \leq \ln \left(\frac{\delta}{\|A\|} + 1 \right) (2\|H_{\text{int}} \cap A\|)^{-1}. \quad (6.17)$$

If the sum of the magnitudes of the interaction terms that are neglected in the simulation is a constant then (6.17) shows that t scales at most linearly in δ as $\delta \rightarrow 0$. This is potentially problematic as short experiments can provide much less information than longer experiments. Thus, it may be desirable to increase the size of the trusted simulator as δ shrinks to reduce the experimental time needed to bootstrap the system. QHL is robust to δ [41; 34] (Section 6.2.3.3), such that $\delta \approx 0.01$ often suffices for the inference procedure to proceed without noticeable degradation.

Note that if $H_{\text{int}} \cap A = 0$ then infinite-time simulations are possible because literally no truncation error is incurred. Non-trivial cases for bootstrapping therefore only occur in commuting models with long range interactions.

As a particular example, we assume that the Hamiltonian is an Ising model on a line of length ℓ with exponentially-decaying interactions, such that non- nearest-neighbor couplings between sites i and j are at most $be^{-\alpha|i-j|}$. We then take A to be supported on a sites, and that the trusted simulator can simulate w sites. Then,

$$\|H_{\text{int}} \cap A\|^{-1} \geq (1 - e^{-\alpha})e^{\lfloor \frac{w-a}{2} \rfloor} / ab. \quad (6.18)$$

It therefore suffices to take $w - a$ logarithmic in t to guarantee error of δ for any fixed t . Similarly, if we assume the interaction strength between sites i and j is at most $b/|i-j|^\alpha$ for $\alpha > 1$ then

$$\|H_{\text{int}} \cap A\|^{-1} \geq \frac{(\lfloor \frac{w-a}{2} \rfloor + 1)^\alpha (\alpha - 1)}{ab\alpha}. \quad (6.19)$$

Picking $w - a \in O(t^{1/\alpha})$ guarantees fixed error δ for experimental time t [10]. Using insight from epistemic information locality, we show in Section B.2 that by the introduction of additional `SWAP` gates, we can explore longer evolution times, even with fixed $w - a$.

6.3.3 Scanning and Global SMC Clouds

In order for experimental data drawn from truncated interactive QLE experiments to depend on the entire untrusted register, we must *scan* the observable A across the untrusted device, such that the trusted simulator interacts with a sequence of different

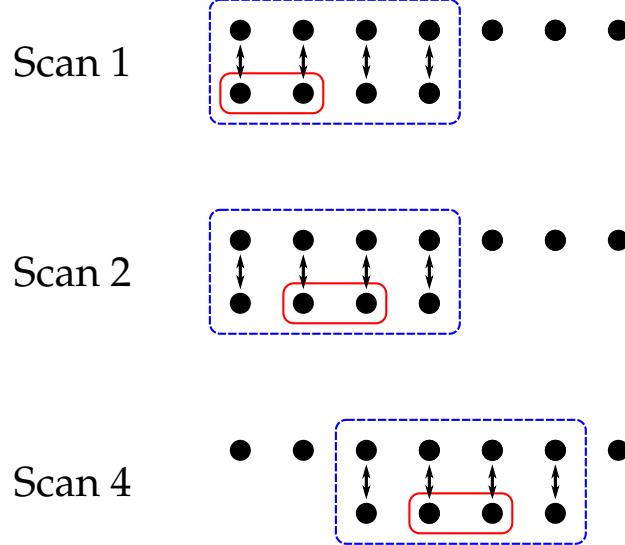


Figure 6.24: Scanning procedure for 7 qubits, a 4 qubit simulator and a 2 qubit observable. Blue (dashed) box is support of simulator, red (solid) box is support of A .

subregisters as the experiment proceeds. For example, in Figure 6.24, we show a 4-qubit trusted simulator scanning a 2-qubit observable across a 7-qubit spin chain.

This scanning procedure poses a difficulty to analysis with the sequential Monte Carlo algorithm, however, in that each datum depends only on a small subset of the full model parameter vector. Here, we address this limitation by extending SMC to use two sets of hypothesis particles, called the *local* and *global clouds*. The local cloud is formed by copying out a subset of the global cloud model parameters, which we then update according to a subset of the entire data record, using conventional SMC. Upon finishing these updates, we use resampling to reintroduce information from the local cloud back into the global cloud before proceeding to the next batch of data.

More formally, let s be a set of indices $\{s(1), \dots, s(k)\}$, and let $\underline{x}[s]$ be a vector of dimension k such that $(\underline{x}[s])_i = \underline{x}_{s(i)}$. That is, let s be a *slice* of \underline{x} that selects a subset of its elements. Then, suppose that the experimental data is partitioned as $D = \{d_{i,j}\}$ such that for some set of slices $\{s_i\}$, $d_{i,j} \perp \underline{x} \mid \underline{x}[s_i]$. If each s_i is the smallest slice for which this holds, then we have effectively partitioned the data and model parameters such that a subset of the model parameters are enough to learn properties of each *batch* of data, even if the full model parameter set is needed to learn from the entire data set. Given such a partitioning of the data and model parameters, the modified SMC algorithm proceeds

Table 6.2: Error in compressed QHL for 50-qubit Ising model with an 8-qubit simulator, for each of several choices of observable size a . Reported values are of the 2-norm distance $\|\hat{\underline{x}} - \underline{x}\|_2$ after 500 experiments / scan.

a	75 th percentile	Median	25 th percentile
2	0.252	0.234	0.225
4	0.0029	0.0018	0.0014
6	0.0043	0.0029	0.0014

as follows:

1. Prepare a global cloud $\{\underline{x}_i\} \sim \pi$ for a global prior $\pi(\underline{x})$.
2. For each batch $B_j = \{d_{j,k} : j\}$ with matching model slice s_j :
 - (a) Prepare a local cloud $\{\underline{z}_i\}$ by slicing each global particle with s_j , $\underline{z}_i = \underline{x}_i[s_j]$.
 - (b) Process B_j using [Algorithm 3](#), with the local cloud as a prior.
 - (c) If, after the last datum in B_j , the local posterior has not been resampled, resample it to ensure that all particle weights are uniform.
 - (d) Modify each global particle \underline{x}_i by overwriting $\underline{x}_i[s_j]$ with \underline{z}_i from the final local posterior.

Since resampling does not preserve particle correspondence, and will in general shuffle the order of particles, this algorithm makes as an approximation that $\Pr(\underline{x}|B_j) = \Pr(\underline{x}[s_j]|B_j) \Pr(\underline{x}[s_j^\perp]|B_j)$ where s_j^\perp is the complement of s_j . That is, any correlations between global and local model parameters are lost after each reset of the global particles by the local particles. If learning proceeds well within each batch, this is a good approximation provided that $|B_j|$ is sufficiently large.

To demonstrate the performance of compressed QHL with modified SMC, we consider the case of a 50-qubit Ising model and an 8-qubit trusted simulator. The interaction graph is taken to be a complete graph, exponentially decaying with distance on a 1D chain. That is, the Hamiltonian is taken to be of the form

$$H(\underline{x}) = \sum_{i \neq j} x_{i,j} \sigma_z^{(i)} \sigma_z^{(j)}, \quad (6.20)$$

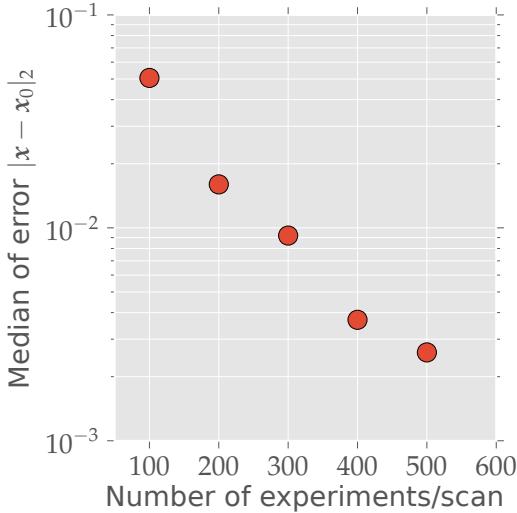


Figure 6.25: Error in compressed QHL for $a = 4$ with varying N_{exp} per scan. Data consistent with $e^{-0.006N_{\text{exp}}}$ scaling.

where the parameters $x_{i,j}$ are distributed according to $x_{i,j} \sim \text{Uni}(0, 1)10^{-2(|i-j|-1)}$. The observable is taken to be $A = (|+\rangle\langle+|)^{\otimes a}$ for $a \in \{2, 4, 6\}$.

In Table 6.2, we show the error incurred by this procedure versus the size a of the observable used. Based on these observations, we choose $a = 4$, and then consider the number of experiments in each scan. In Figure 6.25, we show that the error continues to decay exponentially with the amount of data collected.

6.4 Quantum Bootstrapping

Having demonstrated compressed quantum Hamiltonian learning using epistemic information locality, we are now equipped to apply it towards a long-standing problem in quantum information: how can we apply quantum resources to not only characterize larger quantum devices, but control them? Put differently, we want to *bootstrap* our way to large quantum simulators [263].

6.4.1 Control Characterization and Tuneup

Here, we address this challenge by using compressed QHL (cQHL) to calibrate controls for a larger but untrusted quantum simulator, producing the settings required to use it to characterize a still larger system in turn. As a model, we consider that an uncalibrated simulator has a Hamiltonian that is an affine function of a control vector \underline{c} ,

$$H(\underline{c}) = H_0 + \underline{H}_c \cdot \underline{c}. \quad (6.21)$$

Calibration then consists of finding control vectors \underline{c}_i that produce each of a list of Hamiltonian controls $\{H_i\}$ needed to perform simulations at the next bootstrapping iteration.

This model addresses the problem of *crosstalk* in quantum systems, previously addressed by the use of randomized benchmarking [158] and the CODA algorithm [264]. Crosstalk is also a special case of the distortion models that can be compensated for by quantum-accelerated optimal control algorithms [85; 86; 25] (Section 6.1.3). Quantum bootstrapping with cQHL builds on both of these lines of research by using quantum resources not to design an individual pulse, but to produce full calibration information needed to simulate a wide range of Hamiltonians.

In particular, we proceed by first learning $H_0 = H(0)$ using cQHL, and then learning $H_0 + H_{c,i} = H(\underline{c}_i)$ for each control knob i . We then report as our estimates $\hat{H}_{c,i}$ the difference between our estimates of $H(\underline{c}_i)$ and $H(0)$. Having learned the action of the control function for each knob, we can finish the calibration procedure by using the Moore-Penrose pseudoinverse to find \underline{c}_i that implements the desired Hamiltonian \underline{H}_i .

To implement the simulation of $H' \in \text{span}(\{H_i\})$ at the next iteration, we can use the Moore-Penrose pseudoinverse again for H' directly, but this requires applying the pseudoinverse for each Hamiltonian we wish to implement. As an alternative, we can implement desired Hamiltonians using Trotter-Suzuki sequences [265] of the calibrated control terms. This approach also illustrates that since we can synthesize commutator sequences of calibrated control terms, we need not characterize as many control knobs as in the trusted device, or as in the desired set of interactions. In particular, next-nearest-neighbor terms can be accurately synthesized from nearest-neighbor terms using sequences such as

$$e^{-2iZ \otimes \mathbb{1} \otimes Z \Delta t^2} |\phi\rangle + O(\Delta t^3) = e^{-iZ \otimes X \otimes \mathbb{1} \Delta t} e^{-i\mathbb{1} \otimes Y \otimes Z \Delta t} e^{iZ \otimes X \otimes \mathbb{1} \Delta t} e^{i\mathbb{1} \otimes Y \otimes Z \Delta t}, \quad (6.22)$$

assuming that the middle qubit is prepared in $|0\rangle$. Higher-order and parallel methods for engineering interactions from local control terms are given by the wide quantum channel [40] (Section 5.3.1) and by the Childs-Wiebe algorithm [232].

6.4.2 Numerical Examples

We have demonstrated quantum bootstrapping by again using a numerical example of an 8-qubit Ising model simulator to characterize a 50-qubit Ising model with exponentially decaying interactions along a linear spin chain, again using the Hamiltonian of (6.20). Here, our goal is to produce each nearest-neighbor term in isolation, allowing for crosstalk within the 50-qubit system. The model for this problem thus consists of $49 \times \binom{50}{2} \approx 61 \times 10^3$ parameters, as we must learn the Hamiltonian produced by each of the 49 individual controls. In Figure 6.26, we present histograms over the model parameters of the error in each calibrated Hamiltonian term for examples using 100, 200 and 300 compressed interactive QLE experiments per scan. We note that with only 300 experiments per scan, most of the Hamiltonians are calibrated to within 10^{-2} error relative to the initial distribution.

Quantum bootstrapping therefore provides a valuable tool for characterizing and calibrating large quantum devices using small quantum resources together with novel insights into physics and statistical inference algorithms.

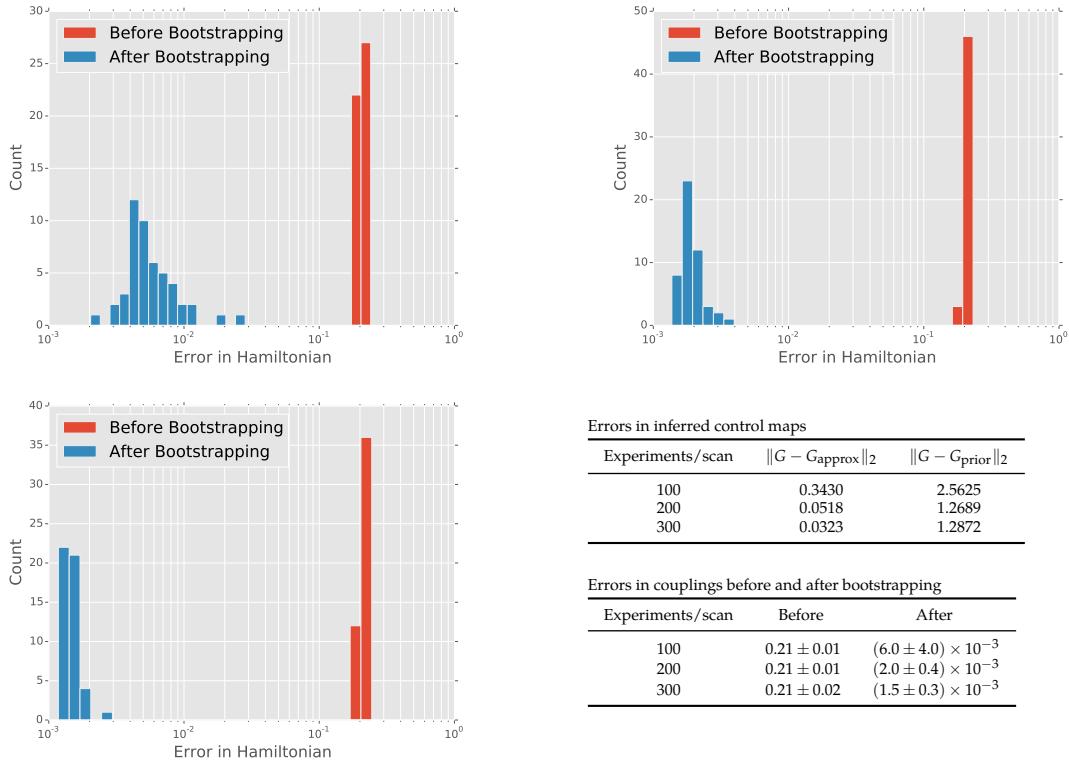


Figure 6.26: Distribution of errors for each of the 49 Hamiltonian terms in the bootstrapped Hamiltonian for a 50 qubit Ising model using (left) 100 (right) 200 and (bottom left) 300 IQLE experiments per scan.

7 Conclusions: Infrastructure for Large Quantum Devices

In this work, we have presented a number of results establishing methods for the characterization, verification and control of quantum systems. In doing so, we have taken an *algorithmic* focus, such that our methods are applicable in a wide range of contexts, from nitrogen-vacancy centers and electronic spin resonance to neutron interferometry.

The algorithms presented here, including sequential Monte Carlo, honest approximation and nonlinear optimal control, all use classical resources to solve concerns that cut across modalities for quantum information processing. That is, we have significantly advanced the *infrastructure* for characterizing and controlling quantum devices. This infrastructure is implemented using readily-available tools that build on existing software tools for experimental practice. Moreover, this infrastructure is robust, as the algorithms we present here characterize their own performance and can be adapted accordingly.

By the inclusion of quantum resources, we extend our infrastructure significantly. In characterization, we demonstrated that using quantum simulation to perform statistical inference, we could mitigate one of the most pressing limitations preventing us from learning the Hamiltonian dynamics of quantum systems. Moreover, we showed that our quantum Hamiltonian learning algorithm remains robust to a wide range of practical considerations.

In control, the use of quantum resources was developed further by showing that

evolutionary algorithms can be used to apply quantum coprocessors to the problem of finding optimal pulses. Here as well, we enjoy robustness to experimental concerns, such as effects of the noise figure when monitoring the output of pulse distortions. By using a population-based algorithm, we avoid depending too strongly on any single objective function evaluation. Moreover, by formulating the optimal control problem as a multi-objective optimization, we can build in a demand that we remain robust over a range of hypotheses.

We have also shown a very striking feature of our inference procedures: Hamiltonian learning remains robust even when we use a model that does not accurately and fully reflect the underlying physics of a device. In particular, excluded couplings do not necessarily cause Hamiltonian learning to fail until those couplings dominate over the uncertainty in the couplings that we are attempting to learn. We are therefore free to employ useful *approximations* to the dynamics of quantum systems that we wish to study. This is especially useful when combined with novel approximations that build on the structure of quantum Hamiltonian learning problems to dramatically reduce the resources required to perform the required simulations. Thus, by drawing on new insights into *epistemic* information locality, we can adaptively use our current knowledge about a quantum system to apply Hamiltonian learning even in quite large systems, and can also be extended to calibrate for control distortions such as crosstalk.

Taken on the whole, we have developed and demonstrated the efficacy of an infrastructure for the control, characterization and verification of quantum systems across a range of modalities and scales. In doing so, we make significant process towards the realization of quantum resources beyond the classical scale.

