Integrating tuning and readout techniques for scaling spin qubits

Joseph Hickie

Oxford, UK

March 2024

Integrating tuning and readout techniques for scaling spin qubits

Joseph Hickie

Supervisors

Professor N. Ares	Department of Engineering
Professor G. A. D. Briggs	Department of Materials

Acknowledgements

Thank you Natalia, the Fedes, Yikai, and Vincent, for leading the group and supporting our work. Of course, this would not have been possible without you.

Thank you: Barners, for your insight and ability to solve problems before I know they exist; Seb, for pushing me and yourself and thereby helping us both grow over the course of this; Will, for opening my eyes and sharing your view of the world, radically different as it is to so many others'. Kush, for your wide-ranging wisdom and intermittent visits to my office with something on your mind; Jonas, similarly, but with a bit of German flair; David, another intellectual powerhouse, but this time with a penchant for a little collegiate rugger. Brandon, we should have kept the quotes channel going. We were on to something with that. Jonathan, I enjoyed our musical conversations and sharing the odd glance of confusion.

Thank you Femi, for the chess. Jean, I wish I saw myself as you see me. Maša, you gem. Emma, for the noodles. My parents, for almost everything, but specifically the binoculars. Suzy, or Susannah, or whatever, I can't remember what I'm allowed any more... you're great and I'm proud of you. John and Conor, who in my mind often consist of a single entity, but one that I am immensely fond of. In honesty, the list could go on, and it will.

The Foxton boys, formative stuff. The Pigs, formative and destructive in equal measure. My housemates in Lancaster, all of them (apart from Arran). The

Europeans (not all of them - I am talking about my friends); Neale Marlow, who I spotted right away and likewise was spotted by; Leonie Schneider, who went the opposite route; Freddie, for making me uncomfortably full every time we went for lunch together; my housemates, for spending money to share the place you live with my new trombone; my football club and its members; the Manchester boys, for the jams and hikes. Anybody else who reads this and does not feel they have been fairly acknowledged, please submit a request for addendum to the University of Oxford.

Abstract

Spin qubits are one of the more realistic propositions for the basis of an operational quantum computer. However, scaling these systems is proving to be challenging for both the academic and industrial communities, in part due to the large parameter space that a spin qubit device lives in. To get what is effectively a transistor to act as a qubit, the experimentalist has to adjust the gate voltages that modify the spin's confinement potential to find the exact combination that results in a well-defined two-state quantum system. Even after bringing a qubit into existence, its performance is heavily parameter-dependent, requiring further optimisation of experimental control knobs.

Manually finding these parameters is time-consuming for even a single qubit. Part of the roadmap for scaling these devices past the proof-of-concept stage is automating these procedures, a goal which is slowly being tackled in the literature. This thesis contributes a series of algorithms towards the eventual goal of a fully-automated qubit tune-up process, specifically using measurement techniques that are used for qubit readout. I begin by presenting an algorithm that automatically tunes an uncalibrated quantum dot array into the double dot regime using only radio-frequency reflectometry measurements. Next, I present a package of algorithms that compensate for the effect of crosstalk between a quantum dot array and the quantum dot charge sensor used to measure its state. Third, I present a series of techniques

and measurements for use in the automated discovery of qubits within a double quantum dot regime. Finally, I present a method that extracts qubit readout metrics from repeated single-shot measurements, such that any one of them can be optimised using conventional optimisation routines.

In combination, these results contribute towards the ultimate goal of a scalable, fully-autonomous tuning pipeline for heterostructure quantum dot spin qubits.

Contents

Acknowledgements **iii**

Abstract **v**

1 Introduction **1**

1.1 Organisation of the thesis 3

2 Background **7**

2.1 Qubits and quantum computation 8

2.1.1 The Bloch sphere 10

2.2 Spin qubits 11

2.3 Electrostatically-defined quantum dots in semiconducting heterostructures 13

2.3.1 Coulomb blockade 14

2.3.2 Material choice 21

2.3.3 Implementing spin qubits 22

2.3.4 Readout 24

2.4 Current challenges for quantum dot-based spin qubit devices 25

2.5 Related work 27

2.5.1 Automatic tuning . . . 27
2.5.2 ML-assisted tuning . . . 29

3 Experimental and computational methods 39

3.1 Overview of experimental setup . . . 39
3.2 Reflectometry . . . 41
3.2.1 Impedance matching . . . 43
3.3 Measurements . . . 45
3.3.1 Charge stability diagrams . . . 45
3.3.2 Transport . . . 47
3.3.3 Charge sensing . . . 48
3.3.4 Fast measurements . . . 50
3.3.5 Data acquisition . . . 53
3.4 Computational methods . . . 54
3.4.1 Principal component analysis . . . 54
3.4.2 Gaussian processes . . . 57
3.4.3 Bayesian optimisation with Gaussian processes . . . 59

4 Tuning a double quantum dot using radio-frequency reflectometry 63

4.1 Introduction . . . 64
4.1.1 Experiment . . . 65
4.2 The rf hypervolume . . . 68
4.3 Algorithm . . . 69
4.3.1 Overview . . . 70
4.3.2 Gaussian process . . . 72
4.3.3 Classifying features against noise . . . 73
4.3.4 Double-dot feature score function . . . 74
4.4 Results . . . 77

4.4.1 Score function validation 77
4.4.2 Double-dot tuning 78
4.5 Conclusions 81

5 Charge sensor compensation 85
5.1 Introduction 86
5.1.1 Problem analysis: constant capacitance model 89
5.1.2 Experiments 91
5.1.3 Non-constant capacitance 93
5.2 Algorithm 94
5.2.1 Optimally tuning charge sensing dot barriers 94
5.2.2 Measuring gate-to-dot coupling 98
5.2.3 Compensating for crosstalk 101
5.3 Results 103
5.3.1 Validation of variance-based noise function 103
5.3.2 Compensated gate performance 106
5.4 Conclusions 109
5.4.1 Further work 111

6 Towards automated fine tuning of quantum dot devices in semiconductor heterostructures 113
6.1 Introduction 114
6.1.1 Experiment 115
6.2 Inter-dot transition identification 116
6.2.1 Stability diagram histogram fitting 116
6.2.2 Circular measurements for transition identification 118
6.2.3 Identifying transitions 120
6.2.4 Future experimental direction 121

6.3 Maintaining inter-dot position . . . 121
6.3.1 Control using charge transition points . . . 121
6.3.2 Virtual gates . . . 124
6.4 Identifying Pauli spin blockade . . . 128
6.4.1 PSB measurement sequence . . . 128
6.4.2 Difference measurements . . . 129
6.5 Conclusions . . . 130

7 Qubit readout optimisation **133**
7.1 Introduction . . . 133
7.2 Derivation of signal probability distribution function . . . 136
7.2.1 Parametric model . . . 139
7.3 Maximum likelihood estimation . . . 143
7.4 Simulator . . . 145
7.5 Algorithm . . . 148
7.5.1 Eliminating measurements without spin . . . 148
7.5.2 MLE parameter inference . . . 150
7.5.3 Optimal threshold voltage . . . 151
7.6 Results . . . 152
7.6.1 Simulator . . . 152
7.6.2 Spin qubit . . . 154
7.7 Conclusions . . . 157

8 Conclusions **159**
8.1 Summary of results . . . 160
8.2 Limitations and further work . . . 162
8.2.1 Automated tuning versus device reproducibility . . . 164

Bibliography **165**

A Impedance matching circuit derivation **187**

B Propagation of errors in score function **191**

C Derivation of optimal threshold voltage **193**

Chapter 1

Introduction

Look up 'quantum computing' on the world's favourite online encyclopaedia. The first line[1] states the following:

> A quantum computer is a computer that takes advantage of quantum mechanical phenomena.