References

- [1] J. R. Johansson, P. D. Nation, and Franco Nori. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, April 2013. ISSN 00104655. doi: 10.1016/j.cpc.2012.11.019. URL <http://arxiv.org/abs/1211.6518>. arXiv:1211.6518 [quant-ph].
- [2] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Tim Purcell. *A survey of general-purpose computation on graphics hardware*. 2007.
- [3] Andrew Putnam, Adrian Caulfield, Eric Chung, Derek Chiou, Kypros Constantinides, John Demme, Hadi Esmaeilzadeh, Jeremy Fowers, Gopi Prashanth Gopal, Jan Gray, Michael Haselman, Scott Hauck, Stephen Heil, Amir Hormati, Joo-Young Kim, Sitaram Lanka, Jim Larus, Eric Peterson, Simon Pope, Aaron Smith, Jason Thong, Phillip Yi Xiao, and Doug Burger. A reconfigurable fabric for accelerating large-scale datacenter services. In *41st Annual International Symposium on Computer Architecture (ISCA)*, June 2014. URL <http://research.microsoft.com/apps/pubs/default.aspx?id=212001>.
- [4] Scott Aaronson. *Quantum Computing since Democritus*. Cambridge University Press, New York, April 2013. ISBN 9780521786492.
- [5] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. 1011.3245, November 2010. URL <http://arxiv.org/abs/1011.3245>.

- [6] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982. ISSN 0020-7748, 1572-9575. doi: 10.1007/BF02650179. URL <http://link.springer.com/article/10.1007/BF02650179>.
- [7] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *arXiv:quant-ph/9508027*, August 1995. URL <http://arxiv.org/abs/quant-ph/9508027>. SIAM J.Sci.Statist.Comput. 26 (1997) 1484.
- [8] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer. Improving quantum algorithms for quantum chemistry. *arXiv:1403.1539 [quant-ph]*, March 2014. URL <http://arxiv.org/abs/1403.1539>.
- [9] Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. arXiv e-print quant-ph/0405146, May 2004. URL <http://arxiv.org/abs/quant-ph/0405146>. Phys. Rev. Lett. 95, 050501 (2005).
- [10] Nathan Wiebe, Christopher Granade, and David G. Cory. Quantum bootstrapping via compressed quantum Hamiltonian learning. *arXiv:1409.1524 [quant-ph]*, September 2014. URL <http://arxiv.org/abs/1409.1524>. arXiv: 1409.1524.
- [11] Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore. Quantum deep learning. *SciRate*, December 2014. URL <https://scirate.com/arxiv/1412.3489>.
- [12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109(5):050505, August 2012. doi: 10.1103/PhysRevLett.109.050505. URL <http://link.aps.org/doi/10.1103/PhysRevLett.109.050505>.
- [13] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011. ISBN 978-0-387-85819-7. URL <http://dblp.uni-trier.de/db/reference/rsh/rsh2011.html#DesrosiersK11>.
- [14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691, 1573-1405. doi: 10.1023/B:VISI.0000029664.99615.94. URL <http://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94>.

- [15] Brad Miller, Ling Huang, A. D. Joseph, and J. D. Tygar. I know why you went to the clinic: Risks and realization of HTTPS traffic analysis. *arXiv:1403.0297 [cs]*, March 2014. URL <http://arxiv.org/abs/1403.0297>.
- [16] Mark Regan. Timeseries classification: KNN & DTW, September 2014. URL http://nbviewer.ipython.org/github/markdregan/K-Nearest-Neighbors-with-Dynamic-Time-Warping/blob/master/K_Nearest_Neighbor_Dynamic_Time_Warping.ipynb.
- [17] R. M. Parry, W. Jones, T. H. Stokes, J. H. Phan, R. A. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. D. Wang. k-nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *The Pharmacogenomics Journal*, 10(4):292–309, August 2010. ISSN 1470-269X. doi: 10.1038/tpj.2010.56. URL <http://www.nature.com/tpj/journal/v10/n4/full/tpj201056a.html>.
- [18] Tao Ban, Ruibin Zhang, Shaoning Pang, Abdolhossein Sarrafzadeh, and Daisuke Inoue. Referential kNN regression for financial time series forecasting. In Minho Lee, Akira Hirose, Zeng-Guang Hou, and Rhee Man Kil, editors, *Neural Information Processing*, number 8226 in Lecture Notes in Computer Science, pages 601–608. Springer Berlin Heidelberg, January 2013. ISBN 978-3-642-42053-5, 978-3-642-42054-2. URL http://link.springer.com/chapter/10.1007/978-3-642-42054-2_75.
- [19] Troy William Borneman. *Techniques for noise suppression and robust control in spin-based quantum information processors*. Thesis, Massachusetts Institute of Technology, 2012. URL <http://dspace.mit.edu/handle/1721.1/80658>. Thesis (Ph. D.)—Massachusetts Institute of Technology, Dept. of Nuclear Science and Engineering, February 2013.
- [20] Alexandre Blais, Jay Gambetta, A. Wallraff, D. I. Schuster, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Quantum-information processing with circuit quantum electrodynamics. *Physical Review A*, 75(3):032329, March 2007. doi: 10.1103/PhysRevA.75.032329. URL <http://link.aps.org/doi/10.1103/PhysRevA.75.032329>.
- [21] Dohun Kim, Zhan Shi, C. B. Simmons, D. R. Ward, J. R. Prance, Teck Seng Koh, John King Gamble, D. E. Savage, M. G. Lagally, Mark Friesen, S. N. Coppersmith, and M. A. Eriksson. Quantum control and process tomography of a semiconductor

- quantum dot hybrid qubit. *arXiv:1401.4416 [cond-mat]*, January 2014. URL <http://arxiv.org/abs/1401.4416>. arXiv: 1401.4416.
- [22] D. Kielpinski, C. Monroe, and D. J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417(6890):709–711, June 2002. ISSN 0028-0836. doi: 10.1038/nature00784. URL <http://www.nature.com/nature/journal/v417/n6890/full/nature00784.html>.
 - [23] J?rg Wrachtrup and Fedor Jelezko. Processing quantum information in diamond. *Journal of Physics: Condensed Matter*, 18(21):S807, May 2006. ISSN 0953-8984. doi: 10.1088/0953-8984/18/21/S08. URL <http://iopscience.iop.org/0953-8984/18/21/S08>.
 - [24] OferM. Shir, Jonathan Roslund, Zaki Leghtas, and Herschel Rabitz. Quantum control experiments as a testbed for evolutionary multi-objective algorithms. *Genetic Programming and Evolvable Machines*, 13(4):445–491, 2012. ISSN 1389-2576. doi: 10.1007/s10710-012-9164-7. URL <http://dx.doi.org/10.1007/s10710-012-9164-7>.
 - [25] Christopher Ferrie and Osama Moussa. Robust and efficient in situ quantum control. *arXiv:1409.3172 [quant-ph]*, September 2014. URL <http://arxiv.org/abs/1409.3172>. arXiv: 1409.3172.
 - [26] Philipp Hauke, Fernando M. Cucchietti, Luca Tagliacozzo, Ivan Deutsch, and Maciej Lewenstein. Can one trust quantum simulators? *Reports on Progress in Physics*, 75(8):082401, August 2012. ISSN 0034-4885. doi: 10.1088/0034-4885/75/8/082401. URL <http://iopscience.iop.org/0034-4885/75/8/082401>.
 - [27] P. Cappellaro, J. Emerson, N. Boulant, C. Ramanathan, S. Lloyd, and D. G. Cory. Entanglement assisted metrology. *Physical Review Letters*, 94(2):020502, January 2005. doi: 10.1103/PhysRevLett.94.020502. URL <http://link.aps.org/doi/10.1103/PhysRevLett.94.020502>.
 - [28] Alexander Hentschel and Barry C. Sanders. Machine learning for precise quantum measurement. *Physical Review Letters*, 104(6):063603, February 2010. doi: 10.1103/PhysRevLett.104.063603. URL <http://link.aps.org/doi/10.1103/PhysRevLett.104.063603>.
 - [29] David Poulin, M. B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doherty, and Matthias Troyer. The Trotter step size required for accurate quantum simu-

- lation of quantum chemistry. *arXiv:1406.4920 [quant-ph]*, June 2014. URL <http://arxiv.org/abs/1406.4920>. arXiv: 1406.4920.
- [30] Christopher Ferrie, Christopher E. Granade, and D. G. Cory. Adaptive Hamiltonian estimation using Bayesian experimental design. *AIP Conference Proceedings*, 1443(1):165–173, May 2012. ISSN 0094243X. doi: doi:10.1063/1.3703632. URL http://proceedings.aip.org/resource/2/apcpc/1443/1/165_1?isAuthorized=no.
 - [31] Christopher Ferrie, Christopher E. Granade, and D. G. Cory. How to best sample a periodic probability distribution, or on the accuracy of Hamiltonian finding strategies. *Quantum Information Processing*, 12(1):611–623, January 2013. ISSN 1570-0755, 1573-1332. doi: 10.1007/s11128-012-0407-6. URL <http://link.springer.com/article/10.1007/s11128-012-0407-6>.
 - [32] Christopher E Granade, Christopher Ferrie, Nathan Wiebe, and D G Cory. Robust online Hamiltonian learning. *New Journal of Physics*, 14(10):103013, October 2012. ISSN 1367-2630. doi: 10.1088/1367-2630/14/10/103013. URL <http://iopscience.iop.org/1367-2630/14/10/103013>.
 - [33] Christopher Granade, Christopher Ferrie, and D. G. Cory. Accelerated randomized benchmarking. *arXiv:1404.5275 [quant-ph]*, April 2014. URL <https://scirate.com/arxiv/1404.5275>.
 - [34] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David Cory. Quantum Hamiltonian learning using imperfect quantum resources. *Physical Review A*, 89(4):042314, April 2014. doi: 10.1103/PhysRevA.89.042314. URL <http://link.aps.org/doi/10.1103/PhysRevA.89.042314>.
 - [35] Christopher Ferrie. High posterior density ellipsoids of quantum states. *New Journal of Physics*, 16(2):023006, February 2014. ISSN 1367-2630. doi: 10.1088/1367-2630/16/2/023006. URL <http://iopscience.iop.org/1367-2630/16/2/023006>.
 - [36] Christopher Ferrie. Quantum model averaging. *arXiv:1405.6329 [quant-ph]*, May 2014. URL <http://arxiv.org/abs/1405.6329>.
 - [37] Easwar Magesan, Daniel Puzzuoli, Christopher E. Granade, and David G. Cory. Modeling quantum noise for efficient testing of fault-tolerant circuits. *Physical Review A*, 87(1):012324, January 2013. doi: 10.1103/PhysRevA.87.012324. URL <http://link.aps.org/doi/10.1103/PhysRevA.87.012324>.

- [38] Daniel Puzzuoli. *Honest Approximations to Realistic Fault Models and Their Applications to Efficient Simulation of Quantum Error Correction*. PhD thesis, April 2014. URL <https://uwspace.uwaterloo.ca/handle/10012/8360>.
- [39] Ian N. Hincks, Christopher Granade, Troy W. Borneman, and D. G. Cory. Accounting for classical hardware in the control of quantum devices. *SciRate*, September 2014. URL <https://scirate.com/arxiv/1409.8178>.
- [40] T. W. Borneman, C. E. Granade, and D. G. Cory. Parallel information transfer in a multinode quantum information processor. *Physical Review Letters*, 108(14):140502, April 2012. doi: 10.1103/PhysRevLett.108.140502. URL <http://link.aps.org/doi/10.1103/PhysRevLett.108.140502>.
- [41] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and D. G. Cory. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*, 112(19):190501, May 2014. doi: 10.1103/PhysRevLett.112.190501. URL <http://link.aps.org/doi/10.1103/PhysRevLett.112.190501>.
- [42] Christopher Wood, Ian Hincks, and Christopher Granade. *QuantumUtils for Mathematica*. 2015. URL <https://github.com/QuantumUtils/quantum-utils-mathematica/>.
- [43] Ian Hincks, and Christopher Wood, Christopher Granade, D. G. Cory, et al. *Quantum Utils for MATLAB*. 2014. URL <https://github.com/CoryGroup/quantum-utils-matlab>.
- [44] Christopher Granade, Christopher Ferrie, et al. *QInfer: Library for Statistical Inference in Quantum Information*. 2012. URL <https://github.com/csferrie/python-qinfer>.
- [45] Christopher Granade, Ben Criger, et al. *QuaEC: Quantum Error Correction Analysis in Python*. 2012. URL <https://github.com/cgranade/python-quaec>.
- [46] Max Born. On the quantum mechanics of collisions. In *Quantum theory and measurement*. December 1926.
- [47] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 1 edition, September 2000. ISBN 0521635039.
- [48] Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 463(2088):3089–3114, December

2007. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2007.0113. URL <http://rspa.royalsocietypublishing.org/content/463/2088/3089>.
- [49] K. Vogel and H. Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. *Physical Review A*, 40(5):2847--2849, September 1989. doi: 10.1103/PhysRevA.40.2847. URL <http://link.aps.org/doi/10.1103/PhysRevA.40.2847>.
 - [50] A. I. Lvovsky and M. G. Raymer. Continuous-variable optical quantum-state tomography. *Reviews of Modern Physics*, 81(1):299–332, March 2009. doi: 10.1103/RevModPhys.81.299. URL <http://link.aps.org/doi/10.1103/RevModPhys.81.299>.
 - [51] Robin Blume-Kohout. Optimal, reliable estimation of quantum states. *New Journal of Physics*, 12(4):043034, April 2010. ISSN 1367-2630. doi: 10.1088/1367-2630/12/4/043034. URL <http://iopscience.iop.org/1367-2630/12/4/043034>.
 - [52] Daniel F. V. James, Paul G. Kwiat, William J. Munro, and Andrew G. White. Measurement of qubits. *Physical Review A*, 64(5):052312, October 2001. doi: 10.1103/PhysRevA.64.052312. URL <http://link.aps.org/doi/10.1103/PhysRevA.64.052312>.
 - [53] Paul Busch. Informationally complete sets of physical quantities. *International Journal of Theoretical Physics*, 30(9):1217–1227, September 1991. ISSN 0020-7748, 1572-9575. doi: 10.1007/BF00671008. URL <http://link.springer.com/article/10.1007/BF00671008>.
 - [54] G. M. D'Ariano, L. Maccone, and M. G. A. Paris. Orthogonality relations in quantum tomography. *Physics Letters A*, 276(1-4):25–30, October 2000. ISSN 0375-9601. doi: 10.1016/S0375-9601(00)00660-5. URL <http://www.sciencedirect.com/science/article/pii/S0375960100006605>.
 - [55] Carlton M. Caves, Christopher A. Fuchs, and Rüdiger Schack. Unknown quantum states: The quantum de Finetti representation. *Journal of Mathematical Physics*, 43(9):4537–4559, September 2002. ISSN 0022-2488, 1089-7658. doi: 10.1063/1.1494475. URL <http://scitation.aip.org/content/aip/journal/jmp/43/9/10.1063/1.1494475>.
 - [56] Steven T. Flammia, Andrew Silberfarb, and Carlton M. Caves. Minimal informationally complete measurements for pure states. *Foundations of Physics*, 35

- (12):1985–2006, December 2005. ISSN 0015-9018, 1572-9516. doi: 10.1007/s10701-005-8658-z. URL <http://link.springer.com/article/10.1007/s10701-005-8658-z>.
- [57] G. Mauro D'Ariano, Martina De Laurentis, Matteo G. A. Paris, Alberto Porzio, and Salvatore Solimeno. Quantum tomography as a tool for the characterization of optical devices. *Journal of Optics B: Quantum and Semiclassical Optics*, 4(3):S127, June 2002. ISSN 1464-4266. doi: 10.1088/1464-4266/4/3/366. URL <http://iopscience.iop.org/1464-4266/4/3/366>.
 - [58] Robin Blume-Kohout. Hedged maximum likelihood estimation. 2010. URL <http://arxiv.org/abs/1001.2029>.
 - [59] F. Huszár and N. M. T. Houslsby. Adaptive Bayesian quantum tomography. *Physical Review A*, 85(5):052120, May 2012. doi: 10.1103/PhysRevA.85.052120. URL <http://link.aps.org/doi/10.1103/PhysRevA.85.052120>.
 - [60] Marcus P. da Silva, Olivier Landon-Cardinal, and David Poulin. Practical characterization of quantum devices without tomography. *Physical Review Letters*, 107(21):210404, November 2011. doi: 10.1103/PhysRevLett.107.210404. URL <http://link.aps.org/doi/10.1103/PhysRevLett.107.210404>.
 - [61] N. Boulant, T. F. Havel, M. A. Pravia, and D. G. Cory. Robust method for estimating the lindblad operators of a dissipative quantum process from measurements of the density operator at multiple time points. *Physical Review A*, 67(4):042322, April 2003. doi: 10.1103/PhysRevA.67.042322. URL <http://link.aps.org/doi/10.1103/PhysRevA.67.042322>.
 - [62] Daniel Gottesman. Stabilizer codes and quantum error correction. *quant-ph/9705052*, May 1997. URL <http://arxiv.org/abs/quant-ph/9705052>.
 - [63] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, November 2004. doi: 10.1103/PhysRevA.70.052328. URL <http://link.aps.org/doi/10.1103/PhysRevA.70.052328>.
 - [64] Christopher J. Wood, Jacob D. Biamonte, and David G. Cory. Tensor networks and graphical calculus for open quantum systems. *arXiv:1111.6950 [quant-ph]*, November 2011. URL <http://arxiv.org/abs/1111.6950>. arXiv: 1111.6950.

- [65] John Watrous. *CS 766: Theory of Quantum Information (lecture notes)*. 2013. URL <https://cs.uwaterloo.ca/~watrous/CS766/>.
- [66] Martin Ringbauer, Christopher J. Wood, Kavan Modi, Alexei Gilchrist, Andrew G. White, and Alessandro Fedrizzi. Characterizing quantum dynamics with initial system-environment correlations. *arXiv:1410.5826 [quant-ph]*, October 2014. URL <http://arxiv.org/abs/1410.5826>. arXiv: 1410.5826.
- [67] Joseph Emerson. *Open Quantum Systems*. 2011.
- [68] Christopher Wood. PhD thesis, 2014.
- [69] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. Ancilla-assisted quantum process tomography. *Physical Review Letters*, 90(19):193601, May 2003. doi: 10.1103/PhysRevLett.90.193601. URL <http://link.aps.org/doi/10.1103/PhysRevLett.90.193601>.
- [70] Isaac L. Chuang and M. A. Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, November 1997. ISSN 0950-0340. doi: 10.1080/09500349708231894. URL <http://www.tandfonline.com/doi/abs/10.1080/09500349708231894>.
- [71] G. Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, June 1976. ISSN 0010-3616, 1432-0916. doi: 10.1007/BF01608499. URL <http://link.springer.com/article/10.1007/BF01608499>.
- [72] A. Kossakowski. On quantum statistical mechanics of non-Hamiltonian systems. *Reports on Mathematical Physics*, 3(4):247–274, December 1972. ISSN 0034-4877. doi: 10.1016/0034-4877(72)90010-9. URL <http://www.sciencedirect.com/science/article/pii/0034487772900109>.
- [73] Wilhelm Magnus. On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics*, 7(4):649–673, 1954. ISSN 1097-0312. doi: 10.1002/cpa.3160070404. URL <http://onlinelibrary.wiley.com/doi/10.1002/cpa.3160070404/abstract>.
- [74] D. P. Burum and W. K. Rhim. Analysis of multiple pulse NMR in solids. III. *The Journal of Chemical Physics*, 71(2):944–956, July 1979. ISSN 00219606. doi: doi:10.

1063/1.438385. URL http://jcp.aip.org/resource/1/jcpa6/v71/i2/p944_s1?isAuthorized=no.

- [75] M. Matti Maricq. Application of average Hamiltonian theory to the NMR of solids. *Physical Review B*, 25(11):6622, June 1982. doi: 10.1103/PhysRevB.25.6622. URL <http://link.aps.org/doi/10.1103/PhysRevB.25.6622>.
- [76] Fernando Casas. Sufficient conditions for the convergence of the Magnus expansion. *Journal of Physics A: Mathematical and Theoretical*, 40(50):15001--15017, December 2007. ISSN 1751-8113, 1751-8121. doi: 10.1088/1751-8113/40/50/006. URL <http://arxiv.org/abs/0711.2381>. arXiv:0711.2381 [math].
- [77] Jon H. Shirley. *Interaction of a quantum system with a strong oscillating field*. PhD thesis, California Institute of Technology, January 1963. URL <http://thesis.library.caltech.edu/1805/>.
- [78] T. O. Levante, M. Baldus, B. H. Meier, and R. R. Ernst. Formalized quantum mechanical Floquet theory and its application to sample spinning in nuclear magnetic resonance. *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, 86(5):1195, 1995. ISSN 0026-8976. doi: 10.1080/00268979500102671. URL <http://www.informaworld.com/10.1080/00268979500102671>.
- [79] Michal Leskes, P.K. Madhu, and Shimon Vega. Floquet theory in solid-state nuclear magnetic resonance. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 57(4):345--380, November 2010. ISSN 0079-6565. doi: 10.1016/j.pnmrs.2010.06.002. URL <http://www.sciencedirect.com/science/article/B6THC-50DYH0S-1/2/52572a0c4adbd19ac8250b82ea773dd8>.
- [80] Ryogo Kubo. Generalized cumulant expansion method. *Journal of the Physical Society of Japan*, 17(7):1100--1120, July 1962. ISSN 0031-9015. doi: 10.1143/JPSJ.17.1100. URL <http://jpsj.ipap.jp/cgi-bin/getarticle?journal=JPSJ&volume=17&page=1100>.
- [81] P. Cappellaro, J. S Hodges, T. F Havel, and D. G Cory. Principles of control for decoherence-free subsystems. *The Journal of Chemical Physics*, 125(4):044514--044514--10, July 2006. ISSN 00219606. doi: doi:10.1063/1.2216702. URL http://jcp.aip.org/resource/1/jcpa6/v125/i4/p044514_s1.

- [82] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900–911, February 1997. doi: 10.1103/PhysRevA.55.900. URL <http://link.aps.org/doi/10.1103/PhysRevA.55.900>.
- [83] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Physical Review Letters*, 77(1):198–201, July 1996. doi: 10.1103/PhysRevLett.77.198. URL <http://link.aps.org/doi/10.1103/PhysRevLett.77.198>.
- [84] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296–305, February 2005. ISSN 1090-7807. doi: 10.1016/j.jmr.2004.11.004. URL <http://www.sciencedirect.com/science/article/pii/S1090780704003696>.
- [85] D. J. Egger and F. K. Wilhelm. Adaptive hybrid optimal quantum control for imprecisely characterized systems. *arXiv:1402.7193 [quant-ph]*, February 2014. URL <http://arxiv.org/abs/1402.7193>.
- [86] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. J. J. O’Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, A. N. Cleland, and John M. Martinis. Optimal quantum control using randomized benchmarking. *arXiv:1403.0035 [cond-mat, physics:quant-ph]*, February 2014. URL <http://arxiv.org/abs/1403.0035>.
- [87] B. P. Lanyon, C. Hempel, D. Nigg, M. Müller, R. Gerritsma, F. Zähringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, M. Hennrich, P. Zoller, R. Blatt, and C. F. Roos. Universal digital quantum simulation with trapped ions. *Science*, 334(6052):57–61, October 2011. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1208001. URL <http://www.sciencemag.org/content/334/6052/57>.
- [88] R. Gerritsma, B. P. Lanyon, G. Kirchmair, F. Zähringer, C. Hempel, J. Casanova, J. J. García-Ripoll, E. Solano, R. Blatt, and C. F. Roos. Quantum simulation of the Klein paradox with trapped ions. *Physical Review Letters*, 106(6):060503, February 2011. doi: 10.1103/PhysRevLett.106.060503. URL <http://link.aps.org/doi/10.1103/PhysRevLett.106.060503>.

- [89] K. Kim, M.-S. Chang, S. Korenblit, R. Islam, E. E. Edwards, J. K. Freericks, G.-D. Lin, L.-M. Duan, and C. Monroe. Quantum simulation of frustrated Ising spins with trapped ions. *Nature*, 465(7298):590–593, June 2010. ISSN 0028-0836. doi: 10.1038/nature09071. URL <http://www.nature.com/nature/journal/v465/n7298/full/nature09071.html>.
- [90] Matteo G. A. Paris and Jaroslav Rehacek, editors. *Quantum State Estimation*. 2004. URL <http://www.springer.com/physics/quantum+physics/book/978-3-540-22329-0>.
- [91] Ariel Bendersky, Fernando Pastawski, and Juan Pablo Paz. Selective and efficient estimation of parameters for quantum process tomography. *Physical Review Letters*, 100(19):190403, May 2008. doi: 10.1103/PhysRevLett.100.190403. URL <http://link.aps.org/doi/10.1103/PhysRevLett.100.190403>.
- [92] Ariel Bendersky, Fernando Pastawski, and Juan Pablo Paz. Selective and efficient quantum process tomography. *Physical Review A*, 80(3):032116, September 2009. doi: 10.1103/PhysRevA.80.032116. URL <http://link.aps.org/doi/10.1103/PhysRevA.80.032116>.
- [93] M. Mohseni and A. T. Rezakhani. Equation of motion for the process matrix: Hamiltonian identification and dynamical control of open quantum systems. *Physical Review A*, 80(1):010101, July 2009. doi: 10.1103/PhysRevA.80.010101. URL <http://link.aps.org/doi/10.1103/PhysRevA.80.010101>.
- [94] M. P. A. Branderhorst, J. Nunn, I. A. Walmsley, and R. L. Kosut. Simplified quantum process tomography. *New Journal of Physics*, 11(11):115010, November 2009. ISSN 1367-2630. doi: 10.1088/1367-2630/11/11/115010. URL <http://iopscience.iop.org/1367-2630/11/11/115010>.
- [95] Steven T. Flammia and Yi-Kai Liu. Direct fidelity estimation from few Pauli measurements. *Physical Review Letters*, 106(23):230501, June 2011. doi: 10.1103/PhysRevLett.106.230501. URL <http://link.aps.org/doi/10.1103/PhysRevLett.106.230501>.
- [96] F. E. Becerra, J. Fan, G. Baumgartner, J. Goldhar, J. T. Kosloski, and A. Migdall. Experimental demonstration of a receiver beating the standard quantum limit for multiple nonorthogonal state discrimination. *Nature Photonics*, 7(2):147–152, February 2013. ISSN 1749-4885. doi: 10.1038/nphoton.2012.260.

- nphoton.2012.316. URL <http://www.nature.com/nphoton/journal/v7/n2/full/nphoton.2012.316.html>.
- [97] Christopher Ferrie and Robin Blume-Kohout. Estimating the bias of a noisy coin. In *AIP Conference Proceedings*, volume 1443, pages 14--21. AIP Publishing, May 2012. doi: 10.1063/1.3703615. URL <http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.3703615>.
 - [98] Michael D. Shulman, Shannon P. Harvey, John M. Nichol, Stephen D. Bartlett, Andrew C. Doherty, Vladimir Umansky, and Amir Yacoby. Suppressing qubit dephasing using real-time Hamiltonian estimation. *arXiv:1405.0485 [cond-mat, physics:quant-ph]*, May 2014. URL <http://arxiv.org/abs/1405.0485>.
 - [99] Alexandr Sergeevich, Anushya Chandran, Joshua Combes, Stephen D. Bartlett, and Howard M. Wiseman. Characterization of a qubit Hamiltonian using adaptive measurements in a fixed basis. *Physical Review A*, 84(5):052315, November 2011. doi: 10.1103/PhysRevA.84.052315. URL <http://link.aps.org/doi/10.1103/PhysRevA.84.052315>.
 - [100] Alexandr Sergeevich, Joshua Combes, Anushya Chandran, Howard Wiseman, and Stephen Bartlett. Bayesian approach to Hamiltonian parameter estimation and measurement of double quantum dot. In *Proceedings of the International Quantum Electronics Conference and Conference on Lasers and Electro-Optics Pacific Rim 2011*, page I658. Optical Society of America, August 2011. doi: 10.1364/IQEC.2011.I658. URL <http://www.opticsinfobase.org/abstract.cfm?URI=IQEC-2011-I658>.
 - [101] S.G. Schirmer and F.C. Langbein. Quantum system identification: Hamiltonian estimation using spectral and Bayesian analysis. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 1--5, March 2010. doi: 10.1109/ISCCSP.2010.5463437.
 - [102] Sonia G. Schirmer and Daniel K. L. Oi. Two-qubit Hamiltonian tomography by Bayesian analysis of noisy data. *Physical Review A*, 80(2):022333, 2009. doi: 10.1103/PhysRevA.80.022333. URL <http://link.aps.org/doi/10.1103/PhysRevA.80.022333>.
 - [103] Alexandr Sergeevich and Stephen D. Bartlett. Optimizing qubit Hamiltonian parameter estimation algorithms using PSO. *arXiv:1206.3830*, June 2012. doi:

10.1109/CEC.2012.6252948. URL <http://arxiv.org/abs/1206.3830>. Proceedings of 2012 IEEE Conference on Evolutionary Computation (CEC), 10-15 June 2012.

- [104] Søren Gammelmark and Klaus Mølmer. Bayesian parameter inference from continuously monitored quantum systems. *Physical Review A*, 87(3):032115, March 2013. doi: 10.1103/PhysRevA.87.032115. URL <http://link.aps.org/doi/10.1103/PhysRevA.87.032115>.
- [105] James O. Berger and Robert L. Wolpert. The likelihood principle and generalizations. In *Institute of Mathematical Statistics Lecture Notes - Monograph Series*, pages 19--64. Institute of Mathematical Statistics, Hayward, CA, 1988. ISBN 0-940600-13-7. URL <http://projecteuclid.org/euclid.lnms/1215466214>.
- [106] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of Hilbert space. *Physical Review Letters*, 106(17):170501, April 2011. doi: 10.1103/PhysRevLett.106.170501. URL <http://link.aps.org/doi/10.1103/PhysRevLett.106.170501>.
- [107] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 2nd edition, August 1985. ISBN 0387960988.
- [108] T. M Cover and Joy A Thomas. *Elements of information theory*. Wiley-Interscience, Hoboken, N.J., 2006. ISBN 0471241954 9780471241959.
- [109] Robin Blume-Kohout and Patrick Hayden. Accurate quantum state estimation via "keeping the experimentalist honest". 2006. URL <http://arxiv.org/abs/quant-ph/0603116>.
- [110] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. Technical Report 463R, Department of Statistics, University of Washington, November 2005. URL <https://www.stat.washington.edu/research/reports/2004/list.shtml>.
- [111] Richard D. Gill and Boris Y. Levit. Applications of the Van Trees inequality: a Bayesian cramér-Rao bound. *Bernoulli*, 1(1):59--79, March 1995. ISSN 1350-7265. URL <http://projecteuclid.org/euclid.bj/1186078362>. Mathematical Reviews number (MathSciNet): MR1354456.

- [112] Harry L Van Trees and Kristine L Bell. *Detection estimation and modulation theory. Part 1 Part 1*. Wiley-Blackwell, Oxford, 2 edition, 1968. ISBN 9780470542965 0470542969.
- [113] J. Dauwels. Computing Bayesian Cramer-Rao bounds. In *International Symposium on Information Theory, 2005. ISIT 2005. Proceedings*, pages 425--429. IEEE, September 2005. ISBN 0-7803-9151-9. doi: 10.1109/ISIT.2005.1523369.
- [114] P. Tichavsky, C.H. Muravchik, and Arye Nehorai. Posterior Cramer-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5): 1386--1396, 1998. ISSN 1053-587X. doi: 10.1109/78.668800.
- [115] Michael J. W. Hall and Howard M. Wiseman. Does nonlinear metrology offer improved resolution? answers from quantum information theory. *Physical Review X*, 2(4):041006, October 2012. doi: 10.1103/PhysRevX.2.041006. URL <http://link.aps.org/doi/10.1103/PhysRevX.2.041006>.
- [116] Richard R. Ernst, Geoffrey Bodenhausen, and Alexander Wokaun. *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*. Number 14 in International Series of Monographs on Chemistry. Oxford University Press, USA, September 1990. ISBN 0198556470.
- [117] H. Nyquist. Certain topics in telegraph transmission theory. *American Institute of Electrical Engineers, Transactions of the*, 47(2):617--644, April 1928. ISSN 0096-3860. doi: 10.1109/T-AIEE.1928.5055024.
- [118] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *STATISTICS AND COMPUTING*, 10(3):197--208, 2000.
- [119] Miodrag Bolic. *Architectures for Efficient Implementation of Particle Filters*. PhD thesis, State University of New York at Stony Brook, Stony Brook, NY, USA, 2004. AAI3149104.
- [120] Cathal Ó Broin and L. A. A. Nikolopoulos. A GPGPU based program to solve the TDSE in intense laser fields through the finite difference approach. arXiv e-print 1308.1856, August 2013. URL <http://arxiv.org/abs/1308.1856>.
- [121] Eladio Gutierrez, Sergio Romero, Maria Trenas, and Emilio Zapata. Parallel quantum computer simulation on the CUDA architecture. In Marian Bubak, Geert van Albada, Jack Dongarra, and Peter Sloot, editors, *Computational Science – ICCS 2008*,

volume 5101 of *Lecture Notes in Computer Science*, pages 700--709. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-69383-3. URL <http://www.springerlink.com/content/wq28572x1w181182/abstract/>.

- [122] A.U. Khalid, Z. Zilic, and K. Radecka. FPGA emulation of quantum circuits. In *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, pages 310 -- 315, October 2004. doi: 10.1109/ICCD.2004.1347938.
- [123] Steven Casagrande. On design and testing of a spectrometer based on an FPGA development board for use with optimal control theory and high-Q resonators. February 2014. URL <https://uwspace.uwaterloo.ca/handle/10012/8281>.
- [124] J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955--993, April 2000. ISSN 0899-7667. doi: 10.1162/089976600300015664. URL <http://dx.doi.org/10.1162/089976600300015664>.
- [125] Nicolas Courty and Elise Arnaud. Inverse kinematics using sequential Monte Carlo methods. In Francisco J. Perales and Robert B. Fisher, editors, *Articulated Motion and Deformable Objects*, number 5098 in Lecture Notes in Computer Science, pages 1--10. Springer Berlin Heidelberg, January 2008. ISBN 978-3-540-70516-1, 978-3-540-70517-8. URL http://link.springer.com/chapter/10.1007/978-3-540-70517-8_1.
- [126] Drew Creal. A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, 31(3):245--296, October 2011. ISSN 0747-4938. doi: 10.1080/07474938.2011.607333. URL <http://dx.doi.org/10.1080/07474938.2011.607333>.
- [127] Markku P. V. Stenberg, Yuval R. Sanders, and Frank K. Wilhelm. Efficient estimation of resonant coupling between quantum systems. *Physical Review Letters*, 113(21):210404, November 2014. doi: 10.1103/PhysRevLett.113.210404. URL <http://link.aps.org/doi/10.1103/PhysRevLett.113.210404>.
- [128] Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278--288, March 1994. doi: 10.2307/2291224. URL <http://www.jstor.org/stable/2291224>.