It contains correct information, but is somewhat optimistic. We are yet to see the results of the promises of quantum computing.

Of the contending hardware platforms, spin qubits are not the current race leaders. However, their similarity to transistors, which are the world's most manufactured objects, has convinced some academics that they may be the key to a future of useful quantum computation. That said, there are significant challenges that stand in the way of such a future. A typical smartphone contains on the order of 10^{10} transistors; modern spin qubit devices have yet to move beyond the order of 10 qubits [1].

One of the field's major challenges is that the transistor-like devices require a complex *tuning* procedure before they behave like qubits. A spin qubit exists only when the

[1] as of 1st February, 2024

spin is confined, which requires a device-specific combination of gate voltages for the depletion or accumulation of charges. From here, we use nanosecond-scale pulses to manipulate the spins and measure their orientation. The system's operation and measurement fidelities are dependent upon tunable parameters, which makes for a fun, if rather frustrating, experimental playground.

Until recently, the tuning process has been an entirely manual affair. While the process can be somewhat time consuming, it represents something of a rite of passage for a PhD student in the field. However, the future of spin qubits relies on the ability to tune large arrays of quantum dot devices into the qubit regime. Measuring the state of these qubits accurately, and automatically calibrating the various inputs required for their operation, is also a critical step. Doing this by hand is laborious enough for a single qubit device, and is unimaginable for an array of, say, one hundred spin qubits. Useful quantum computers will need at least hundreds of qubits, if not thousands or millions, setting up a requirement for the automation of every stage of the tuning process. This is one of the key requirements I address in this thesis.

Alongside automation, measurement techniques have progressed from the early transport-based measurements to high-bandwidth radio-frequency reflectometry measurements that are sensitive enough to (indirectly) measure single spins. Spin qubits encode information through the state of a small number (typically between one and three) of electron (or hole) spins. Measuring their state necessitates exceptionally high sensitivity while maintaining robust noise tolerance. This requirement has led to the use of reflectometry to dispersively measure the state of the qubit via coupled charge sensors or device gate electrodes. The hardware necessary for these measurements is likely to become indispensable in the scale-up process, making their incorporation into the tuning procedure a natural progression. Additionally, using a quantum dot as a sensor implies the need for its own tuning procedure. This

thesis, in Chapters 4 and 5, contributes to this objective by addressing some of the complications that have thus far hindered the integration of advanced measurements with device tuning.

As devices become more homogeneous, it is likely that tuning algorithms will become less important. For the near term, however, they are still a critical part of the effort towards building a spin-based quantum computer. The fine-tuning of qubits, once initialised, is an area that is likely to remain important long into the future. In Chapter 6, I present an algorithm that optimises any measurable qubit metric by parameter extraction. Work in the literature typically looks into the physical reasons behind anisotropic qubit properties, such as Refs. [2, 3]; here, I propose a black-box solution based on Bayesian optimisation.

The key contributions of this thesis are a fast, reflectometry-based double dot tuning algorithm; a series of algorithms to automate the tuning and compensation of a charge sensing quantum dot; a series of tools to aid in the fine-tuning of a double quantum dot using fast and effective algorithms that replace manual approaches in existing labs; and an algorithm for improving the quality of a qubit once it has been defined in a double quantum dot.

1.1 Organisation of the thesis

There are no surprises in the layout of this thesis. In Chapter 2, I go over some of the existing literature and motivate my research. In Chapter 3, I discuss the key details of the experimental setup, as well as some of the main machine learning approaches that I use extensively throughout.

In Chapter 4, I tackle the problem of device tuning. A standard device architecture has not yet been agreed on in the community, with each device design needing different

combinations of gate voltages to form quantum dots. Furthermore, variability and inconsistency in the device manufacturing process means we cannot assume that devices with a shared design will share voltage configurations for specific physical regimes (such as a single or double dot). In this chapter, I present an algorithm that takes a device from being cold in the fridge to displaying characteristic double dot behaviour, using only advanced radio-frequency (rf) reflectometry measurements. This is a key contribution for two reasons. Firstly, our use of rf reflectometry allows for faster tuning algorithms than transport-based approaches such as in Ref. [4]. Secondly, transport is not sensitive enough to measure charge transitions in the low charge regime, which is typically where qubits are encoded. Reflectometry measurements are sensitive enough for this, meaning tuning algorithms based on this technique can take the device to the appropriate regime for qubits without a further tuning step.

In Chapter 5, I discuss charge sensor compensation. Quantum dot charge sensors are widely used in the community due to their high sensitivity levels, but their use is complicated by the crosstalk that occurs between the gates that are used to control the quantum dot array and the charge sensing dot itself. This crosstalk pulls the sensor into and out of Coulomb blockade, periodically changing its sensitivity from high to zero. To address this issue, I developed an algorithm that automatically calibrates compensated device gates that take into account this crosstalk and reduce its effect. By performing this compensation, my algorithm ensures that the charge sensing dot provides a constant level of sensitivity across the device voltage space. The algorithm is scalable to multiple charge sensors, which is critical for larger device arrays.

Chapter 6 presents advanced techniques for fully-automated fine-tuning of quantum dot devices using radio-frequency reflectometry. Fine-tuning is a critical step in

preparing quantum dots for qubit operations, requiring precise adjustments to achieve the desired electronic states. In this chapter, I introduce several methods designed to automate this process, significantly reducing the manual effort and time required. These techniques include advanced algorithms for identifying inter-dot transitions, maintaining stable charge states, and detecting Pauli spin blockade. By automating these fine-tuning steps, the methods presented here contribute to the development of scalable quantum dot architectures, moving us closer to the realisation of large-scale quantum computers.

In Chapter 7, I focus on optimising qubit readout, which is a crucial aspect of quantum computation that directly impacts the fidelity and reliability of qubit operations. This chapter introduces an algorithm that extracts qubit quality metrics from repeated single-shot measurements of the qubit's state. These metrics, which include readout fidelity, error rates, and coherence times, are essential for assessing and optimising the performance of qubits. The algorithm enables the optimisation of device parameters to improve qubit quality. By providing a systematic approach to evaluate and enhance qubit performance, this chapter contributes to the broader goal of developing robust and high-fidelity qubit readout techniques essential for practical quantum computing applications.

Chapter 8 concludes the thesis by summarising the key findings and contributions of this work. The algorithms and techniques presented across the chapters address critical challenges in the tuning, stabilisation, and optimisation of quantum dot qubits. The chapter reflects on the progress made towards fully-automated and scalable quantum dot architectures and discusses the limitations and potential areas for future research.

Chapter 2

Background

Computer power has come a long way since Charles Babbage designed his Analytical Machine in 1837. This machine, although never actually built, would have reportedly been able to multiply two 20-digit numbers in three minutes [5]. While at the time this was revolutionary, today a commercial-grade central processing unit (CPU) can perform over a billion such calculations per second, and state-of-the-art supercomputers are now in the regime of 10^{18} total calculations per second.

The exponential increase in computational power since the 1940s when the first electronic, general-purpose machines were introduced is arguably one of the great successes of the human species. In 1965, Gordon Moore (who went on to co-found Intel three years later) observed that the density of individual transistors in integrated circuits—which has a direct impact on the computational power of the device—doubled approximately every two years. This was initially a forecast for the following decade, but, with minor revisions, it has remained true well into the 21st century.

This trend cannot continue indefinitely. At some point, we will reach the physical, universal limits of manufacture. In 2008, basic calculations showed that the quantum limit will be reached by 2036 [6], at which stage the manufacturing resolution of

transistors will be constrained by fundamental physical limits like Heisenberg's uncertainty principle.