- [129] Jane Liu and Mike West. Combined parameter and state estimation in simulation-based filtering. In De Freitas and NJ Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001. URL <http://ftp.stat.duke.edu/WorkingPapers/99-14.html>.
- [130] Mike West. Approximating posterior distributions by mixture. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(2):409–422, January 1993. ISSN 0035-9246. URL <http://www.jstor.org/stable/2346202>.
- [131] Nathan Wiebe, Ashish Kapoor, and Krysta Svore. Quantum nearest-neighbor algorithms for machine learning. *arXiv:1401.2142 [quant-ph]*, January 2014. URL <http://arxiv.org/abs/1401.2142>.
- [132] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113(6), April 1993. ISSN 0956-375X. URL <http://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0015>.
- [133] Michael Isard and Andrew Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998. ISSN 0920-5691, 1573-1405. doi: 10.1023/A:1008078328650. URL <http://link.springer.com/article/10.1023/A%3A1008078328650>.
- [134] Arnaud Doucet, Nando de Freitas, and Neil Gordon. An introduction to sequential Monte Carlo methods. In Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 3–14. Springer New York, January 2001. ISBN 978-1-4419-2887-0, 978-1-4757-3437-9. URL http://link.springer.com/chapter/10.1007/978-1-4757-3437-9_1.
- [135] A. Jasra and A. Doucet. Sequential Monte Carlo methods for diffusion processes. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2112):3709–3727, 2009. ISSN 1471-2946. doi: 10.1098/rspa.2009.0206.
- [136] Sarah (Sarah Elizabeth) Sheldon. *Optimal control in an open quantum system : selecting DNP pathways in an electron-nuclear system*. Thesis, Massachusetts Institute of Technology, 2013. URL <http://dspace.mit.edu/handle/1721.1/82867>. Thesis (Ph. D.)--Massachusetts Institute of Technology, Department of Nuclear Science and Engineering, 2013.

- [137] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL <http://www.scipy.org/>.
- [138] W. P. Aue, E. Bartholdi, and R. R. Ernst. Two-dimensional spectroscopy. application to nuclear magnetic resonance. *The Journal of Chemical Physics*, 64(5):2229–2246, March 1976. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.432450. URL <http://scitation.aip.org/content/aip/journal/jcp/64/5/10.1063/1.432450>.
- [139] E.D Laue, M.R Mayger, J Skilling, and J Staunton. Reconstruction of phase-sensitive two-dimensional NMR spectra by maximum entropy. *Journal of Magnetic Resonance* (1969), 68(1):14–29, June 1986. ISSN 0022-2364. doi: 10.1016/0022-2364(86)90312-4. URL <http://www.sciencedirect.com/science/article/pii/0022236486903124>.
- [140] Sven G Hyberts, Haribabu Arthanari, and Gerhard Wagner. Applications of non-uniform sampling and processing. *Topics in Current Chemistry*, July 2011. ISSN 0340-1022. doi: 10.1007/128_2011_187. URL <http://www.ncbi.nlm.nih.gov/pubmed/21796515>.
- [141] A. Gruber, A. Dräbenstedt, C. Tietz, L. Fleury, J. Wrachtrup, and C. von Borczyskowski. Scanning confocal optical microscopy and magnetic resonance on single defect centers. *Science*, 276(5321):2012–2014, June 1997. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.276.5321.2012. URL <http://www.sciencemag.org/content/276/5321/2012>.
- [142] Osama Moussa. personal communication.
- [143] Joseph Emerson, Robert Alicki, and Karol Zyczkowski. Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S347–S352, 2005. ISSN 1464-4266. URL <http://www.iop.org/EJ/abstract/1464-4266/7/10/021/>.
- [144] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. Randomized benchmarking of quantum gates. *Physical Review A*, 77(1):012307, January 2008. doi: 10.1103/PhysRevA.77.012307. URL <http://link.aps.org/doi/10.1103/PhysRevA.77.012307>.
- [145] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Physical Review A*, 85(4), April 2012. ISSN

1050-2947, 1094-1622. doi: 10.1103/PhysRevA.85.042311. URL <http://arxiv.org/abs/1109.6887>.

- [146] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and Approximate Unitary 2-Designs: Constructions and Applications. *arXiv:quant-ph/0606161*, June 2006. URL <http://arxiv.org/abs/quant-ph/0606161>. Physical Review A 80, 012304 (2009).
- [147] Joseph Emerson, Marcus Silva, Osama Moussa, Colm Ryan, Martin Laforest, Jonathan Baugh, David G. Cory, and Raymond Laflamme. Symmetrized Characterization of Noisy Quantum Processes. *Science*, 317(5846):1893–1896, September 2007. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1145699. URL <http://www.sciencemag.org/content/317/5846/1893>.
- [148] Robert Koenig and John A. Smolin. How to efficiently select an arbitrary Clifford group element. *arXiv:1406.2170 [quant-ph]*, June 2014. URL <http://arxiv.org/abs/1406.2170>. arXiv: 1406.2170.
- [149] Daniel Gottesman. *Surviving as a Quantum Computer in a Classical World*. 2016.
- [150] C. A. Ryan, M. Laforest, and R. Laflamme. Randomized benchmarking of single- and multi-qubit control in liquid-state NMR quantum information processing. *New Journal of Physics*, 11(1):013034, January 2009. ISSN 1367-2630. doi: 10.1088/1367-2630/11/1/013034. URL <http://iopscience.iop.org/1367-2630/11/1/013034>.
- [151] J. M. Chow, J. M. Gambetta, L. Tornberg, Jens Koch, Lev S. Bishop, A. A. Houck, B. R. Johnson, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Randomized benchmarking and process tomography for gate errors in a solid-state qubit. *Physical Review Letters*, 102(9):090502, March 2009. doi: 10.1103/PhysRevLett.102.090502. URL <http://link.aps.org/doi/10.1103/PhysRevLett.102.090502>.
- [152] S. Olmschenk, R. Chicireanu, K. D. Nelson, and J. V. Porto. Randomized benchmarking of atomic qubits in an optical lattice. *New Journal of Physics*, 12(11):113007, November 2010. ISSN 1367-2630. doi: 10.1088/1367-2630/12/11/113007. URL <http://iopscience.iop.org/1367-2630/12/11/113007>.
- [153] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland. Single-qubit-gate error below 10^{-4} in a trapped ion. *Physical Review A*, 84(3):030303, September 2011. doi: 10.

1103/PhysRevA.84.030303. URL <http://link.aps.org/doi/10.1103/PhysRevA.84.030303>.

- [154] Osama Moussa, Marcus P. da Silva, Colm A. Ryan, and Raymond Laflamme. Practical experimental certification of computational quantum gates using a twirling procedure. *Physical Review Letters*, 109(7):070504, August 2012. doi: 10.1103/PhysRevLett.109.070504. URL <http://link.aps.org/doi/10.1103/PhysRevLett.109.070504>.
- [155] T. R. Tan, J. P. Gaebler, R. Bowler, Y. Lin, J. D. Jost, D. Leibfried, and D. J. Wineland. Demonstration of a dressed-state phase gate for trapped ions. *Physical Review Letters*, 110(26):263002, June 2013. doi: 10.1103/PhysRevLett.110.263002. URL <http://link.aps.org/doi/10.1103/PhysRevLett.110.263002>.
- [156] A. D. Córcoles, Jay M. Gambetta, Jerry M. Chow, John A. Smolin, Matthew Ware, Joel Strand, B. L. T. Plourde, and M. Steffen. Process verification of two-qubit quantum gates by randomized benchmarking. *Physical Review A*, 87(3):030301, March 2013. doi: 10.1103/PhysRevA.87.030301. URL <http://link.aps.org/doi/10.1103/PhysRevA.87.030301>.
- [157] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O’Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and John M. Martinis. Logic gates at the surface code threshold: Superconducting qubits poised for fault-tolerant quantum computing. *arXiv:1402.4848 [cond-mat, physics:quant-ph]*, February 2014. URL <http://arxiv.org/abs/1402.4848>.
- [158] Jay M. Gambetta, A. D. Córcoles, S. T. Merkel, B. R. Johnson, John A. Smolin, Jerry M. Chow, Colm A. Ryan, Chad Rigetti, S. Poletto, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Characterization of addressability by simultaneous randomized benchmarking. *Physical Review Letters*, 109(24):240504, December 2012. doi: 10.1103/PhysRevLett.109.240504. URL <http://link.aps.org/doi/10.1103/PhysRevLett.109.240504>.
- [159] Simon Gustavsson, Olger Zwier, Jonas Bylander, Fei Yan, Fumiki Yoshihara, Yasunobu Nakamura, Terry P. Orlando, and William D. Oliver. Improving quantum gate fidelities by using a qubit to measure microwave pulse distortions. *Physical Review Letters*, 110(4):040502, January 2013. doi: 10.1103/PhysRevLett.110.040502. URL <http://link.aps.org/doi/10.1103/PhysRevLett.110.040502>.

- [160] P. J. J. O’Malley, J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, A. N. Korotkov, A. N. Cleland, and John M. Martinis. Overcoming correlated noise in quantum systems: How mediocre clocks make good qubits. *SciRate*, November 2014. URL <https://scirate.com/arxiv/1411.2613>.
- [161] Joel J. Wallman and Steven T. Flammia. Randomized benchmarking with confidence. *arXiv:1404.6025 [quant-ph]*, April 2014. URL <http://arxiv.org/abs/1404.6025>.
- [162] Shelby Kimmel, Marcus P. da Silva, Colm A. Ryan, Blake R. Johnson, and Thomas Ohki. Robust extraction of tomographic information via randomized benchmarking. *Physical Review X*, 4(1):011050, March 2014. doi: 10.1103/PhysRevX.4.011050. URL <http://link.aps.org/doi/10.1103/PhysRevX.4.011050>.
- [163] Easwar Magesan, Jay M. Gambetta, B. R. Johnson, Colm A. Ryan, Jerry M. Chow, Seth T. Merkel, Marcus P. da Silva, George A. Keefe, Mary B. Rothwell, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Physical Review Letters*, 109(8):080505, August 2012. doi: 10.1103/PhysRevLett.109.080505. URL <http://link.aps.org/doi/10.1103/PhysRevLett.109.080505>.
- [164] Daniel Puzzuoli, Christopher Granade, Holger Haas, Ben Criger, Easwar Magesan, and D. G. Cory. Tractable simulation of error correction with honest approximations to realistic fault models. *Physical Review A*, 89(2):022306, February 2014. doi: 10.1103/PhysRevA.89.022306. URL <http://link.aps.org/doi/10.1103/PhysRevA.89.022306>.
- [165] Christopher Ferrie and Christopher E. Granade. Likelihood-free methods for quantum parameter estimation. *Physical Review Letters*, 112(13):130402, April 2014. doi: 10.1103/PhysRevLett.112.130402. URL <http://link.aps.org/doi/10.1103/PhysRevLett.112.130402>.
- [166] P. Cappellaro, L. Jiang, J. S. Hodges, and M. D. Lukin. Coherence and control of quantum registers based on electronic spin in a nuclear spin bath. *Physical Review Letters*, 102(21):210502, May 2009. doi: 10.1103/PhysRevLett.102.210502. URL <http://link.aps.org/doi/10.1103/PhysRevLett.102.210502>.

- [167] R. S. Said, D. W. Berry, and J. Twamley. Nanoscale magnetometry using a single-spin system in diamond. *Physical Review B*, 83(12):125410, March 2011. doi: 10.1103/PhysRevB.83.125410. URL <http://link.aps.org/doi/10.1103/PhysRevB.83.125410>.
- [168] N. M. Nusran and M. V. Gurudev Dutt. Optimizing phase estimation algorithms for diamond spin magnetometry. *arXiv:1403.4506 [quant-ph]*, March 2014. URL <http://arxiv.org/abs/1403.4506>.
- [169] A. Cooper, E. Magesan, H. N. Yum, and P. Cappellaro. Time-resolved magnetic sensing with electronic spins in diamond. *Nature Communications*, 5, January 2014. doi: 10.1038/ncomms4141. URL <http://www.nature.com/ncomms/2014/140124/ncomms4141/full/ncomms4141.html>.
- [170] B. J. Maertz, A. P. Wijnheijmer, G. D. Fuchs, M. E. Nowakowski, and D. D. Awschalom. Vector magnetic field microscopy using nitrogen vacancy centers in diamond. *Applied Physics Letters*, 96(9):092504, March 2010. ISSN 0003-6951, 1077-3118. doi: 10.1063/1.3337096. URL <http://scitation.aip.org/content/aip/journal/apl/96/9/10.1063/1.3337096>.
- [171] N. B. Manson, J. P. Harrison, and M. J. Sellars. Nitrogen-vacancy center in diamond: Model of the electronic structure and associated dynamics. *Physical Review B*, 74(10):104303, 2006. doi: 10.1103/PhysRevB.74.104303. URL <http://link.aps.org/doi/10.1103/PhysRevB.74.104303>.
- [172] Marcus W. Doherty, Neil B. Manson, Paul Delaney, Fedor Jelezko, Jörg Wrachtrup, and Lloyd C.L. Hollenberg. The nitrogen-vacancy colour centre in diamond. *Physics Reports*, 528(1):1–45, July 2013. ISSN 0370-1573. doi: 10.1016/j.physrep.2013.02.001. URL <http://www.sciencedirect.com/science/article/pii/S0370157313000562>.
- [173] Om Patange. *On an Instrument for the Coherent Investigation of Nitrogen-Vacancy Centres in Diamond*. PhD thesis, September 2013. URL <https://uwspace.uwaterloo.ca/handle/10012/7955>.
- [174] Ian Hincks, D. G. Cory, et al. *NVSim*. 2014.
- [175] Dennis Wackerly, William Mendenhall, and Richard L. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Press, 6 edition, May 2001. ISBN 0534377416.

- [176] Ward Edwards, Harold Lindman, and Leonard J. Savage. Bayesian statistical inference for psychological research. *Psychological Review*, 70(3):193–242, 1963. ISSN 1939-1471(Electronic);0033-295X(Print). doi: 10.1037/h0044139.
- [177] Steven Goodman. A dirty dozen: Twelve p-value misconceptions. *Seminars in Hematology*, 45(3):135–140, July 2008. ISSN 0037-1963. doi: 10.1053/j.seminhematol.2008.04.003. URL <http://www.sciencedirect.com/science/article/pii/S0037196308000620>.
- [178] L. A. Harvey. Statistical power calculations reflect our love affair with p-values and hypothesis testing: time for a fundamental change. *Spinal Cord*, 52(1):2–2, January 2014. ISSN 1362-4393. doi: 10.1038/sc.2013.117. URL <http://www.nature.com/sc/journal/v52/n1/full/sc2013117a.html>.
- [179] Rink Hoekstra, Richard D. Morey, Jeffrey N. Rouder, and Eric-Jan Wagenmakers. Robust misinterpretation of confidence intervals. *Psychonomic Bulletin & Review*, pages 1–8, January 2014. ISSN 1069-9384, 1531-5320. doi: 10.3758/s13423-013-0572-3. URL <http://link.springer.com/article/10.3758/s13423-013-0572-3>.
- [180] Regina Nuzzo. Scientific method: Statistical errors. *Nature*, 506(7487):150–152, February 2014. ISSN 0028-0836, 1476-4687. doi: 10.1038/506150a. URL <http://www.nature.com/doifinder/10.1038/506150a>.
- [181] S. J. van Enk and Robin Blume-Kohout. When quantum tomography goes wrong: drift of quantum sources and other errors. *New Journal of Physics*, 15(2):025024, February 2013. ISSN 1367-2630. doi: 10.1088/1367-2630/15/2/025024. URL <http://iopscience.iop.org/1367-2630/15/2/025024>.
- [182] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974. ISSN 0018-9286. doi: 10.1109/TAC.1974.1100705.
- [183] J. O. S. Yin and S. J. van Enk. Information criteria for efficient quantum state estimation. *Physical Review A*, 83(6):062110, June 2011. doi: 10.1103/PhysRevA.83.062110. URL <http://link.aps.org/doi/10.1103/PhysRevA.83.062110>.
- [184] Pavel Lougovski and S. J. van Enk. Characterizing entanglement sources. *Physical Review A*, 80(5):052324, November 2009. doi: 10.1103/PhysRevA.80.052324. URL <http://link.aps.org/doi/10.1103/PhysRevA.80.052324>.

- [185] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, March 1978. ISSN 0090-5364. doi: 10.1214/aos/1176344136. URL <http://projecteuclid.org/euclid-aos/1176344136>. Mathematical Reviews number (MathSciNet): MR468014; Zentralblatt MATH identifier: 0379.62005.
- [186] Hirotugu Akaike. Likelihood and the Bayes procedure. *Trabajos de Estadistica Y de Investigacion Operativa*, 31(1):143–166, February 1980. ISSN 0041-0241. doi: 10.1007/BF02888350. URL <http://link.springer.com/article/10.1007/BF02888350>.
- [187] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, January 1933. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.1933.0009. URL <http://rsta.royalsocietypublishing.org/content/231/694-706/289>.
- [188] Lucia Schwarz and S. J. van Enk. Error models in quantum computation: An application of model selection. *Physical Review A*, 88(3):032318, September 2013. doi: 10.1103/PhysRevA.88.032318. URL <http://link.aps.org/doi/10.1103/PhysRevA.88.032318>.
- [189] Dmitry A Pushin. *Coherent control of neutron interferometry*. PhD thesis, Massachusetts Institute of Technology, 2007. URL <http://hdl.handle.net/1721.1/39292>.
- [190] D. A. Pushin, M. Arif, M. G. Huber, and D. G. Cory. Measurements of the vertical coherence length in neutron interferometry. *Physical Review Letters*, 100(25):250404, June 2008. doi: 10.1103/PhysRevLett.100.250404. URL <http://link.aps.org/doi/10.1103/PhysRevLett.100.250404>.
- [191] Peter W Shor. Fault-tolerant quantum computation. *arXiv:quant-ph/9605011*, May 1996. URL <http://arxiv.org/abs/quant-ph/9605011>.
- [192] Austin G. Fowler, Ashley M. Stephens, and Peter Groszkowski. High-threshold universal quantum computation on the surface code. *Physical Review A*, 80(5):052312, November 2009. doi: 10.1103/PhysRevA.80.052312. URL <http://link.aps.org/doi/10.1103/PhysRevA.80.052312>.
- [193] Yuval R. Sanders, Joel J. Wallman, and Barry C. Sanders. Bounding Quantum Gate Error Rate Based on Reported Gate Fidelity. *arXiv:1501.04932 [quant-ph]*, January 2015. URL <http://arxiv.org/abs/1501.04932>. arXiv: 1501.04932.

- [194] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002. ISBN 0821832298. URL <http://portal.acm.org/citation.cfm?id=863284>.
- [195] Christopher A Fuchs and Jeroen van de Graaf. Cryptographic distinguishability measures for quantum mechanical states. *quant-ph/9712042*, December 1997. URL <http://arxiv.org/abs/quant-ph/9712042>.
- [196] John Watrous. Simpler semidefinite programs for completely bounded norms. *arXiv:1207.5726*, July 2012. URL <http://arxiv.org/abs/1207.5726>.
- [197] Michael Grant and Stephen Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. March 2014. URL <http://cvxr.com/cvx>.
- [198] M. Van den Nest. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. arXiv e-print 0811.0898, November 2008. URL <http://arxiv.org/abs/0811.0898>. Quant. Inf. Comp. 10, 3-4 pp. pp0258-0271 (2010).
- [199] M. Van den Nest. Simulating quantum computers with probabilistic methods. arXiv e-print 0911.1624, November 2009. URL <http://arxiv.org/abs/0911.1624>. Quant. Inf. Comp. 11, 9-10 pp. 784-812 (2011).
- [200] Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, January 2002. ISSN 0097-5397, 1095-7111. doi: 10.1137/S0097539700377025. URL <http://pubs.siam.org/doi/abs/10.1137/S0097539700377025?journalCode=smjcat&volume=31&issue=4>.
- [201] Mark Howard, Joel Wallman, Victor Veitch, and Joseph Emerson. Contextuality supplies the 'magic' for quantum computation. *Nature*, 510(7505):351–355, June 2014. ISSN 0028-0836. doi: 10.1038/nature13460. URL <http://www.nature.com/nature/journal/v510/n7505/full/nature13460.html>.
- [202] Evan M. Fortunato, Marco A. Pravia, Nicolas Boulant, Grum Teklemariam, Timothy F. Havel, and David G. Cory. Design of strongly modulating pulses to implement precise effective hamiltonians for quantum information processing. *The Journal of Chemical Physics*, 116(17):7599, 2002. ISSN 00219606. doi: 10.1063/1.1465412. URL <http://link.aip.org/link/JCPA6/v116/i17/p7599/s1&Agg=doi>.

- [203] J. S. Hodges, J. C. Yang, C. Ramanathan, and D. G. Cory. Universal control of nuclear spins via anisotropic hyperfine interactions. *Physical Review A*, 78(1):010303, July 2008. doi: 10.1103/PhysRevA.78.010303. URL <http://link.aps.org/doi/10.1103/PhysRevA.78.010303>.
- [204] H. J. Mamin, M. Kim, M. H. Sherwood, C. T. Rettner, K. Ohno, D. D. Awschalom, and D. Rugar. Nanoscale nuclear magnetic resonance with a nitrogen-vacancy spin sensor. *Science*, 339(6119):557–560, February 2013. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1231540. URL <http://www.sciencemag.org/content/339/6119/557>.
- [205] J. M. Taylor, P. Cappellaro, L. Childress, L. Jiang, D. Budker, P. R. Hemmer, A. Yacoby, R. Walsworth, and M. D. Lukin. High-sensitivity diamond magnetometer with nanoscale resolution. *Nature Physics*, 4(10):810–816, October 2008. ISSN 1745-2473. doi: 10.1038/nphys1075. URL <http://www.nature.com/nphys/journal/v4/n10/abs/nphys1075.html>.
- [206] C. Ryan, C. Negrevergne, M. Laforest, E. Knill, and R. Laflamme. Liquid-state nuclear magnetic resonance as a testbed for developing quantum control methods. *Physical Review A*, 78(1):012328, July 2008. doi: 10.1103/PhysRevA.78.012328. URL <http://link.aps.org/doi/10.1103/PhysRevA.78.012328>.
- [207] F. Motzoi, J. Gambetta, P. Rebentrost, and F. Wilhelm. Simple pulses for elimination of leakage in weakly nonlinear qubits. *Physical Review Letters*, 103(11):110501, September 2009. doi: 10.1103/PhysRevLett.103.110501. URL <http://link.aps.org/doi/10.1103/PhysRevLett.103.110501>.
- [208] G. Passante, O. Moussa, D. Trottier, and R. Laflamme. Experimental detection of nonclassical correlations in mixed-state quantum computation. *Physical Review A*, 84(4):044302, October 2011. doi: 10.1103/PhysRevA.84.044302. URL <http://link.aps.org/doi/10.1103/PhysRevA.84.044302>.
- [209] D. I. Hoult. Fast recovery, high sensitivity NMR probe and preamplifier for low frequencies. *Review of Scientific Instruments*, 50(2):193–200, February 1979. ISSN 0034-6748, 1089-7623. doi: 10.1063/1.1135786. URL <http://scitation.aip.org/content/aip/journal/rsi/50/2/10.1063/1.1135786>.
- [210] Troy W. Borneman and David G. Cory. Bandwidth-limited control and ringdown suppression in high-Q resonators. *Journal of Magnetic Resonance*, 225:120–129, December 2012. ISSN 1090-7807. doi: 10.1016/j.jmr.2012.10.011. URL <http://www.sciencedirect.com/science/article/pii/S1090780712003333>.

- [211] H. Malissa, D. I. Schuster, A. M. Tyryshkin, A. A. Houck, and S. A. Lyon. Superconducting coplanar waveguide resonators for low temperature pulsed electron spin resonance spectroscopy. *arXiv:1202.6305 [cond-mat]*, February 2012. URL <http://arxiv.org/abs/1202.6305>. arXiv: 1202.6305.
- [212] O.W.B. Benningshof, H.R. Mohebbi, I.A.J. Taminiau, G.X. Miao, and D.G. Cory. Superconducting microstrip resonator for pulsed ESR of thin films. *Journal of Magnetic Resonance*, 230:84–87, May 2013. ISSN 1090-7807. doi: 10.1016/j.jmr.2013.01.010. URL <http://www.sciencedirect.com/science/article/pii/S1090780713000244>.
- [213] H. R. Mohebbi, O. W. B. Benningshof, I. a. J. Taminiau, G. X. Miao, and D. G. Cory. Composite arrays of superconducting microstrip line resonators. *Journal of Applied Physics*, 115(9):094502, March 2014. ISSN 0021-8979, 1089-7550. doi: 10.1063/1.4866691. URL <http://scitation.aip.org/content/aip/journal/jap/115/9/10.1063/1.4866691>.
- [214] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquières, A. Gruslys, S. Schirmer, and T. Schulte-Herbrüggen. Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework. *Physical Review A*, 84(2):022305, August 2011. doi: 10.1103/PhysRevA.84.022305. URL <http://link.aps.org/doi/10.1103/PhysRevA.84.022305>.
- [215] Michael Tinkham. *Introduction to Superconductivity: Second Edition*. Dover Publications, Mineola, N.Y., second edition edition edition, June 2004. ISBN 9780486435039.
- [216] Maas. *Nonlinear Microwave and RF Circuits*. Artech House Publishers, Boston, MA, 2 edition edition, February 2003. ISBN 9781580534840.
- [217] V. Rizzoli, C. Cecchetti, Alessandro Lipparini, and F. Mastri. General-purpose harmonic balance analysis of nonlinear microwave circuits under multitone excitation. *IEEE Transactions on Microwave Theory and Techniques*, 36(12):1650–1660, December 1988. ISSN 0018-9480. doi: 10.1109/22.17396.
- [218] Troy W. Borneman, Martin D. Hürlimann, and David G. Cory. Application of optimal control to CPMG refocusing pulse design. *Journal of Magnetic Resonance*, In Press, Corrected Proof, December 2010. ISSN 1090-7807. doi: 10.1016/j.jmr.2010.09.003. URL <http://www.sciencedirect.com/science/article/B6WJX-511G1Y5-2/2/dbb8975ea0ab760e5bd8a26ce6ee2ab1>.

- [219] Thomas E. Skinner, Michael Braun, Klaus Woelk, Naum I. Gershenzon, and Steffen J. Glaser. Design and application of robust rf pulses for toroid cavity NMR spectroscopy. *Journal of Magnetic Resonance*, 209(2):282–290, April 2011. ISSN 1090-7807. doi: 10.1016/j.jmr.2011.01.026. URL <http://www.sciencedirect.com/science/article/pii/S1090780711000425>.
- [220] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, August 1997. ISSN 0934-4373, 1615-1488. doi: 10.1007/BF01197559. URL <http://link.springer.com/article/10.1007/BF01197559>.
- [221] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, May 1998. ISSN 0028-0836. doi: 10.1038/30156. URL <http://dx.doi.org/10.1038/30156>.
- [222] W. Harneit, C. Meyer, A. Weidinger, D. Suter, and J. Twamley. Architectures for a spin quantum computer based on endohedral fullerenes. *physica status solidi (b)*, 233(3):453–461, October 2002. ISSN 1521-3951. doi: 10.1002/1521-3951(200210)233:3<453::AID-PSSB453>3.0.CO;2-N. URL [http://onlinelibrary.wiley.com/doi/10.1002/1521-3951\(200210\)233:3<453::AID-PSSB453>3.0.CO;2-N/abstract](http://onlinelibrary.wiley.com/doi/10.1002/1521-3951(200210)233:3<453::AID-PSSB453>3.0.CO;2-N/abstract).
- [223] J. Twamley. Quantum-cellular-automata quantum computing with endohedral fullerenes. *Physical Review A*, 67(5):052318, May 2003. doi: 10.1103/PhysRevA.67.052318. URL <http://link.aps.org/doi/10.1103/PhysRevA.67.052318>.
- [224] Michael Mehring and Jens Mende. Spin-bus concept of spin quantum computing. *Physical Review A*, 73(5):052303, May 2006. doi: 10.1103/PhysRevA.73.052303. URL <http://link.aps.org/doi/10.1103/PhysRevA.73.052303>.
- [225] Liang Jiang, Jacob M. Taylor, Anders S. Sørensen, and Mikhail D. Lukin. Distributed quantum computation based on small quantum registers. *Physical Review A*, 76(6):062323, December 2007. doi: 10.1103/PhysRevA.76.062323. URL <http://link.aps.org/doi/10.1103/PhysRevA.76.062323>.
- [226] Naomi H. Nickerson, Joseph F. Fitzsimons, and Simon C. Benjamin. Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy

- photonic links. *arXiv:1406.0880 [quant-ph]*, June 2014. URL <http://arxiv.org/abs/1406.0880>. arXiv: 1406.0880.
- [227] J. S Waugh. Theory of broadband spin decoupling. *Journal of Magnetic Resonance (1969)*, 50(1):30–49, October 1982. ISSN 0022-2364. doi: 10.1016/0022-2364(82)90029-4. URL <http://www.sciencedirect.com/science/article/pii/0022236482900294>.
 - [228] S. Meiboom and D. Gill. Modified spin-echo method for measuring nuclear relaxation times. *Review of Scientific Instruments*, 29(8):688–691, August 1958. ISSN 0034-6748, 1089-7623. doi: 10.1063/1.1716296. URL <http://scitation.aip.org/content/aip/journal/rsi/29/8/10.1063/1.1716296>.
 - [229] J. S. Waugh, L. M. Huber, and U. Haeberlen. Approach to high-resolution nmr in solids. *Physical Review Letters*, 20(5):180–182, January 1968. doi: 10.1103/PhysRevLett.20.180. URL <http://link.aps.org/doi/10.1103/PhysRevLett.20.180>.
 - [230] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *arXiv:1111.4022*, November 2011. URL <http://arxiv.org/abs/1111.4022>.
 - [231] Andrew J. Landahl and Ciaran Ryan-Anderson. Quantum computing by color-code lattice surgery. *SciRate*, July 2014. URL <https://scirate.com/arxiv/1407.5103>.
 - [232] Andrew M. Childs and Nathan Wiebe. Product formulas for exponentials of commutators. *Journal of Mathematical Physics*, 54(6):062202, June 2013. ISSN 0022-2488. doi: 10.1063/1.4811386. URL <http://scitation.aip.org/content/aip/journal/jmp/54/6/10.1063/1.4811386>.
 - [233] J. L. Walsh. A closed set of normal orthogonal functions. *American Journal of Mathematics*, 45(1):5–24, January 1923. ISSN 0002-9327. doi: 10.2307/2387224. URL <http://www.jstor.org/stable/2387224>.
 - [234] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput.*, 12(11-12):901–924, November 2012. ISSN 1533-7146. URL <http://dl.acm.org/citation.cfm?id=2481569.2481570>.

- [235] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. *arXiv:1312.1414 [quant-ph]*, December 2013. doi: 10.1145/2591796.2591854. URL <http://arxiv.org/abs/1312.1414>. arXiv: 1312.1414.
- [236] R. Islam, C. Senko, W. C. Campbell, S. Korenblit, J. Smith, A. Lee, E. E. Edwards, C.-C. J. Wang, J. K. Freericks, and C. Monroe. Emergence and frustration of magnetism with variable-range interactions in a quantum simulator. *Science*, 340(6132):583–587, May 2013. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1232296. URL <http://www.sciencemag.org/content/340/6132/583>.
- [237] Jonathan Simon, Waseem S. Bakr, Ruichao Ma, M. Eric Tai, Philipp M. Preiss, and Markus Greiner. Quantum simulation of antiferromagnetic spin chains in an optical lattice. *Nature*, 472(7343):307–312, April 2011. ISSN 0028-0836. doi: 10.1038/nature09994. URL <http://www.nature.com/nature/journal/v472/n7343/full/nature09994.html>.
- [238] Joseph W. Britton, Brian C. Sawyer, Adam C. Keith, C.-C. Joseph Wang, James K. Freericks, Hermann Uys, Michael J. Biercuk, and John J. Bollinger. Engineered two-dimensional Ising interactions in a trapped-ion quantum simulator with hundreds of spins. *Nature*, 484(7395):489–492, April 2012. ISSN 0028-0836. doi: 10.1038/nature10981. URL <http://www.nature.com/nature/journal/v484/n7395/full/nature10981.html>.
- [239] Philip Richerme, Crystal Senko, Simcha Korenblit, Jacob Smith, Aaron Lee, Wesley C. Campbell, and Christopher Monroe. Trapped-ion quantum simulation of an Ising model with transverse and longitudinal fields. *arXiv:1303.6983*, March 2013. URL <http://arxiv.org/abs/1303.6983>.
- [240] Constantin Brif, Raj Chakrabarti, and Herschel Rabitz. Control of quantum phenomena: past, present and future. *New Journal of Physics*, 12(7):075008, July 2010. ISSN 1367-2630. doi: 10.1088/1367-2630/12/7/075008. URL <http://iopscience.iop.org/1367-2630/12/7/075008>.
- [241] Raj Chakrabarti, Rebing Wu, and Herschel Rabitz. Quantum multiobservable control. *Physical Review A*, 77(6):063425, June 2008. doi: 10.1103/PhysRevA.77.063425. URL <http://link.aps.org/doi/10.1103/PhysRevA.77.063425>.

- [242] Raj Chakrabarti, Rebing Wu, and Herschel Rabitz. Quantum Pareto optimal control. *Physical Review A*, 78(3):033414, September 2008. doi: 10.1103/PhysRevA.78.033414. URL <http://link.aps.org/doi/10.1103/PhysRevA.78.033414>.
- [243] Vincent Beltrani, Pritha Ghosh, and Herschel Rabitz. Exploring the capabilities of quantum optimal dynamic discrimination. *The Journal of Chemical Physics*, 130(16):164112, April 2009. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.3114679. URL <http://scitation.aip.org/content/aip/journal/jcp/130/16/10.1063/1.3114679>.
- [244] F. Gembicki and Y.Y. Haimes. Approach to performance and sensitivity multiobjective optimization: The goal attainment method. *IEEE Transactions on Automatic Control*, 20(6):769--771, December 1975. ISSN 0018-9286. doi: 10.1109/TAC.1975.1101105.
- [245] A. M. Bagirov, L. Jin, N. Karmitsa, A. Al Nuaimat, and N. Sultanova. Subgradient method for nonconvex nonsmooth optimization. *Journal of Optimization Theory and Applications*, 157(2):416--435, May 2013. ISSN 0022-3239, 1573-2878. doi: 10.1007/s10957-012-0167-6. URL <http://link.springer.com/article/10.1007/s10957-012-0167-6>.
- [246] Abdullah Konak, David W. Coit, and Alice E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992--1007, September 2006. ISSN 0951-8320. doi: 10.1016/j.ress.2005.11.018. URL <http://www.sciencedirect.com/science/article/pii/S0951832005002012>.
- [247] Peter A. N. Bosman and Dirk Thierens. The naive MIDEA: A baseline multi-objective EA. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, EMO'05, pages 428--442, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-24983-4, 978-3-540-24983-2. doi: 10.1007/978-3-540-31880-4_30. URL http://dx.doi.org/10.1007/978-3-540-31880-4_30.
- [248] P. A N Bosman. On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(1): 51--69, February 2012. ISSN 1089-778X. doi: 10.1109/TEVC.2010.2051445.
- [249] James C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *John Hopkins APL Technical Digest*, 19(4):482--492, 1998.