While classical computing is approaching its limit, a new form of computing—one based on the manipulation of quantum states—is in its infancy. Where classical computers use binary units of information, stored either as a 0 or 1, quantum computers encode information into two-state quantum systems known as qubits, which, by a fascinating result of quantum mechanics, can be 0, 1, or a superposition of those two states. It is this result, along with entanglement, that should allow quantum computers to tackle complex computational problems that would take an exceedingly long time on a classical computer.

General-purpose quantum computers currently lie far over the scientific horizon. Current research is focused on near-term intermediate scale quantum (NISQ) computation, with small numbers of noisy qubits. We are firmly in this era, for it is now possible to access quantum computers with over 100 qubits on the cloud via companies such as IBM. Today's research is mainly focused on scaling up existing systems, but questions remain as to which platform will be the eventual winner. Presently, the leading qubit candidates are superconducting [7], trapped-ion [8, 9], photonic [10], and semiconducting [1, 11, 12].

2.1 Qubits and quantum computation

The basic unit of quantum information is the qubit. A qubit can be realised using any two-state quantum system, of which there are many. However, there are other criteria into which a qubit must fall if it is to be the basis of a useful and usable quantum computer. A qubit's decoherence time—that is, the time scale before a qubit loses its information—is affected by its interaction with its environment, thus a successful qubit candidate will have limited interaction with the outside world.

This creates a counter-problem: limited interaction with the outside world means we, the experimenters, have limited control of the qubit itself. A useful qubit will sit somewhere in the balance with both control and readout possible within the coherent lifespan of the qubit.

A set of requirements for building a quantum computer was outlined by David DiVincenzo in 2000 [13]. In his eyes (and the eyes of those who follow him), a quantum computer requires:

1. A scalable physical system with well-characterised qubits;
2. The ability to initialise the qubits;
3. Qubit coherence times that are much longer than the gate operation time (by $\sim 10^4$ times [11]);
4. A universal set of quantum gates;
5. The ability to measure the state of each qubit.

Two decades of research later, these criteria are starting to be met experimentally [14, 15, 16, 17, e.g.] and basic quantum processors have been realised [18, 19]. The requirements for building a scalable quantum computer have been narrowed down, with a strong focus on experiments attaining error rates that can be tolerated by error correction algorithms. Recent advances in noise correction algorithms mean that noise errors of up to approximately 5% could theoretically be tolerated [20] under biased noise, although readout and gate fidelity errors of approximately 1% are generally accepted for standard error correction codes such as the surface code [21].

Most individual physical qubits do not perform well enough to satisfy the requirements of a quantum error correction code, so *logical* qubits are used instead. Depending on

the type of error correction used, a single logical qubit could be encoded from tens, hundreds, or even thousands of physical qubits [22, 23].

2.1.1 The Bloch sphere

A qubit's state is a superposition of its basis states. These basis states represent the analogue of the 0 and 1 of a classical bit, and are labelled $|0\rangle$ and $|1\rangle$ respectively. We describe the qubit's state by

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.1.1}$$

where α and β represent the probability amplitude of measuring each basis state. Measuring the qubit results in obtaining one of the two basis states, and as such $|\alpha|^2 + |\beta|^2 = 1$. By change of coordinates, we can reduce the degrees of freedom to two by

$$\alpha = \cos\frac{\theta}{2} \tag{2.1.2}$$

$$\beta = e^{i\phi}\sin\frac{\theta}{2}, \tag{2.1.3}$$

with parameters θ and ϕ. These parameters are typically used to visualise the state of a qubit on a Bloch sphere (Fig. 2.1.1). While classical bits would only have access to the poles (the basis states of the quantum bit), a pure quantum state exists at every point on the surface of the Bloch sphere.

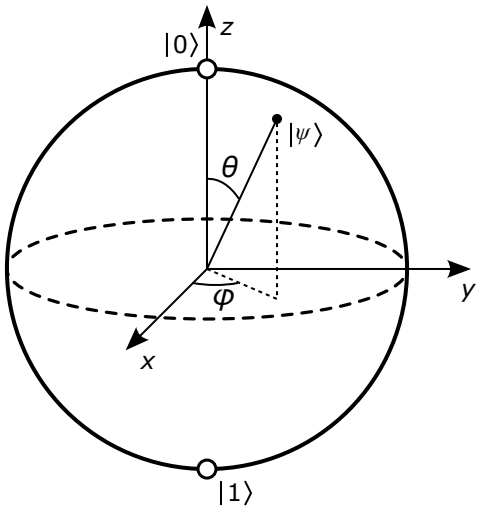


Figure 2.1.1: The Bloch sphere. On the poles are the basis states $|0\rangle$ and $|1\rangle$, which are the analogues of the classical 0 and 1. Image used under Creative Commons licence CC BY-SA 3.0 from [24].

2.2 Spin qubits

A spin qubit encodes information in the spin degree of freedom of one or more spins, depending on the implementation. Several proposals for spin qubit platforms exist; rather than describing them all here, I point the reader to recent reviews in Refs. [25], [26], and [27]. In this thesis, I will focus on spin qubits in gate-defined semiconductor quantum dots due to their compatibility with existing semiconductor fabrication facilities, small physical size [28], demonstrated coherence times [16], and gate fidelities [29].

Qubits based on electron spins in semiconducting quantum dots were first proposed by Loss and DiVincenzo in 1998 [30]. There are several motivating factors for using spins as the degree of freedom in which to encode quantum information. The first is that the magnetic moment of a spin is tiny (approximately the Bohr magneton, $\mu_B = 9.274{\times}10^{-24}\ \mathrm{J\,T^{-1}}$), meaning spins are weakly coupled to their environment. As a result, spin qubit devices have long coherence times, especially when implemented in nuclear-spin-free materials such as isotopically-purified silicon.

Additionally, spin qubits are scalable, both in terms of size and manufacturing capacity. The global semiconductor industry is well established and qubit devices are based on a similar architecture to existing transistor devices. That said, while the structure of such a qubit device is similar to that of existing transistor technologies, Vandersypen *et al.* argue [31] that there are still serious challenges to overcome due to the amount of control electronics required by a large-scale qubit array; they do, however, conclude that these challenges can be overcome. Quantum dots are tiny, on the scale of 100 nm, so they can be densely packed into a small area. For comparison, a large-scale trapped-ion computer would require millions of individual laser beams [8], with a two-billion trapped-ion qubit computer taking up the area of a football field. A computer with as many superconducting qubits would be smaller but still require approximately 25 m^2 [22, 32]. This is important when one considers that encoding a single fault-tolerant qubit will likely be made up of many—perhaps hundreds or even thousands [21]—of noisy, error-prone physical qubits [33, 34]. Spin qubits in quantum dots can be naturally scaled to two-dimensional arrays, for which well-established error correction codes such as the surface code exist with fault tolerance on the order of 1% [21]. Modular designs of solid state quantum computers could tolerate per-module errors of in excess of 10% [35].

Present research on spin qubit devices is focused on scaling the devices from the single- or low-qubit regime to many more. A six-qubit processor has been demonstrated in silicon [1], but practical quantum computation requires orders of magnitude more qubits to perform useful error-compensated calculations. The literature yields promising signs for our field [15].