- [250] Yufu Ning, Wansheng Tang, and Hui Wang. Hybrid genetic-SPSA algorithm based on random fuzzy simulation for chance-constrained programming. In Lipo Wang and Yaochu Jin, editors, *Fuzzy Systems and Knowledge Discovery*, number 3613 in Lecture Notes in Computer Science, pages 332--335. Springer Berlin Heidelberg, January 2005. ISBN 978-3-540-28312-6, 978-3-540-31830-9. URL http://link.springer.com/chapter/10.1007/11539506_41.
- [251] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171--2175, July 2012.
- [252] Pearu Peterson. F2py: a tool for connecting Fortran and Python programs. *International Journal of Computational Science and Engineering*, 4(4):296 -- 305, 2009. doi: 10.1504/IJCSE.2009.029165. URL http://www.inderscience.com/search/index.php?action=record&rec_id=29165.
- [253] Nicolas Barbey. github:nbarbey/fht, 2010. URL <https://github.com/nbarbey/fht>.
- [254] Rodolfo A. Jalabert and Horacio M. Pastawski. Environment-Independent Decoherence Rate in Classically Chaotic Systems. *Physical Review Letters*, 86(12):2490--2493, March 2001. doi: 10.1103/PhysRevLett.86.2490. URL <http://link.aps.org/doi/10.1103/PhysRevLett.86.2490>.
- [255] Fritz Haake. *Quantum Signatures of Chaos*, volume 54 of *Springer Series in Synergetics*. Third edition, 2010. URL <http://www.springer.com/physics/complexity/book/978-3-642-05427-3>.
- [256] Cozmin Ududec, Nathan Wiebe, and Joseph Emerson. Information-theoretic equilibration: The appearance of irreversibility under complex quantum dynamics. *Physical Review Letters*, 111(8):080403, August 2013. doi: 10.1103/PhysRevLett.111.080403. URL <http://link.aps.org/doi/10.1103/PhysRevLett.111.080403>.
- [257] E. L. Hahn. Spin echoes. *Physical Review*, 80(4):580--594, November 1950. doi: 10.1103/PhysRev.80.580. URL <http://link.aps.org/doi/10.1103/PhysRev.80.580>.
- [258] Alexandros Beskos, Dan Crisan, and Ajay Jasra. On the stability of sequential Monte Carlo methods in high dimensions. arXiv e-print 1103.3965, March 2011. URL <http://arxiv.org/abs/1103.3965>.

- [259] Matthew B. Hastings and Tohru Koma. Spectral gap and exponential decay of correlations. *Communications in Mathematical Physics*, 265(3):781–804, April 2006. ISSN 0010-3616, 1432-0916. doi: 10.1007/s00220-006-0030-4. URL <http://arxiv.org/abs/math-ph/0507008>. arXiv:math-ph/0507008.
- [260] Bruno Nachtergael and Robert Sims. Lieb-Robinson bounds and the exponential clustering theorem. *Communications in Mathematical Physics*, 265(1):119–130, July 2006. ISSN 0010-3616, 1432-0916. doi: 10.1007/s00220-006-1556-1. URL <http://link.springer.com/article/10.1007/s00220-006-1556-1>.
- [261] Bruno Nachtergael and Robert Sims. Much ado about something: Why Lieb-Robinson bounds are useful. *arXiv:1102.0835 [math-ph]*, February 2011. URL <http://arxiv.org/abs/1102.0835>. IAMP News Bulletin, October 2010, pp 22–29.
- [262] M. B. Hastings. Locality in quantum systems. *arXiv:1008.5137 [math-ph, physics:quant-ph]*, August 2010. URL <http://arxiv.org/abs/1008.5137>. arXiv: 1008.5137.
- [263] Troy Borneman. personal communication.
- [264] Adam Frees, John King Gamble, Daniel R. Ward, Robin Blume-Kohout, M. A. Eriksson, Mark Friesen, and S. N. Coppersmith. Compressed optimization of device architectures. *arXiv:1409.3846 [cond-mat, physics:quant-ph]*, September 2014. URL <http://arxiv.org/abs/1409.3846>. arXiv: 1409.3846.
- [265] Nathan Wiebe, Dominic Berry, Peter Høyer, and Barry C Sanders. Higher order decompositions of ordered operator exponentials. *Journal of Physics A: Mathematical and Theoretical*, 43:065203, January 2010. ISSN 1751-8113, 1751-8121. doi: 10.1088/1751-8113/43/6/065203. URL <http://iopscience.iop.org/1751-8113/43/6/065203>.
- [266] Xavier Didelot, Richard G. Everitt, Adam M. Johansen, and Daniel J. Lawson. Likelihood-free estimation of model evidence. *Bayesian Analysis*, 6(1):49–76, February 2011. URL <http://ba.stat.cmu.edu/abstracts/Didelot.php>.
- [267] S A Sisson and Y. Fan. Likelihood-free markov chain Monte Carlo. *arXiv:1001.2058*, January 2010. URL <http://arxiv.org/abs/1001.2058>.
- [268] Catalin Catana, Theodore Kypraios, and Mădălin Guță. Maximum likelihood versus likelihood-free quantum system identification in the atom maser. *Journal of Physics A: Mathematical and Theoretical*, 47(41):415302, October 2014. ISSN

1751-8113, 1751-8121. doi: 10.1088/1751-8113/47/41/415302. URL <http://iopscience.iop.org/1751-8121/47/41/415302/refs>.

- [269] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL <http://ipython.org>.
- [270] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [271] Steven Casagrande and Christopher Granade. *InstrumentKit: Python package for interacting with laboratory equipment*. 2013. URL <https://github.com/Galvant/InstrumentKit>.

APPENDICES

A Likelihood-Free SMC

A.1 Weak and Strong Simulation

As a step towards integrating quantum simulation resources into SMC, remove the explicit dependence of SMC on [strong simulation](#) of likelihood functions, producing a *likelihood-free* method. Such methods have been used in a variety of classical contexts to mitigate simulation costs [266; 267]. Likelihood-free methods have also been used in quantum information to perform system identification based on quantum trajectories [268].

As illustrated in [Figure A.1](#), a strong simulator produces the exact probability $\Pr(d|x)$ of obtaining d from a hypothesis x , given a possible datum d and a particular hypothesis. By contrast, [weak simulation](#) produces samples according to a hypothetical distribution. Our approach to generalizing SMC to a likelihood-free method, then, is to reconstruct *approximate* likelihoods from repeated sampling of weak simulators. In particular, we will treat the likelihood that would be produced by a strong simulator as a parameter to be estimated in a secondary estimation problem; this reduces to estimating the bias of a coin or die, which has been previously been solved as a toy model for tomography [97]. As an additional advantage, this approach allows us to deal with weak simulators that produce samples from a distribution related to the true likelihood added to a noisy channel.

More formally, we treat learning the likelihood function evaluated for a specific hy-

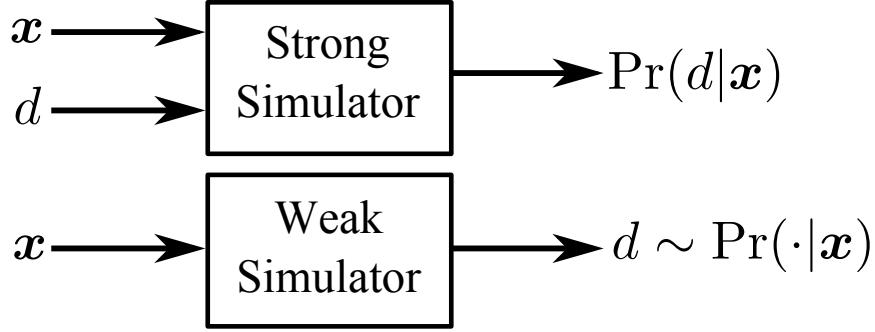


Figure A.1: Inputs and outputs for weak and strong simulations of the same likelihood function.

pothesis \underline{x}_i and observation d , $\ell_i := \Pr(d|\underline{x}_i)$, as a secondary estimation problem. In the limit that we estimate ℓ_i with very high accuracy, we approximately recover strong simulation, but to do so using strong measurement or ensemble measurement with small polarization can be prohibitively expensive. Instead, we are interested in using a smaller number of samples and assessing the impact on learning performance.

To learn each likelihood evaluation ℓ_i , we collect a dataset $D'_i \sim \text{Bin}(m, \ell_i)$ from our weak simulator consisting of m samples per hypothesis under evaluation. We can then let $\hat{\ell}_i = |\{d' \in D'_i : d = d'\}|$ be an estimate of the likelihood, based on the frequency with which the weak simulator agrees with the experimental observations. The approximate Bayes update used by SMC is then given by

$$w_i \mapsto w_i \times \hat{\ell}_i = w_i \times (\ell_i + \epsilon_i), \quad (\text{A.1})$$

for some error term ϵ_i introduced by the finite sampling of the weak simulator.

This presents a hazard, though, if D'_i never agrees with d , as this would lead to $\hat{\ell}_i = 0$ for that hypothesis, which in turn would completely eliminate a particle from the SMC approximation, reducing numerical stability as described in [Section 2.2.1.3](#). To avoid this, we use the result of [97], in which we add *hedging* to our data, giving the add- β rule for a hedged binomial estimator,

$$\hat{\ell}_i = \frac{\beta + k}{\beta n_o + m}, \quad (\text{A.2})$$

where n_o is the number of possible outcomes, $k = |\{d' \in D'_i : d = d'\}|$, $m = |D'_i|$, and where $\beta > 0$ is a hedging parameter whose optimal value depends on n_o and the expected noise in the weak simulator. Implicitly, this reduces the model to a two-outcome

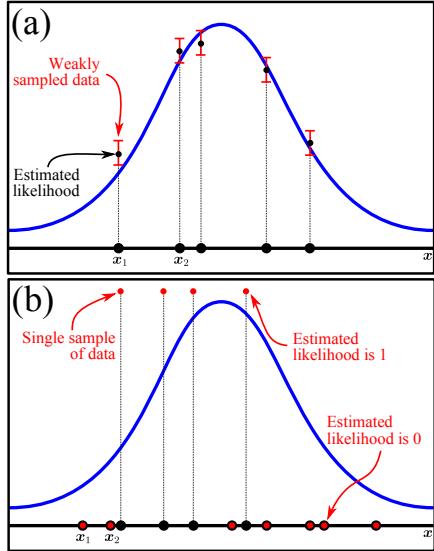


Figure A.2: Comparison of different extremes of likelihood-free sampling strategies, $m \rightarrow \infty$ (a) and $m = 1$ (b).

(Bernoulli) trial, where the weak simulator either agrees with the experiment or does not; we label the former case as a ``success'', and then reconstruct the likelihood by estimating the success probability from the success frequency. The hedging used here can be thought of as reducing slightly the amount that we update our hypotheses based on any one datum to represent our finite confidence about the results of estimating likelihood evaluations. Indeed, the hedging has a much more pronounced effect as we take fewer samples per hypothesis.

Since the variance of hedged binomial estimators is a well-known function of the true success probability, we need not choose m *a priori*, but can draw samples until the estimated variance falls below an acceptable level. This procedure is called adaptive likelihood estimation (ALE) [165], and can then be substituted into SMC to produce likelihood-free parameter estimation.

As an alternative, we can also explore the extreme case of $m = 1$, using a single weak simulation for each particle, and without hedging, thus completely removing support for a large number of particles with each evaluation. For very large numbers of particles, this can be effective, as we gain in a variety of hypothesis what we lose in the accuracy with which we evaluate each, in analogy to the one-bit limits explored in [Section 2.1.3](#)

and [Section 2.3.2](#). The dichotomy between these strategies is illustrated in [Figure A.2](#). In the next section, we shall explore the robustness of likelihood-free SMC in terms of the total number of weak simulations to address this tradeoff.

A.2 Robustness to Finite Sampling

In order to study the robustness of likelihood-free SMC without reference to the complexities of a particular model, we shall assume a very simple likelihood function as a stand-in for a more useful one. Thus, we study the effect of finite sampling itself. The particular model that we consider is that of a photodetector with known bright and dark references α and β , respectively, monitoring a source that emits a photon with probability p . Thus, the likelihood is given by

$$\Pr(\text{click}|p) = p(1 - \beta) + (1 - p)(1 - \alpha). \quad (\text{A.3})$$

In practice, $p = p(\underline{x})$ for some more interesting model parameters \underline{x} , as in the examples given by [Section 2.3.1.3](#) and [Section 2.3.3](#).

The Fisher information for this model is simple to find, and yields the Cramér-Rao bound [[165](#)]

$$L(\hat{p}, p) \geq \frac{1}{6(\alpha - \beta)^2 N}, \quad (\text{A.4})$$

where N is the number of measurements made. Since this is independent of the true emission probability p , we need not consider the BCRB in this case.

In [Figure A.3](#), we show that for a fixed number of particles, likelihood-free SMC approaches this bound as $m \rightarrow \infty$, and that there is a regime for finite m that well-approximates this bound. Finally, in [Figure A.4](#), we show that for adaptive likelihood estimation (ALE), we also reach the optimal bound as $\epsilon \rightarrow 0$.

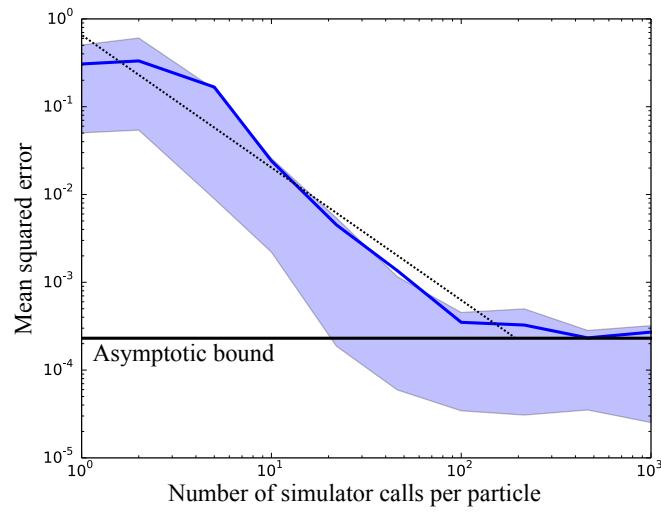


Figure A.3: Risk incurred by likelihood-free SMC for the noisy-coin photodector model versus the number of weak simulations m per particle.

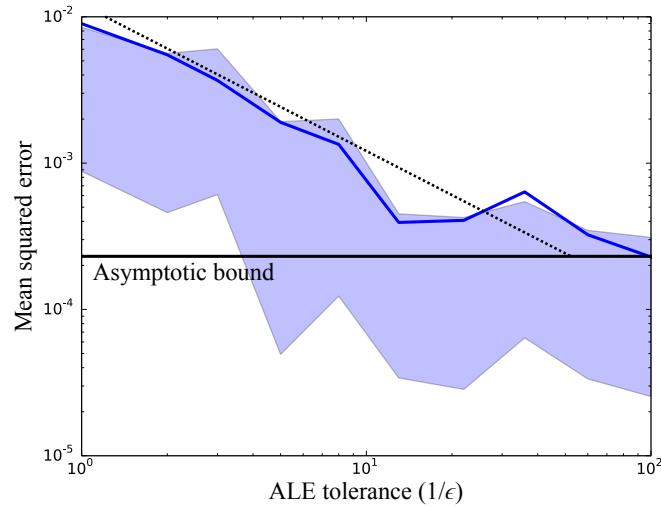


Figure A.4: Risk incurred by likelihood-free SMC for the noisy-coin photodector model versus adaptive likelihood estimation tolerance ϵ .

B Bounds for Quantum Hamiltonian Learning

In this Appendix, we consider several different bounds on the performance of quantum Hamiltonian learning and quantum bootstrapping, as introduced in [Chapter 6](#).

B.1 Sampling Error

We first consider the robustness of sequential Monte Carlo with either non-interactive or interactive quantum likelihood evaluation to errors introduced by finite sampling of the trusted device. In particular, we will consider that the estimated likelihood $\hat{\ell}_i$ is related to the true likelihood function $\ell_i := \Pr(d|H(\underline{x}_i))$ for a single datum d by an additive error,

$$\hat{\ell}_i = \ell_i + \eta_i. \tag{B.1}$$

We shall also assume that the prior distribution carries a finite error η'_i , such that $w_i = \Pr(\underline{x}_i) + \eta'_i$. Finally, we shall assume that the effective error

$$\eta := \sum_i [\Pr(d|\underline{x}_i)|\eta'_i| + \Pr(\underline{x}_i)|\eta_i| + |\eta_i\eta'_i|] \tag{B.2}$$

is small compared to the true SMC estimate of the total likelihood, such that

$$\eta \leq \frac{1}{2} \Pr(d) = \frac{1}{2} \sum_i \Pr(d|\underline{x}_i) \Pr(\underline{x}_i). \quad (\text{B.3})$$

Using the triangle inequality together with Taylor expansion, we then find [41] that the error

$$\epsilon_i := \left| \frac{\Pr(d|\underline{x}_i) \Pr(\underline{x}_i)}{\sum_j \Pr(d|\underline{x}_j) \Pr(\underline{x}_j)} - \frac{(\Pr(d|\underline{x}_i) + \eta_i)(\Pr(\underline{x}_i) + \eta'_i)}{\sum_j (\Pr(d|\underline{x}_j) + \eta_j)(\Pr(\underline{x}_j) + \eta'_j)} \right| \quad (\text{B.4})$$

in the posterior particle weights is bounded above by

$$\epsilon \leq \frac{3 \left(\sqrt{\sum_k \Pr^2(d|\underline{x}_k) \|\underline{\eta}'\|_2} + \sqrt{\sum_k \Pr^2(\underline{x}_k) \|\underline{\eta}\|} \right)}{\sum_k \Pr(d|\underline{x}_k) \Pr(\underline{x}_k)} + O(\eta^2), \quad (\text{B.5})$$

where $\underline{\eta}$ is a vector of the likelihood errors, $\underline{\eta}'$ is a vector of the prior errors, and where $\epsilon := \sum_i |\epsilon_i|$ is the 1-norm of the posterior weight errors. Moreover, the errors introduced are small enough to provide asymptotic stability against surprising data if we ensure that [41]

$$\|\underline{\eta}\|_2 \ll \frac{\sum_k \Pr(d|\underline{x}_k) \Pr(\underline{x}_k)}{\sqrt{\sum_k \Pr^2(\underline{x}_k)}} = \sqrt{n_{\text{ess}}} \left(\sum_k \Pr(d|\underline{x}_k) \Pr(\underline{x}_k) \right). \quad (\text{B.6})$$

This condition is very useful, as it gives that resampling is a resource for stabilizing our distributions against outlying estimates of likelihoods. Thus, our results complement those showing asymptotic stability of sequential Monte Carlo in high dimensions [258] to demonstrate a robust solution to Hamiltonian learning.