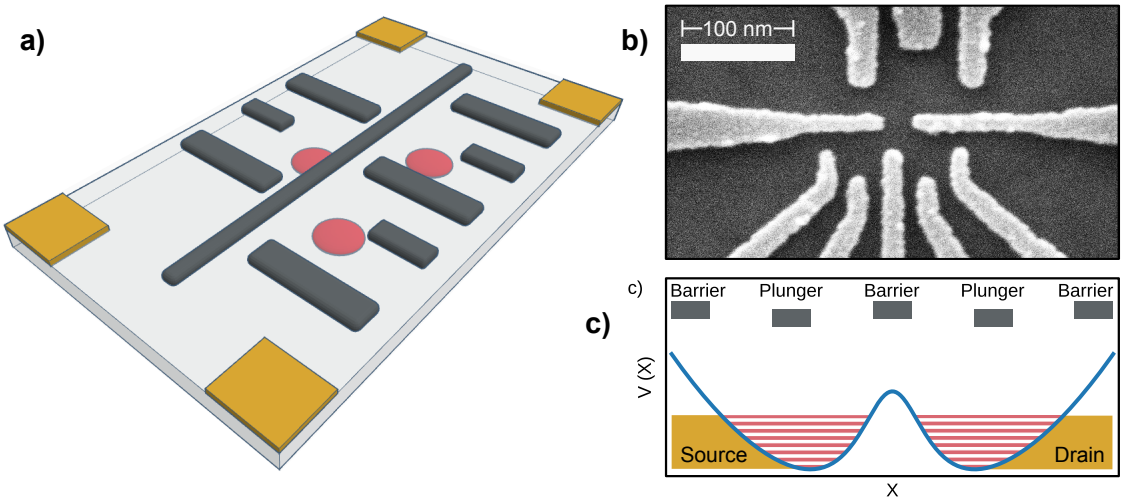


Figure 2.3.1: **a)** Cartoon of a double quantum dot array with a third quantum dot that is used as a charge sensor. The gates are in black, with the ideal location of the dots in red. The ohmic contacts are in yellow. **b)** A microscopy image of a real device similar to the cartoon. **c)** A cartoon representation of the electrostatic potential underneath each gate in the device. The source and drain potentials are indicated in yellow.

2.3 Electrostatically-defined quantum dots in semiconducting heterostructures

To form a spin qubit, we must confine the spin and be able to control and measure it. One approach for confining spins is to use a quantum dot. These structures allow the so-called zero-dimensional confinement of charge carriers (electrons or holes) in space. In the remainder of this work, *quantum dot* refers to an electrostatically-defined quantum dot in a semiconducting heterostructure.

To create a quantum dot in such a device, the first ingredient is to form a two-dimensional electron/hole gas (2DEG/2DHG) at the heterostructure interface. Here, charge carriers are trapped in a potential well and form a thin layer with a depth on the order of 10 nm. This confines the charge carriers to two dimensions, in the plane of the device. During fabrication, metal gate electrodes are patterned on the top of the heterostructure. We apply voltages to these gates, and the resulting electric

field, shaped by the gate architecture, selectively depletes the 2DEG/2DHG[1].

The architecture (layout) of the gates is such that, with the correct application of voltages, we can form isolated islands of charges, which we call quantum dots [11, 36]. The three-dimensional confinement in quantum dots results in the quantised electronic energy levels, this being the motivation for their nickname *artificial atoms*. Ohmic (low resistance) contacts are made to the reservoirs of the 2DEG/2DHG, which allow us to bias the device. Different gates control different parameters of the system, and are generally split into two types: *barrier* and *plunger* gates. Typically, barrier gates are used to set the inter-dot tunnelling rates, while plunger gates are used to control the electrochemical potential of a given dot (Fig. 3.3.1). Due to electrostatic (capacitive) coupling, known as crosstalk, single gates generally do not only control a single device parameter, but have an influence on many. The mitigation of this effect is the basis of Chapter 5.

2.3.1 Coulomb blockade

Electron transport through quantum dots exhibits *Coulomb blockade* (Fig. 2.3.2 **b)**). This effect is a direct result of the discrete energy levels and electron-electron interactions in quantum dots.

Charge carriers are loaded into a dot via reservoirs (ohmic contacts in Fig. 3.3.1). When a charge is added to a quantum dot, it experiences Coulomb repulsion due to the presence of the other charges on the dot. Due to this repulsive force, loading a charge onto a quantum dot requires an energy cost known as the *charging energy* E_C, given by

[1]In this passage I focus on depletion mode devices, for which we deplete a 2DEG/2DHG. Accumulation mode devices exist in which the material is engineered to be depleted of charge carriers. Rather than selectively depleting the 2DEG/2DHG, these devices accumulate charges under electrostatic gates.

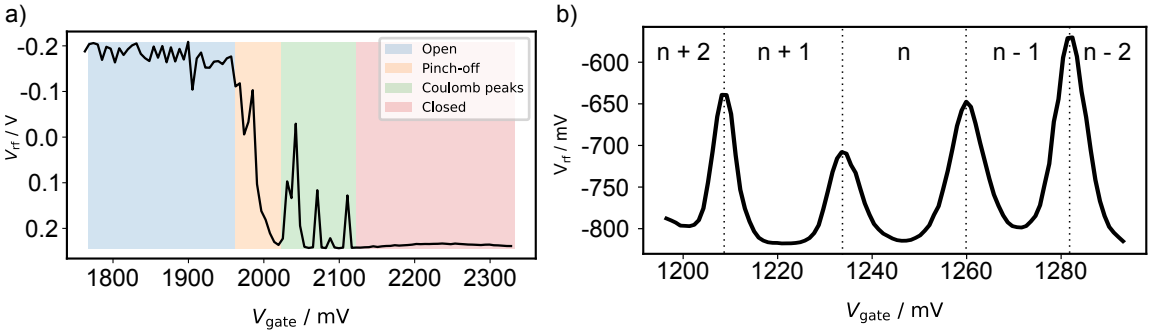


Figure 2.3.2: **a)** An example pinch-off curve of a device, with the sections of the trace labelled. When the device is open, current flows. During pinch-off, transport is rapidly suppressed. If the device is in the quantum dot regime, Coulomb peaks follow device pinch-off. With further closing, transport and tunnelling are totally suppressed. **b)** Coulomb peaks occur in the quantum dot regime; the number of charges occupying the dot is labelled. The number of charges fluctuates between n and $n+1$ within the Coulomb peak. When the number of charges on the dot is constant, the device is said to be in Coulomb blockade.

$$E_C = \frac{e^2}{2C}, \tag{2.3.1}$$

where e is the charge and C is the dot's total capacitance. For an approximate measure, we can model the dot as a circular capacitor with $C = 8\epsilon_r\epsilon_0 R$, where ϵ_r is the dielectric constant of the material (which, for germanium, is approximately 16), ϵ_0 is the dielectric constant, and R is the radius of the circle (on the order of 100 nm), leading to an approximate charging energy of $E_C \approx 1\,\text{meV}$.

At low temperatures, such as the sub-20 mK environment of a dilution refrigerator, the electron thermal energy k_BT (where k_B is the Boltzmann constant and T is the temperature) is $k_BT \approx 1.38 \times 10^{-3}\,\text{J K}^{-1} \times 20 \times 10^{-3}\text{K} \approx 2.76 \times 10^{-25}\text{J} \approx 1.72 \times 10^{-4}\text{meV}$. Since $E_C \gg k_BT$, charges cannot load onto the dot without additional energy, leading to Coulomb blockade.

We can provide the energy required to lift Coulomb blockade by tuning the dot's electrochemical potential using a nearby gate electrode. When the potential of the

quantum dot aligns with the chemical potential of the leads, the charging energy matches the energy of the charge in the lead, resulting in a charging event. As we continuously sweep the gate voltage, the electrochemical potential of the dot periodically aligns with the Fermi levels of the source and drain electrodes. This results in quantised conductance, known as Coulomb oscillations (Fig. 2.3.2).

Charge stability diagrams

Charge stability diagrams are one of the fundamental measurements for characterising quantum dots. In the total voltage space of the device, there are regions in which the charge occupation of each dot in the system is constant (within Coulomb blockade), separated by hyper-planes representing charge loading events. When we measure Coulomb peaks by sweeping a single gate voltage, as in Fig. 2.3.2, we are taking a one-dimensional slice through this space.

By sweeping multiple gates, we can map out regions of constant charge in the dot or dot array in a measurement known as a charge stability diagram. Typically, we sweep one gate per dot, meaning a double dot is characterised using a two-dimensional charge stability diagram.

Effects of crosstalk Crosstalk between gate electrodes and dots in an array heavily influences the shape of a system's charge stability diagram. Crosstalk occurs when a gate voltage intended to control the potential of one quantum dot also affects the potential of nearby dots. This coupling causes deviations from the ideal behaviour of each gate controlling a single dot.

In such an idealised system with no crosstalk, the charge stability diagram would resemble a grid (Fig. 2.3.3 **a)**). By introducing crosstalk, the transition lines rotate towards each other, since the gates are in effect linear combinations of the case without crosstalk (Fig. 2.3.3 **b)**).

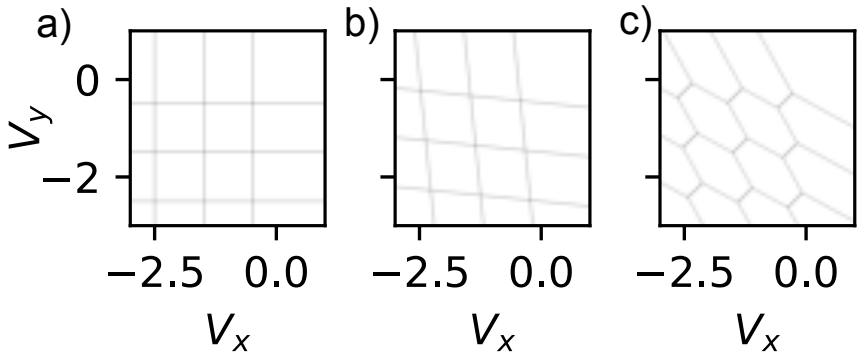


Figure 2.3.3: Three simulated charge stability diagrams. **a)** A diagram representing the case when the gates V_x and V_y do not have any crosstalk to the opposing dot. In **b)**, I have introduced crosstalk between gate V_x and dot y, and vice versa. In **c)**, the inter-dot transition splits the quadruple points into two triple points, as the $(n, m) \Leftrightarrow (n+1), (m+1)$ degeneracy is lifted.

Triple points In Fig. 2.3.3 **b)**, the corners of each charge stability region are quadruple points, representing voltages where four energy levels are degenerate. These points represent configurations where the electrochemical potential of both dots in the double-dot system align with the Fermi levels of the leads, allowing charge to transfer between the source and drain via both dots simultaneously. Typically, however, we observe that these quadruple points split into two triple points (Fig. 2.3.3 **c)**; Fig. 2.3.4). The transition connecting the two triple points is the inter-dot transition, which occurs when charge moves from one dot to the other. In dots that are capacitively coupled, the addition of an electron into one dot alters the electrostatic energy of the other dot. This splits the quadruple point into two triple points, and is identical to how charge sensing dots are used.

Lever arms The dot system is controlled using voltages applied to the electrostatic gates that define the dots. The *lever arm* quantifies the efficiency with which a gate voltage V_g translates to a change in the electrochemical potential μ of a dot. The lever arm is defined as

$$\alpha_g = \frac{\partial \mu}{\partial V_g}. \tag{2.3.2}$$

The lever arm of a specific gate is typically not constant over the device parameter space due to changes in capacitive coupling, leading to a non-linear relationship between gate voltages and dot chemical potentials.

The lever arm can also be defined as the ratio of capacitances in the capacitance matrix between the various dots and gates in the system (this concept is expanded in Section 2.3.1), given by

$$\alpha_G = -\frac{C_G}{C_\Sigma}, \tag{2.3.3}$$

where C_G is the capacitance between the dot in question and gate G and C_Σ is the self-capacitance of the dot [37]. The absolute value of the lever arm can be extracted by the geometry of features induced in the device's charge stability diagram by bias [37, 38], but in this thesis I am only interested in the ratio of the lever arms of different gates (Chapter 5).

Constant capacitance model

A quantum dot system can be classically modelled using an equivalent circuit framework known as the constant capacitance model [39]. The model describes arrays of quantum dots and their associated electrostatic gates as nodes in a network of fixed capacitors [40]. Each node i is modelled with a charge Q_i, along with an electrostatic potential V_i. The capacitive coupling between each node is given by a capacitance matrix $\mathbf{C}$, with the capacitance between nodes i and j given by C_{ij}, with $q_{ij} := C_{ij}(V_i - V_j)$. The total charge on node i is related to its and other nodes' potentials according to

$$Q_i = \sum_i q_{ij} = \sum_i C_{ij}(V_i - V_j). \tag{2.3.4}$$

The simplifying assumptions of the model are that the Coulomb interactions between charges in quantum dots can be modelled by constant capacitances, and that quantum effects can be neglected. By making these assumptions, we are able to model the key characteristic features of quantum dot arrays in an efficient manner [40]. This assumption of linearity also allows us to simplify the model into a matrix formalism,

$$\vec{Q} = \mathbf{C}\vec{V}, \tag{2.3.5}$$

where $\mathbf{C}$ is the capacitance matrix introduced earlier. We can split these matrices into sub-matrices to separate nodes representing dots from nodes representing gates:

$$\begin{bmatrix} \vec{Q}_d \\ \vec{Q}_g \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{dd} & \mathbf{c}_{gd} \\ \mathbf{c}_{dg} & \mathbf{c}_{gg} \end{bmatrix} \begin{bmatrix} \vec{V}_d \\ \vec{V}_g \end{bmatrix}, \tag{2.3.6}$$

where $\vec{Q}_{d(g)}$ represents the charge on the quantum dots (gates) and $\vec{V}_{d(g)}$ denotes the potential on the quantum dots (gates). The matrices c_{dd}, c_{gd}, c_{dg}, c_{gg} encode dot-dot, gate-dot, dot-gate and gate-gate capacitive couplings, respectively. The potential on each quantum dot can then be computed by

$$\vec{V}_d = \mathbf{c}_{dd}^{-1}(\vec{Q}_d - \mathbf{c}_{gd}\vec{V}_g). \tag{2.3.7}$$

The free energy of the system, $F(\vec{Q};\vec{V}) = U(\vec{Q};\vec{V}) - W(\vec{Q};\vec{V}) = \vec{V}^T\mathbf{c}\vec{V}/2 - \vec{Q}_g\vec{V}_g$, can then be written as

$$F(\vec{Q}_d;\vec{V}_g) = \frac{1}{2}\vec{Q}_d^T\mathbf{c}_{dd}^{-1}\vec{Q}_d - (\mathbf{c}_{dd}^{-1}\mathbf{c}_{gd}\vec{V}_g)^T\vec{Q}_d, \tag{2.3.8}$$

by dropping the terms that are not dependent upon $\vec{Q}_d$.

The allowed charge states $\vec{Q} = \pm e\vec{N}$ (for holes and electrons, respectively) are heavily constrained in a quantum dot array. In particular, for an open quantum dot array, i.e. when the array is coupled to fermionic reservoirs from which it can draw an arbitrary number of charges, the constraints are:

1. The number of charges must be integers, represented as $\vec{N} \in \mathbb{Z}^{n_{\text{dot}}}$.
2. The number of charges on any given quantum dot must be non-negative, meaning $N_i \geq 0$ for all i.