B.2 Compressed Hamiltonian Learning with Non-commuting Hamiltonians

In practice, when considering compressed quantum Hamiltonian learning, we are interested in the more general case where we no longer assume that terms in the Hamiltonian under study mutually commute. When we relax this assumption, the factorization used to prove (6.16) no longer holds, such that we must use a more sophisticated approach. Moreover, because the dynamics arising at the cut between the simulated and truncated

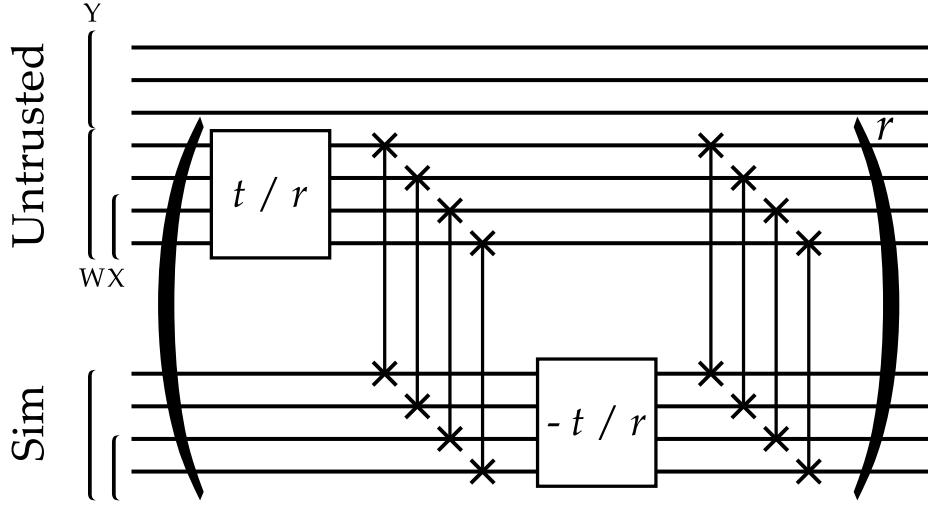


Figure B.1: Schematic of repeated SWAPing between untrusted device and trusted simulator, showing regions of support W , X and Y for the simulator, observable and complement of the simulator, respectively. Boxes indicate Hamiltonian evolution under H or H_- for the untrusted and trusted devices.

registers no longer commute with the dynamics internal to trusted simulator, there can be a significant causal influence on evolution of the observable.

We address these concerns by using multiple inversion steps, generalizing interactive quantum likelihood evaluation. As shown in Figure B.1, we use r rounds of SWAP gates between the trusted and untrusted system, such that for large r , the total evolution is described by the Trotter formula as

$$\left(e^{iH_- t/r} e^{-iH t/r} \right)^r \approx e^{-i(H - H_-)t} = e^{-i(H_{\text{out}} + H_{\text{in}} + \Lambda)t}, \quad (\text{B.7})$$

where $\Lambda := H_{\text{in}} - H_-$.

Introducing additional rounds of interactivity in this way effectively reduces the degree to which the noncommutativity of H and H_- affect the dynamics of the observable. In particular, the interactions at the cut are less able to affect $A(t)$.

This intuition is made precise by recursively bounding the dynamics for each iteration.

tion of the IQLE protocol. We first define two observables,

$$A^{(n)} := e^{-iHt/r} e^{-iH_{-t}/r} A^{(n-1)} e^{iH_{-t}/r} e^{-iHt/r} \quad (\text{B.8a})$$

$$\tilde{A}^{(n)} := e^{i\Lambda t/r} \tilde{A}^{(n-1)} e^{-i\Lambda t/r}, \quad (\text{B.8b})$$

respectively representing evolution of the initial observable $A^{(0)} = \tilde{A}^{(0)} = A(0)$ under r rounds of the actual IQLE protocol and the simulation of this protocol in the trusted system alone. Then, the norm between these observables after all r rounds is bounded by [10]

$$\begin{aligned} \|A^{(r)} - \tilde{A}^{(r)}\| / \|A\| &\leq (\|[H_{\text{in}}, \Lambda]\| + \|[H_{\text{int}}, H_{\text{in}}]\|) \frac{t^2}{r} + 2\|H_{\text{int}} \cap A\|t + \\ &2\|H_{\text{int} \setminus A}\| |\sup A| t e^{\mu \text{dist}(A, H_{\text{out}})} [e^{2s|t|} - 1] e^{2\|H_{\text{out}} + H_{\text{int} \setminus A}\|t/r}, \end{aligned} \quad (\text{B.9})$$

where $\sup A$ is the number of sites on which the observable A is supported, $\text{dist}(A, H_{\text{out}})$ is the width of "buffer" between the observable and the truncated terms, and where s and μ are constants that depend only on Λ [259].

This leaves that, in the limit of many inversion steps and good characterization of the internal dynamics, that the linear term $\|H_{\text{int}} \cap A\|t$ is the limiting term in the achievable accuracy. For exponentially decaying interactions, we can suppress this term exponentially, however, by widening the buffer between the observable and untrusted devices such that this imposes at most a modest cost.

C Source Code

Listing C.1: Source code for [Figure 2.5](#).

```
## IMPORTS ##

import numpy as np
import matplotlib
5 try:
    import mpltools.style
    mpltools.style.use('ggplot')
except ImportError:
    pass
10 # We want to make the plot text large enough to
# read in a printed paper.
matplotlib.rcParams['axes.titlesize'] = 'x-large'
matplotlib.rcParams['axes.labelsize'] = 'large'
15 import matplotlib.pyplot as plt

# Import things from QInfer.
from qinfer.distributions import UniformDistribution
from qinfer.test_models import SimplePrecessionModel
20 from qinfer.resamplers import LiuWestResampler
from qinfer.smc import SMCUpdater

## SETUP ##
# Set a prior  $\omega \sim \text{Uni}(-1,1)$ .
25 prior = UniformDistribution([-1, 1])
```

```

# Set the likelihood to be that of (2.11).
model = SimplePrecessionModel()
# Choose a particular true value for  $\omega$  extreme enough to illustrate the problem.
true_model = np.array([[0.65]])

# Make a new figure large enough to hold both subfigures.
plt.figure(figsize=(12,6))

# Update and plot for  $a^2 + h^2 = 1$ .
plt.subplot(1, 2, 1)
updater = SMCUpdater(
    model, 1000, prior,
    resampler=LiuWestResampler(0.98, postselect=False)
)
for idx_datum in xrange(35):
    t_k = (9.0/8)**(idx_datum+1)
    expparams = np.array([t_k], dtype=model.expparams_dtype)
    datum = model.simulate_experiment(true_model, expparams)
    updater.update(datum, expparams)
updater.plot_posterior_marginal()
plt.xlabel(r' $\omega'$ )
plt.ylabel(r' $\Pr(\omega|D)$ ')
plt.title(r' $a=0.98, h=\sqrt{1-0.98^2}$ ')
plt.xlim(-1, 1)

# Update and plot for  $a^2 + h^2 > 1$ .
plt.subplot(1, 2, 2)
updater = SMCUpdater(model, 1000, prior,
    resampler=LiuWestResampler(1, h=0.005, postselect=False)
)
for idx_datum in xrange(35):
    t_k = (9.0/8)**(idx_datum+1)
    expparams = np.array([t_k], dtype=model.expparams_dtype)
    datum = model.simulate_experiment(true_model, expparams)
    updater.update(datum, expparams)
updater.plot_posterior_marginal()
plt.xlabel(r' $\omega'$ )
plt.xlim(-1, 1)
plt.title(r' $a=1, h=0.005$ ')

plt.show()

```

Listing C.2: QInfer implementation of simple precession model of (2.11), with full metadata and Fisher score calculation.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
##
# test_models.py: Simple models for testing inference engines.
##
# © 2012 Chris Ferrie (csferrie@gmail.com) and
# Christopher E. Granade (cgranade@gmail.com)
#
# This file is a part of the Qinfer project.
# Licensed under the AGPL version 3.
##
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU Affero General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU Affero General Public License for more details.
#
# You should have received a copy of the GNU Affero General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
## FEATURES #####
from __future__ import division # Ensures that a/b is always a float.
##
## EXPORTS #####
__all__ = [
    'SimpleInversionModel',
    'SimplePrecessionModel',
    'NoisyCoinModel',
    'NDieModel'
]
##
## IMPORTS #####
import numpy as np
from utils import binomial_pdf
from abstract_model import Model, DifferentiableModel
```

```

## CLASSES #####
49 class SimpleInversionModel(DifferentiableModel):
    """
    Describes the free evolution of a single qubit prepared in the
     $|+\rangle$  state under a Hamiltonian  $H = \omega\sigma_z/2$ ,
    using the interactive QLE model proposed by [WGFC13a]_.

54     :param float min_freq: Minimum value for  $\omega$  to accept as valid.
        This is used for testing techniques that mitigate the effects of
        degenerate models; there is no "good" reason to ever set this other
        than zero, other than to test with an explicitly broken model.
    """
59

    ## INITIALIZER ##

64     def __init__(self, min_freq=0):
        super(SimpleInversionModel, self).__init__()
        self._min_freq = min_freq

    ## PROPERTIES ##

69     @property
70     def n_modelparams(self):
        return 1

74     @property
75     def modelparam_names(self):
        return [r'\omega']

79     @property
80     def expparams_dtype(self):
        return [('t', 'float'), ('w_', 'float')]

84     @property
85     def is_n_outcomes_constant(self):
        """
        Returns ``True`` if and only if the number of outcomes for each
        experiment is independent of the experiment being performed.

        This property is assumed by inference engines to be constant for
        the lifetime of a Model instance.
    """
89     return True

```

```

## METHODS ##

94 def are_models_valid(self, modelparams):
    return np.all(modelparams > self._min_freq, axis=1)

104 def n_outcomes(self, expparams):
    """
    Returns an array of dtype ``uint`` describing the number of outcomes
    for each experiment specified by ``expparams``.

    :param numpy.ndarray expparams: Array of experimental parameters. This
        array must be of dtype agreeing with the ``expparams_dtype``
        property.
    """
    return 2

109 def likelihood(self, outcomes, modelparams, expparams):
    # By calling the superclass implementation, we can consolidate
    # call counting there.
    super(SimpleInversionModel, self).likelihood(
        outcomes, modelparams, expparams
    )

114 # Possibly add a second axis to modelparams.
if len(modelparams.shape) == 1:
    modelparams = modelparams[..., np.newaxis]

```

Listing C.3: Source code for Figure 2.8.

```

from __future__ import division

3 import numpy as np
import qinfer as qi

class DriftingCosineModel(qi.Model):

8     step_dist = qi.NormalDistribution(0, 1)

    @property
    def n_modelparams(self):
        return 2
13    @property
    def is_n_outcomes_constant(self):
        return True
    def n_outcomes(self, expparams):

```

```

    return 2
18  def are_models_valid(self, modelparams):
        return np.all(np.logical_and(
            modelparams > 0, modelparams <= 1
        ), axis=1)
@property
23  def expparams_dtype(self):
        return [('t', 'float')]

def update_timestep(self, modelparams, expparams):
    # Note that the timestep update is presumed to be independent of the
    # experiment.

    steps = (
        self.step_dist.sample(n=modelparams.shape[0] * expparams.shape[0])
        * np.sqrt(modelparams[:, 1, np.newaxis] * expparams['t'])
    )[:, :]
28

    new_modelparams = modelparams.copy()[:, :, np.newaxis]
    new_modelparams = np.repeat(new_modelparams, expparams.shape[0], -1)

    new_modelparams[:, 0, :] += steps
    np.clip(new_modelparams, 0, 1, out=new_modelparams)

    return new_modelparams

38  def likelihood(self, outcomes, modelparams, expparams):
        pr0 = np.zeros((modelparams.shape[0], expparams.shape[0]))

        pr0[:, :] = np.cos(modelparams[:, 0, np.newaxis] * expparams['t'])**2

43  return qi.Model.pr0_to_likelihood_array(outcomes, pr0)

```

Listing C.4: QInfer implementation of Fisher score for the randomized benchmarking model (2.46).

```

lastlinelastline
2      n_e = expparams.shape[0]
n_o = outcomes.shape[0]
n_p = self.n_modelparams

m = expparams['m'].reshape((1, 1, 1, n_e))
7
L = self.likelihood(outcomes, modelparams, expparams)[na, ...]
outcomes = outcomes.reshape((1, n_o, 1, 1))

```

```

11     if not self._il:
12
13         p, A, B = modelparams.T[:, :, np.newaxis]
14         p = p.reshape((1, 1, n_m, 1))
15         A = A.reshape((1, 1, n_m, 1))
16         B = B.reshape((1, 1, n_m, 1))
17
18         q = (-1)**(1-outcomes) * np.concatenate(np.broadcast_arrays(
19             A * m * (p ** (m-1)), p**m, np.ones_like(p),
20             ), axis=0) / L
21
22     else:
23
24         p_tilde, p_ref, A, B = modelparams.T[:, :, np.newaxis]
25         p_C = p_tilde * p_ref
26
27         mode = expparams['reference'][np.newaxis, :]
28
29         p = np.where(mode, p_ref, p_C)
30
31         p = p.reshape((1, 1, n_m, n_e))
32         A = A.reshape((1, 1, n_m, 1))
33         B = B.reshape((1, 1, n_m, 1))
34
35         q = (-1)**(1-outcomes) * np.concatenate(np.broadcast_arrays(
36             np.where(mode, 0, A * m * (p_tilde ** (m - 1)) * (p_ref ** m)),
37             np.where(mode,
38                 A * m * (p_ref ** (m - 1)),
39                 A * m * (p_ref ** (m - 1)) * (p_tilde ** m)
40             ),
41             p**m, np.ones_like(p)
42             ), axis=0) / L
43
44     if return_L:
45         # Need to strip off the extra axis we added for broadcasting to q.
46         return q, L[0, ...]
47     else:
48         return q

```

Listing C.5: Example of a Rabi model with bright/dark referencing.

```

# -*- coding: utf-8 -*-
2 ## 
# rabi_demo.py: Demonstrates the use of QInfer for Rabi models.
## 

```

```

#
## FEATURES #####
7
## IMPORTS #####
from __future__ import division

12 ## CONSTANTS #####
import numpy as np
import scipy.linalg as la
import scipy.io as sio

17 import qinfer as qi

import matplotlib.pyplot as plt
from matplotlib import ticker, rcParams
22 import mpltools as mpl
import mpltools.style
mpltools.style.use('ggplot')

rcParams['axes.labelsize'] = 'xx-large'
27 rcParams['xtick.labelsize'] = 'large'
rcParams['ytick.labelsize'] = 'large'
rcParams['figure.autolayout'] = True

## FUNCTIONS #####
32 two_pi = np.pi * 2

## CLASSES #####
37 def mhz_tick_formatter(x, p):
    return "{} MHz".format(x / 1e6)

def us_tick_formatter(x, p):
    return r"{} \mu s".format(x * 1e6)
42

## CLASSES #####
47 class RabiModel(qi.Model):
    """
        Model of a single shot in a Rabi flopping experiment.

    Model parameters:

```

```

0. :math:`\omega_R` , on-resonance Rabi frequency [Hz].
1. :math:`\Delta\omega` , detuning magnitude [Hz].
2. :math:`1 / T_2` , decoherence strength [Hz].
3. :math:`\alpha` , probability of at least one photon in bright ref.
4. :math:`\beta` , probability of at least one photon in dark ref.

52 Experiment parameters:
mode: Specifies whether a reference or signal count is being performed.
t: Pulse width :math:`\tau`.
"""

62 REF_DARK = 0
REF_BRIGHT = 1
SIGNAL = 2

67 @property
def n_modelparams(self):
    return len(self.modelparam_names)

72 @property
def modelparam_names(self):
    return [
        r'\omega_R',
        r'\Delta \omega',
        r'T_2^{-1}',
        r'\alpha', r'\beta'
    ]

77 @property
def expparams_dtype(self):
    return [('t', 'float'), ('mode', 'int')]

82 @property
def is_n_outcomes_constant(self):
    return True

87 @staticmethod
def are_models_valid(modelparams):
    # modelparams has indices [idx_model, idx_model_parameter],
    # such that :math:`\vec{x}_i` corresponds to `modelparams[i, :]`.
    #
    # Our constraints are on each different model parameter
    # (second index), and so we build up a list of constraints,
    # then use np.all to demand all are met at once.
    #

```

```

# For example, modelparams[:, 0] >= 0 demands that the
# :math:`\omega_R` parameter is non-negative for all models.
97   return np.all(
      [
        # Require that all frequencies be positive.
        modelparams[:, 0] >= 0,
        modelparams[:, 1] >= 0,
        modelparams[:, 2] >= 0,
102
        # Require that alpha and beta are probabilities.
        modelparams[:, 3] >= 0,
        modelparams[:, 3] <= 1,
107        modelparams[:, 4] >= 0,
        modelparams[:, 4] <= 1,
      ],
      axis=0 # Axis 0 is the outermost, corresponding to that
             # we have made a list of constraints.
    )
112

def canonicalize(self, modelparams):
    idx_swap = np.nonzero(
        modelparams[:, 4] <= modelparams[:, 3]
    )
117    temp = modelparams[idx_swap, 3].copy()
    modelparams[idx_swap, 3] = modelparams[idx_swap, 4]
    modelparams[idx_swap, 4] = temp

122    return modelparams

def n_outcomes(self, expparams):
    return 2

127 def likelihood(self, outcomes, modelparams, expparams):
    """
    Returns the likelihood of a *success*, that is, a bin with at least
    one photon. A success is described by the outcome label `1`, while
    a bin without photons is given the label `0`.
    """
132

    # Here, it's more convenient to express the probability of 1 than of 0.
    pr1 = np.zeros((modelparams.shape[0], expparams.shape[0]))

    # Give names to each parameter.
    omega_R = modelparams[:, 0] * two_pi
    d_omega = modelparams[:, 1] * two_pi

```

```

T2inv    = modelparams[:, 2]
alpha     = modelparams[:, 3]
beta      = modelparams[:, 4]

142

for idx_experiment in xrange(expparams.shape[0]):
    t = expparams[idx_experiment]['t']
    mode = expparams[idx_experiment]['mode']