For a closed quantum dot array, i.e. when the leads are decoupled from the array, and the number of charges is fixed to a finite value $\hat{N}$ [41], there is an additional constraint:

3. The sum of all charges in the array must equal $\hat{N}$.

At $T = 0$ K, the system will adopt the charge state $\vec{N}^*$ that minimises the system's free energy whilst obeying the appropriate constraints, such that

$$\vec{N}^* = \underset{\vec{N} \in \mathcal{N}}{\arg\min}\, F(\vec{N}; \vec{V}_g), \tag{2.3.9}$$

where $\mathcal{N}$ is the set of all admissible charge configurations for the open or closed regimes of the quantum dot arrays. By computing the ground state charge configuration as a function of the gate voltages, the constant capacitance model captures the charge stability diagrams of such quantum dot arrays when the tunnel couplings between quantum dots are weak.

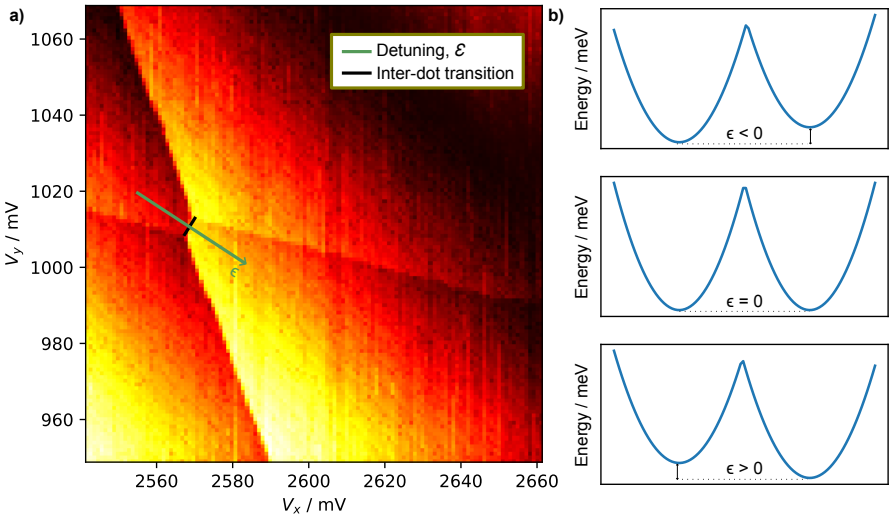


Figure 2.3.4: **a)** An inter-dot transition in a double dot, with the detuning axis labelled. Here, V_x is the plunger for the left dot, and V_y is the plunger for the right dot. The detuning axis is defined as perpendicular to the inter-dot transition line. **b)** A schematic of the energy levels within the dots as a function of detuning.

Dot detuning energy

In a double dot system, it is useful to define an axis that directly controls the difference in energy of the two dots. Along this axis, the average energy of the two dots remains constant, but the difference between them changes [11] (Fig. 2.3.4 **b)**). The detuning axis allows electrical control over the position of the electron/hole wavefunctions that inhabit the dot, which in turn give us electrical control of their exchange coupling.

2.3.2 Material choice

The fundamental research into quantum dots and whether they could be used for quantum information processing was generally conducted using gallium arsenide quantum dots [11]. The low effective mass of electrons in this material implies a larger spatial wavefunction, which in turn leads to relaxed fabrication constraints [42], permitting academics to fabricate them in university laboratories. However, research is now focused on scaling and improving qubit performance. For this purpose, gallium

arsenide is a poor material; the presence of nuclear spin has a negative impact on the coherence of spins in the dot.

An obvious choice for the next phase of development is silicon due to its dominance within the global semiconductor industry. Isotopically-purified silicon-28 has no nuclear spin, leading to improved coherence times. However, electrons in silicon have a large effective mass [42], meaning the manufacture of silicon quantum dots places stricter requirements on the manufacturing processes due to the smaller spatial wavefunction relative to gallium arsenide.

Significant progress has been made on germanium-based devices [43, 44], as this material has a combination of the benefits of gallium arsenide and silicon. The low effective mass leads to relaxed manufacturing constraints [45], and it may also be isotopically purified into a nuclear-spin-free isotope [46].

Germanium hole spin devices

Most of the measurements in this thesis were made on hole-spin germanium devices similar to those described in Ref. [47]. These devices typically consist of an array of two to three quantum dots, with a splitter gate separating this array of dots from an additional quantum dot, designed for use as a charge sensor (Section 3.3.3). Hole spins in germanium are of particular interest because of holes' strong spin-orbit coupling [48], allowing all-electrical spin manipulation at high speeds.

I am very grateful to primarily Georgios Katsaros and his group at IST Austria for their work in fabricating the majority of the samples that I used during my doctorate.

2.3.3 Implementing spin qubits

There are several spin qubit architectures in existence. Presently, the leading candidates are single-spin (Loss-DiVincenzo) [30, 49], donor spin [50, 51], singlet-

triplet qubits [12, 43], and exchange-only [13].

Loss-DiVincenzo qubits

The most basic spin qubit is that which uses a single spin to encode information, with computational states $|\downarrow\rangle$ and $|\uparrow\rangle$ representing the spin-up and spin-down spin states respectively. The spin states are energetically split by applying a static magnetic field via the Zeeman interaction. An oscillating magnetic field is applied perpendicular to the first at a frequency equal to the Zeeman splitting to drive oscillations between the states.

Several approaches have been taken to apply the high (several gigahertz) frequencies required for single qubit operations, including direct magnetic drive (electron spin resonance) [15] and indirect drive with electric fields (electron dipole spin resonance) [14, 52]. The latest work in hole spin qubits harvests the strong hole spin-orbit coupling, which is intrinsic to the material. By applying an oscillating electric field to device gates, the charge carrier is spatially perturbed, and thus experiences an effective magnetic field, allowing all-electrical driving of spin transitions [53].

Singlet-triplet qubits

Singlet-triplet qubits use two spins in a double quantum dot to encode quantum information. They are typically encoded in the $S - T_0$ subspace, since $m = 0$ for both states of these states, leading to global magnetic field noise suppression [25]. The electrically-controllable exchange energy between the hybridised singlet and triplet states allows for z-axis control, while the x-axis rotation is magnetically driven by g-factor differences [43] or magnetic field strength gradients [54]. Singlet-triplet qubits can be operated with all-electrical, baseband pulses [54].

2.3.4 Readout

Directly measuring a single spin is challenging due to its small magnetic moment. Instead, various methods have been developed that allow us to infer the spin state by measuring the device's charge state [11]. The first electrical single-shot spin measurement in a single quantum dot used the Zeeman splitting of two spin states to create spin-selective transport [49].

Pauli spin blockade

In a double quantum dot system, we can leverage the Pauli exclusion principle for spin-to-charge conversion. The exclusion principle prevents aligned spins from simultaneously occupying the same quantum state within a quantum dot. Critically, spin is typically conserved under tunnelling events [11, 55].

For two spins in a single dot at a finite magnetic field, the singlet $S(2,0)$ is the energy ground state, separated from the triplet states by the Zeeman splitting. If we prepare the system into a triplet $T(1,1)$ state and pulse the dot energy landscape back to the configuration in which the singlet $S(2,0)$ is the ground state, the transition is blocked because the triplet $(2,0)$ state is higher in energy than $T(1,1)$ due to the exchange interaction. This also occurs in transport, the only difference being that rather than loading charges by applying ac pulses to the gates, a bias applied to the source and drain electrodes provides the energy for the loading and unloading of charges (Fig. 2.3.5). Tunnelling occurs when the $T(2,0)$ relaxes to $S(2,0)$, on a time scale typically quantified by the qubit T_1 time. Pauli spin blockade is used for the readout of spin qubits in a variety of materials [53, 54, 56], is robust against high temperatures [57], and is observable in measurements made via charge sensors [58, 59].

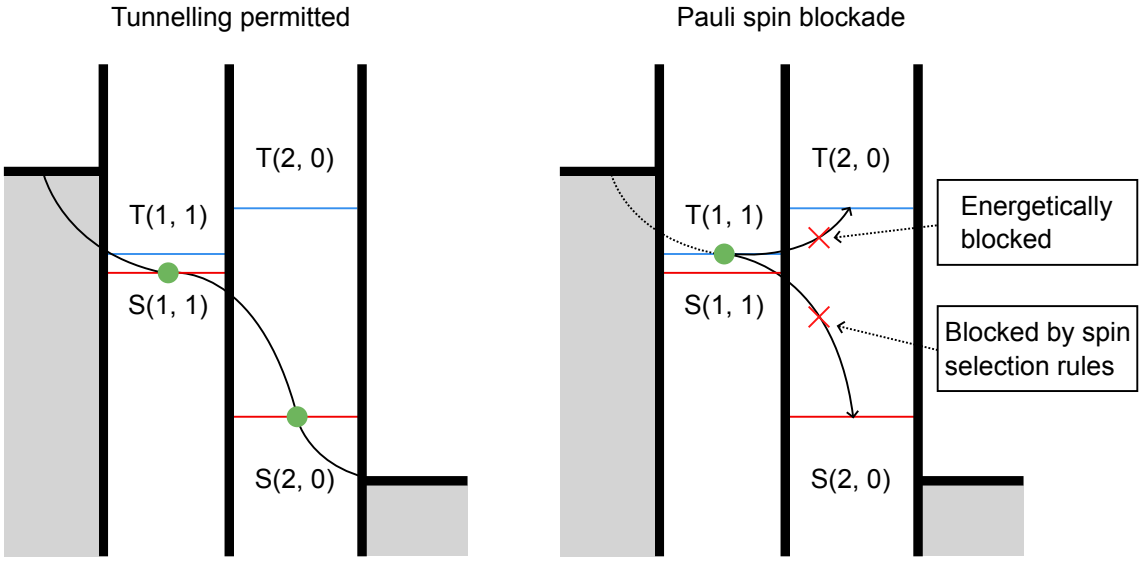


Figure 2.3.5: Pauli spin blockade. In this diagram, a bias is applied such that transport occurs from left to right. When a singlet state S(1, 1) is loaded, the S(2, 0) state is energetically accessible, leading to tunnelling event. If the prepared state is T(1, 1), the T(2, 0) state is energetically inaccessible and the S(2, 0) state is blocked by spin selection rules, leading to Pauli spin blockade.

2.4 Current challenges for quantum dot-based spin qubit devices

Multi-qubit processors have been demonstrated in gallium arsenide [60], silicon [1], and germanium [61]. While there is plenty of research ongoing into such challenges as qubit addressing [62], error correction [63], and architectures for larger arrays [62, 64], there is limited work that demonstrates how we are going to tune these devices to the regime in which they can host qubits. A cold quantum dot device is not a qubit, and there stands one of the major challenges of spin qubits: the tuning problem.

The tuning problem

To form a qubit, we first need to find the gate voltages that result in the formation of quantum dots. This is achieved in certain regions of the gate voltage parameter

space and can take an experienced operator several hours or more to reach. Once the dots are formed, we must tune the device into a specific regime that depends on the qubit implementation. For instance, the tunnel rates must be adjusted such that charge carriers are kept on the dots for sufficiently long periods of time. The process of preparing a quantum dot in this way is referred to in the literature as *coarse tuning*. Following coarse tuning, the qubit has to be encoded. The process of defining and refining the qubit's operating parameters is called *fine tuning*. Semiconducting heterostructures display variability in the potential landscape between devices due to minute fabrication errors and charge defects, meaning a given set of gate voltages that formed a quantum dot in one device will not necessarily form a quantum dot in another [65]. Equally, we must calibrate pulse schemes for qubit operation on a per-device basis. While improved fabrication will improve device reproducibility, designing methods for tuning devices that incorporate some flexibility to account for such device imperfections will improve their robustness.

For small-scale processors, the tuning problem is an experimental pain. However, as devices increase in complexity beyond a few qubits, manual tuning will become impossible. Tuning devices quickly and automatically is an important step on the path to a large-scale spin qubit processor. This problem has spawned a sub-field with several groups publishing work on coarse and fine tuning based on pattern matching [66, 67, 68, 69], neural networks [70, 71], signal processing [72], heuristics [73], and Bayesian optimisation [4].

Sensitive measurements

Measuring the charge state of a quantum dot is most easily achieved using transport (current) measurements. However, dc transport measurements cannot be used to measure charge rearrangements that do not lead to charges tunnelling to the ohmic contacts via the leads (or *reservoirs*), since this is where the detection occurs. The

alternative is to indirectly sense charge rearrangements using a sensitive electrometer, for which various proposals have been suggested and tested. Quantum point contacts [74, 75], quantum dots [76, 77], and device gates [78, 79] all have been shown to work as charge sensors with high sensitivity.

However, all of these sensors have problems. The electrostatic potentials that define quantum point contacts and quantum dot charge sensors are capacitively coupled to the gates that control the device array, leading to patchy sensitivity across device gate voltage space. Dispersive gate sensing is a relatively unstudied technique with promising results [80, 81], but is not yet as widespread as QPC or quantum dot sensors. One limitation is that this method measures changes in capacitance when the rf signal drives a charge across a transition, meaning inter-dot transitions must have a transmission frequency similar to the rf drive frequency in order to be visible [82].

2.5 Related work

2.5.1 Automatic tuning

Some headway has been made into algorithmically tuning quantum dots. In 2016, Baart et al. released the first paper specifically approaching the subject of computer-controlled tuning [66]. However, their approach relies on pattern matching and is slow, taking up to 17 hours per run. It is also a computer-aided, not automated, approach, which is not the eventual goal. A significant drawback of this algorithm's design is in its scalability. Plotting a stability diagram for each single dot and then comparing this to a predetermined reference cross would make a similar algorithm for a large array unfeasibly slow. A bottleneck is in the two-dimensional scans, and these should be as limited and small as possible in an improved algorithm for array

tuning. Faster readout would also improve the speed of this algorithm.

Further work by multiple groups focused on adding automated fine tuning to the techniques developed by Baart et al. Botzem et al. builds on that work to go from a tuned quantum dot—i.e. a double dot in the single-electron regime—to an initialised singlet-triplet qubit [67]. This paper determines the qubit parameters from the tuned double dot, namely the initialisation, manipulation, and read-out points. The algorithm fits data to a model and thereby infers parameters which then inform the remainder of the tuning procedure.

They employ virtual gates in the fine tuning, which are linear combinations of the real gates that correct for cross-capacitances. For instance, when a plunger gate above the first dot is swept, it affects the electrochemical potential of the first dot, but it also has the unintended consequence of affecting the potential in the region around this point, which can include the second dot and certainly the gates used to control it. Virtual gates get around this problem by compensating for this unintended disturbance using more than one gate in combination to alter the potential landscape in a more controlled manner. Virtual gates are used in a number of papers in the literature [66, 67, 68, 69, 83, 84].

An algorithm with a similar outcome was proposed by van Diepen et al. [68]. Again, they start with a coarsely-tuned double dot and form virtual gates for use in a small voltage window around the inter-dot charge transition point. The inter-dot tunnel coupling is then tuned by incrementing gate voltages in a direction that adjusts the coupling strength towards a pre-defined value. When the measured value is within a given threshold of this target value, the algorithm terminates. In this algorithm, image processing is used to calculate the settings for the virtual gates. While a useful automation step, it is not generalisable beyond a double-dot device because it uses knowledge of the shape of a triple point. In a triple-dot device, for instance,

where a quadruple point exists in higher dimensions, this method could only work by splitting the device up into two double dots.