147
    # If we're doing a reference, just return the reference probability.
    if mode == self.REF_DARK:
        pr1[:, idx_experiment] = beta
    elif mode == self.REF_BRIGHT:
        pr1[:, idx_experiment] = alpha
    else:
        # For the actual signal, we need to do some more calculations.
        # Start off by computing the visibility due to T_2.
        eta = np.exp(- t * T2inv)
        # The effective field term sqrt{\Delta \omega^2 + \omega_R^2}
        # appears a lot, so precalculate it.
        w_eff2 = (d_omega**2 + omega_R**2)
        # Now find the signal before photon statistics.
        sig = (
            (1 - eta) * (1 / 2) +
            eta * (
                (2 * (d_omega ** 2) + omega_R**2 * (1 + np.cos(t * np.sqrt(w_eff2))
                (2 * w_eff2)
            )
        )
        # Finally, compute the probability from the signal by
        # incorporating photon statistics.
        pr1[:, idx_experiment] = sig * (alpha - beta) + beta

152
157
162
167
172
177
182

    return qi.Model.pr0_to_likelihood_array(outcomes, 1 - pr1)

if __name__ == "__main__":
    # First, we define our priors.
    # The prior on alpha and beta doesn't matter,
    # as we'll set it to precisely the measured
    # value each time.
    ab_prior = qi.UniformDistribution([
        [0, 0.01],
        [0, 0.01]
    ])
    prior = qi.ProductDistribution(

```

```

    qi.UniformDistribution([
        [0, 3e7],
        [0, 5e6],
        [0, 5e6],
    ]),
    ab_prior
)
192
n_shots = 20000

m = qi.BinomialModel(RabiModel())
u = qi.smc.SMCUpdater(m, int(5e4), prior)
197
data = sio.loadmat('../data/rabi-data.mat')

# Record the estimates and std deviations as we go.
estimate_hist = []
err_hist = []

202
for idx_exp in xrange(data['times'].shape[1]):
    # Extract the bright and dark references, as well as the signal.
    rb, rd, s = n_shots * data['data'][:, idx_exp]

207
    if idx_exp % 10 == 0:
        print idx_exp

        # Pack everything into an expparams array.
        exp = np.array([
            (data['times'][0, idx_exp], RabiModel.REF_BRIGHT, n_shots)
        ], dtype=m.expparams_dtype)

        # Remove everything we know about alpha and beta
        u.particle_locations[:, 3:5] = data['data'][:2, idx_exp]

212
        # Update with the actual data.
        exp['mode'] = RabiModel.SIGNAL
        u.update(int(s), exp)

217
        # Report the mean so far.
        mu = u.est_mean()
        estimate_hist.append(mu)
        err_hist.append(
            np.diag(la.sqrtm(u.est_covariance_mtx())))
    )

```

```

w_eff = np.sqrt(np.sum(mu[0:2]**2))
vis = mu[3] / 3
print u.est_mean(), "{:.2} MHz".format(w_eff/1e6), "{:.2}".format(vis)

232

estimate_hist = np.array(estimate_hist)
err_hist = np.array(err_hist)

237
for idx_modelparam in xrange(3):
    fig = plt.figure()
    mpl.special.errorfill(
        n_shots * np.arange(data['times'].shape[1]),
        estimate_hist[:, idx_modelparam],
        yerr=err_hist[:, idx_modelparam]
    )
    plt.xlabel("Number of shots")
    plt.ylabel("{}".format(m.modelparam_names[idx_modelparam]))
    plt.gca().get_yaxis().set_major_formatter(
        ticker.FuncFormatter(mhz_tick_formatter)
    )
    plt.xlim(0, plt.xlim()[1])
    plt.savefig('../latex/figures/rabi-demo-hist-{}.pdf'.format(
        idx_modelparam
    ))
242

247

252

plt.figure()
u.plot_covariance(corr=False, param_slice=np.s_[:3])
plt.title(r'Cov( $\hat{\omega}_R, \Delta\hat{\omega}, T_2^{-1}$ )', y=1.08)
plt.savefig('../latex/figures/rabi-demo-cov.pdf')

257

plt.figure()
rb, rd, sig = data['data']
if not u.just_resampled:
    u.resample()
final_models = u.particle_locations.copy()
# Set perfect referencing.
final_models[:, 3:5] = [1, 0]
ts = data['times'][0, :]
experiments = np.empty(
    ts.shape,
    dtype=m.underlying_model.expparams_dtype
)
262
267
experiments['mode'] = RabiModel.SIGNAL
experiments['t'] = data['times'][0, :]
L = m.underlying_model.likelihood(np.array([1]),
    final_models,
272

```

```

    experiments
)[0, :, :]
277 mean_L = np.mean(L, axis=0)
lower_ci_L = mean_L - np.percentile(L, 0.025, axis=0)
upper_ci_L = mean_L - np.percentile(L, 0.975, axis=0)
plt.plot(ts, (sig - rd) / (rb - rd), '--', label='Referenced Data')
mpl.special.errorbar(
282     ts,
     mean_L,
     yerr=[lower_ci_L, upper_ci_L],
     ls=':', lw=3,
     label='Mean Simulated Signal',
287     label_fill='95% Credible Region'
)
plt.xlabel('Time')
plt.ylabel('Referenced Counts')
plt.gca().get_xaxis().set_major_formatter(
292     ticker.FuncFormatter(us_tick_formatter)
)
plt.legend()
plt.savefig('../latex/figures/rabi-demo-referenced-data.pdf')

297 plt.show()

```

Listing C.6: Implementation of Floquet-Leskes expansion of the stroboscopic Hamiltonian for a nitrogen vacancy center coupled to a carbon nucleus via a hyperfine interaction.

```

# Zero field splitting and gyromagnetic ratios (all in MHz)
ZFS = 2.87e3
3 ge = 2.8025
gc = 1070.5e-6

Si = np.eye(3)
Sx = np.array([[0,1,0],[1,0,1],[0,1,0]])/np.sqrt(2)
8 Sy = np.array([[0,-1j,0],[1j,0,-1j],[0,1j,0]])/np.sqrt(2)
Sz = np.array([[1,0,0],[0,0,0],[0,0,-1]])

Ii = np.eye(2)
Ix = np.array([[0,1],[1,0]])/2.
13 Iy = np.array([[0,-1j],[1j,0]])/2.
Iz = np.array([[1,0],[0,-1]])/2.

# Also make some common products of matrices, to avoid having to recalculate
# every time.

```

```

18 Sz2Ii = np.kron(np.dot(Sz, Sz), Ii)
SxIx = np.kron(Sx, Ix)
SyIy = np.kron(Sy, Iy)
SzIz = np.kron(Sz, Iz)
SIcross = (np.kron(Sz, Ix) + np.kron(Sx, Iz))

23 def lab_ham(Bx,By,Bz,zfs,Axx,Ayy,Azz,Azx):
"""
    Returns the lab frame hamiltonian of spin-1 electron + carbon 13
"""

28 # ZFS + Zeeman + Hyperfine
return two_pi*(
    zfs * Sz2Ii
    + ge * np.kron(Bx*Sx + By*Sy + Bz*Sz, Ii)
    + gc * np.kron(Si, Bx*Ix + By*Iy + Bz*Iz)
    + (Axx * SxIx + Ayy * SyIy + Azz * SzIz + Azx * SIcross)
)

33 def eff_ham(Bx,By,Bz,zfs,Axx,Ayy,Azz,Azx,woff):
"""
    Returns the effective 2nd order average hamiltonian rotating at D-woff. All
    inputs should be MHz, output is radians/micro-second

    :units Bx: Gauss
"""

43 labH = lab_ham(Bx,By,Bz,zfs,Axx,Ayy,Azz,Azx)
w = 2*np.pi*(zfs-woff)
z = np.zeros((2,2))

        wIi = w * Ii

48 A = labH[0:2,0:2] - wIi
B = labH[2:4,2:4]
C = labH[4:6,4:6] - wIi
D = labH[0:2,2:4]
E = labH[0:2,4:6]
F = labH[2:4,4:6]

53 # Construct the fourier coefficients as block matrices
H0 = np.bmat([[A,z,E],[z,B,z],[E.conj().transpose(),z,C]])
Hm = np.bmat([[z,z,z],[D.conj().transpose(),z,F],[z,z,z]])
Hp = Hm.conj().transpose()

58 # Compute a few of the commutators that appear more often.
H0p = com(H0, Hp)

```

```

63     H0m = -H0p.conj().transpose() #  $[A, B]^\dagger = [B^\dagger, A^\dagger]$ 
       Hmp = com(Hm, Hp)

       # Use the Leskes formulas for average Hamiltonians
       H1 = (H0p - H0m - Hmp) / w
68     H2 = (
           (com(Hm, H0p) + com(Hp, H0m)) / 2.
           - (com(H0, H0p) + com(H0, H0m))
           + (com(Hp, -H0p) - com(Hm, H0m) + com(Hp, H0m) + com(Hm, H0p)) / 2.
           + (com(Hp, Hmp) - com(Hm, Hmp))
         ) / (w**2)

73     return H0 + H1 + H2

def eff_super(Bx,By,Bz,zfs,Axx,Ayy,Azz,Azx,woff,T2c,T2e):
    """
    78     Returns the effective 2nd order average supergenerator rotating at ZFS-woff
    All inputs should be in MHz or microseconds.
    """
    H = eff_ham(Bx,By,Bz,zfs,Axx,Ayy,Azz,Azx,woff)
83     I = np.eye(6)

     # Hamiltonian super generator
     SH = 1j * (np.kron(H.T, I) - np.kron(I, H))

     # Lindblad supergenerator
     Le = np.sqrt(two_pi/T2e) * np.kron(Sz, Ii)
     LLe = np.dot(Le, Le) # note that Le is real-symmetric
     Lc = np.sqrt(two_pi/T2c) * np.kron(Si, Iz)
     LLC = np.dot(Lc, Lc) # note that Lc is real-symmetric
93     SL = np.kron(Le.conj(), Le) - (np.kron(LLe.T, I) + np.kron(I, LLe)) / 2. \
           + np.kron(Lc.conj(), Lc) - (np.kron(LLC.T, I) + np.kron(I, LLC)) / 2.

     return SH + SL

```

Listing C.7: Generation of Walsh-Hadamard basis in sequency order.

```

from __future__ import division

import numpy as np
4 import scipy.linalg as la

import matplotlib.pyplot as plt
try:
    import mpltools.style
9     mpltools.style.use('ggplot')

```

```

except:
    pass

# Start by definining the *sequency* of a row
14 # as the number of zero crossings. For a ±1 matrix
# such as the Hadamard matrices, this takes on a very nice
# form.
def sequency(H):
    return np.sum((1 - (H[:, :-1] * H[:, 1:]))) / 2, axis=1)

19 # Next, we define the Walsh basis by taking the sequency order
# of a Hadamard matrix.
def walsh_basis(n):
    dim = 2 ** n
24     H = la.hadamard(dim)
    return H[:, np.argsort(sequency(H))]

# Finally, we plot a nice example.
if __name__ == "__main__":
29     n = 3
        dim = 2 ** n
        wb = walsh_basis(n)[:, :]

        fig, subplots = plt.subplots(dim, 1, sharex=True)

34     for idx_dim, subplot in enumerate(subplots):
            plt.sca(subplot)
            plt.plot(np.concatenate([[0], wb[idx_dim, :]]), drawstyle='steps')
            plt.ylabel('{}'.format(idx_dim))
            plt.ylim(-1.2, 1.2)
            subplot.get_yaxis().set_ticks([])

        plt.subplots_adjust(bottom=0.15)
        fig.text(0.5, 0.075, 'Time', ha='center', va='center')
39        fig.text(0.06, 0.5, 'Sequency', ha='center', va='center', rotation='vertical')
        plt.show()

```

D QInfer: Implementation of SMC

QInfer [44] is an open-source library for prototyping and implementing sequential Monte Carlo with minimal effort, developed in collaboration with Chris Ferrie, and with kind contributions and testing from Ian Hincks, Nathan Wiebe, Yuval Sanders and Rahul Deshpande. Because of its generality, QInfer enables the rapid application of Bayesian methods such as those described in [Chapter 2](#) to new experimental contexts.

QInfer is provided open-source, and was used to generate most of the results described in this work, aiding in reproducibility as well as documenting the concrete implementation details needed for efficient implementation of SMC. Extensive documentation is available online at <http://python-qinfer.readthedocs.org>.

D.1 Design Considerations

QInfer was designed so as to provide an efficient, generic and easy-to-use tool that can be used either to implement sequential Monte Carlo directly, or to prototype implementations for eventual use in an FPGA or other more ``bare-metal'' application. Therefore, we required that QInfer be developed in a modular enough fashion that it is practical to experiment with resampling algorithms and parameters, experiment design heuristics and SMC quality parameters. In light of these design concerns, we chose to implement QInfer in Python, as this language offers a wide range of programming paradigms with

which code can be efficiently reused and modularized, while not requiring users to submit to arduous licensing requirements that can come along with other scientific computing platforms. An additional advantage of basing QInfer on Python is that it is therefore easy to build off of the rich community of other Python libraries, such as the IPython stack [269], QuaEC and QuTiP for quantum information [45; 1], DEAP and SciKit-Learn for evolutionary computing and machine learning [251; 270], and InstrumentKit for instrument control [271].

Importantly, in order to be effective as an implementation of SMC, QInfer must be performant, so that as little overhead is introduced as possible, leaving the main cost as the calls to a user-provided likelihood function. To accomplish this in Python, we make heavy use of *vectorization*, such that nearly every numerical object considered by QInfer is a tensor of real numbers. For instance, a class implementing a likelihood function $\Pr(d|\underline{x}; \underline{e})$ is expected to return a rank-three tensor

$$L_{ijk} := \Pr(d_i|\underline{x}_j; \underline{e}_k), \quad (\text{D.1})$$

given $\underline{d} = (d_1, \dots, d_{n_d})$, $\underline{\underline{X}} = (\underline{x}_1, \dots, \underline{x}_{n_p})$ and $\underline{e} = (e_1, \dots, e_{n_e})$ as inputs. As a side benefit, information that is shared between different models with the same experiment, or between different experiments with the same model, can now be used efficiently by user-implemented likelihood classes.

Once a likelihood call is represented as a rank-three tensor, the different steps in SMC can be expressed as tensor manipulations and contractions. For example, the Larmor precession model (2.11) can be written using the outer product of a model and experiment vector,

$$\Pr(d = 0|\omega; \underline{t}) = \cos^2(\omega \underline{t}^\top). \quad (\text{D.2})$$

Note that this is a two-index tensor, as QInfer will optionally promote rank-two tensors to rank-three tensors for two-outcome models, where this can be done unambiguously. This design is in particular useful for adaptive experiment design wherein SMC updates must be performed on a range of hypothetical data and conditioned on a list of candidate experiments. The benefits of vectorized likelihood functions can be readily extended to Fisher score vectors and information matrices, as well as to Bayesian information matrices,

$$q_{ijkl} := \nabla_{x_i} \log \Pr(d_j|\underline{x}_k; \underline{e}_l) \quad (\text{D.3a})$$

$$I_{ijkl} := \sum_{k'} L_{k'kl} q_{ik'kl} q_{jk'kl} \quad (\text{D.3b})$$

$$J_{ijl} := \sum_k w_k I_{ijkl}. \quad (\text{D.3c})$$

D.2 Performance Testing

Given the importance of having an efficient SMC implementation, here we discuss the performance of QInfer, demonstrating that the library introduces minimal overhead on top of simulation calls. That is, we show that all other computational tasks, including resampling, are subdominant even for modest likelihood models. We additionally show that, when used with platforms such as multicore architectures or GPUs, QInfer can offer significant improvements to performance without increasing estimation errors.

Our strategy, then, is to test simple models such as the Larmor precession model (2.11), for various numbers of particles and for various implementations. We do so using the `qinfer.perf_testing` module, which automates the process of benchmarking the speed and incurred loss of an SMC inference procedure. This module also supports parallelization using IPython [269]. Here, we test the serial CPU-only model, `SimplePrecessionModel`, against the GPU-only model implemented by `AcceleratedPrecessionModel` and the parallel CPU-only model implemented by `DirectViewParallelizedModel`. We run each of the three models for 3000 trials of 100 measurements each, chosen from the exponentially-sparse sampling heuristic `ExpSparseHeuristic`. We then vary the particles from 2^6 to 2^{18} and consider the elapsed time per particle on a test system with a quad-core Intel Core i7-4790K processor, 32 GB of RAM and an NVidia GeForce 750 Ti on Ubuntu 14.10, using the latest development version of QInfer with Python 2.7, NumPy 1.8.2 and IPython 2.3.0. For particle numbers at least 512, the Bayes risk achieved by all three models were indistinguishable. The timing results are shown in Figure D.1.

These results show that for a fast model such as the Larmor precession model, no advantage is provided by parallelization. Past approximately 4000 particles, the GPU-only model offers an advantage for a small range before other costs begin to dominate.

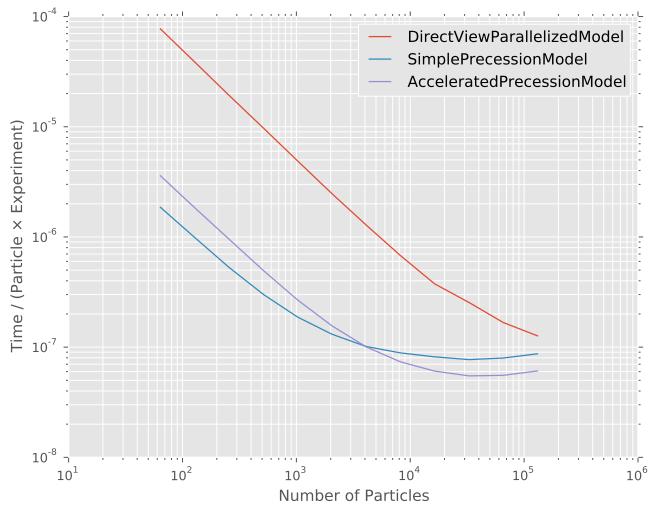


Figure D.1: Performance testing of serial and parallel CPU-only implementations versus GPU-only implementation of the Larmor precession model (2.11). The performance is measured as total elapsed time in seconds, normalized by the number of likelihood calls required.