Mills et al. [69] focus on automatically deriving virtual gates from the device's stability diagram using image processing techniques. Whilst automatically calibrating virtual gates could certainly be useful for the coarse tuning of arrays, Mills et al. manually tune their device in [69] to the triple-dot regime. This step is the one that needs automating if large, multi-dot arrays are going to be routinely tuned using the virtual gate method. However, a serious problem with virtual gates is that they are not valid throughout the gate voltage space. Due to non-linear couplings and charge imperfections, virtual gates work only in the small region in which they have been defined. Thus they must be defined in the region in which they are going to be used, which is generally the tuned regime. This is a sticking point and means that a different approach will have to be found for defining virtual gates if they are to be used for tuning the device to that regime in the first place.

2.5.2 ML-assisted tuning

By 2019, the field had progressed to using machine learning to automate steps that historically had involved experienced human researchers. Using convolutional neural networks (CNNs), Kalantre et al. developed an algorithm that classified the state of quantum dot devices (namely how many dots are formed in the device) based on their stability diagrams [70]. To begin with, the paper demonstrates how a CNN can be used to convert between simulated current measurements showing Coulomb peaks to the charge configuration of the corresponding quantum dot, which it does with an accuracy of 91%. They also demonstrate a CNN that can categorise the state of a simulated five-gate device into one of four categories: a barrier (where no current flows), a single dot, double dot, or short circuit, where current flows freely. The

training data is simulated based on a simple nanowire model and the CNN again produced labels that agreed with the true labels as simulated with an accuracy of 91%. Based on this classifier, they were also able to demonstrate the coarse tuning of a simulated device. The classifier's output was fed into a numerical optimiser that moved the device through the voltage space until the required state was achieved. This was then demonstrated on a real device, for which it took 40 to 50 iterations of the algorithm to reach the double dot state, depending on the starting configuration. While this is quite slow, considering that two-dimensional current scans have to be taken at each stage, the algorithm took the device to the required double-dot state successfully 88.5% of the time.

The Kalantre et al. approach demonstrated that machine learning could be used to replace experienced researchers in time-consuming tasks such as automatic coarse tuning. Additionally, the fact that they were able to tune a real device using only simulated data was a clear demonstration of the power of machine learning in this field. However, the algorithm was not highly optimised and did not intelligently reduce the number of measurements it made, which, as discussed previously, is generally a bottleneck in tuning these devices.

Lennon et al. [85] demonstrated how machine learning could be used to optimise measurements of a quantum device. In this paper, they used a conditional variational auto-encoder (CVAE) to reduce the number of measurements required to characterise a quantum dot device and map out its charge configurations. While not directly relevant for preparing a quantum dot array, this paper demonstrated that machine learning approaches can be used to optimise measurements and reduce the amount of time necessary to perform steps in these types of algorithms. The basic principle is that certain measurements provide more information than others, and using machine learning it is possible to identify and record those more informative measurements

than the less informative.

Durrer et al. present work to tune a double quantum dot into a predetermined charge state [71]. This is somewhere between coarse and fine tuning, as it requires that the device is already tuned into the double dot regime, but is not quite fine tuning in that it can be used to bring the device into any specific charge state, which is useful for changing between qubit operation points in gate space. In summary, the algorithm presented in this paper moves the charge state of a double-dot device (n, m) (indicating n charges on the left dot and m on the right dot) to a pre-selected charge configuration (i, j). This is achieved in two steps: firstly, it takes the device from (n, m) to $(0, 0)$. Secondly, it loads electrons into each dot until the desired configuration is reached. To do this it uses a neural network (NN) which can recognise the charge transition lines and hence it knows how many electrons have been loaded.

Although this is an interesting use of NNs in improving the tuning of quantum dots, it is unclear how much of a benefit the NN provides in this case. Taking one-dimensional current scans is far quicker; a one-dimensional scan could be taken with one plunger until no charge transitions are observed, and this could then be repeated in the remaining dimension. When this point is reached, a two-dimensional scan could be taken. If there are no current peaks in this region, the $(0, 0)$ charge state has been reached. However, the accuracy of the NN in this experiment demonstrates that they can be used reliably in these types of algorithms. The NN correctly predicted whether a given state was occupied or empty 98.9% of the time, with the first part of the algorithm—the step that takes the device to an empty configuration—working 90% of the time. A common theme is developing here: this algorithm also starts with a fully-tuned double dot, where the unknown is the charge configuration. While this type of algorithm is useful for fine-tuning and preparing a double dot into a specific charge configuration for use as a qubit, actually initialising the double dot to

begin with is still taken for granted.

Teske et al. [72] have demonstrated the use of machine learning in fine-tuning a double quantum dot. To surmount the challenge presented by the various non-linearities in gate and tunnel barrier couplings, the authors used a Kalman filter to monitor and update the gradients of these couplings such that they can automatically tune the coupled parameters. These parameters are the inter-dot coupling strength and the singlet reload time, which is related to the reservoir-dot coupling strength of one of the dots. These parameters are controlled by four of the nine gates used to define the double dot device and have a non-linear response to these voltages.

Their algorithm checks if the parameters are in the desired range, and if so, terminates the program. If not, the Kalman filter is used to calculate the parameter function's gradient, which is then used to update the gate voltages using Newton's method. The algorithm demonstrates rapid fine-tuning of their qubit and employs interesting and useful machine learning techniques, but is still lacking in one major regard: it is assumed that the device is tuned to the double-dot regime to begin with. While the authors briefly mention the course tuning ideas discussed in the previous section, this is not the topic of the paper.

Darulová et al. present a software package for tuning double quantum dots [86]. Prior information is required about the device's architecture and the experimental setup. Initially, the algorithm characterises each of the device's gates and uses a binary classifier to determine whether they work or not; if all gates work, the device is viable and the algorithm proceeds. Then, by running a series of one- and two-dimensional scans and analysing the outcomes using a set of binary classifiers, the device is brought into the required state regime. This algorithm automates the measurements and their analysis, including finding the voltage range which will contain features of interest in which to run the remainder of the algorithm. Based on some rules

generated by prior experience with good single- and double-dot regimes, the location of the two-dimensional scan is updated and then classified by three classifiers, trained to detect the regime (classification into either single- or double-dot) and classify the quality of a single- or double-dot, agnostically of which is present.

This algorithm was demonstrated to work in both identifying dysfunctional devices and also tuning functional devices to the double-dot regime in all but two cases in the tests. The techniques employed, however, are quite similar to other algorithms in that the primary use of machine learning is to classify scans. If the resulting classification is not in the regime required by the experimenters, some parameter is iterated and the procedure starts again. While this reduces the requirement for human intervention, it does not improve the efficiency of the tuning. Furthermore, the algorithm put forward by Darulová et al. requires detailed knowledge of the gate architecture of the device; while this can presumably be changed in the experimental configuration, it does make the algorithm overall less generalised. The set of rules for updating the current ranges in which to measure are based on previous knowledge and are heuristic; if a device can be reproduced reliably such that these rules work between devices, then this is not an issue. However, in the real world of semiconductor fabrication, this is unlikely to be the case, and even if it is, device architecture design is a field of its own and is regularly updated. Eventually, these rules will require updating, and at that point, the algorithm will have to be updated or rewritten entirely.

Moon et al. [4] fully tune a double quantum dot, with no prior knowledge of the device's gate architecture (other than identifying gates as either plungers or barriers), in approximately 70 minutes. The authors identified that interesting transport features, including double dots, occur at the edges of a hypersurface in N dimensions, where N is the number of gates on the device. The bottleneck in tuning these devices

is taking the two-dimensional scans with the plunger gates of each dot, so to be efficient, the algorithm must aim to reduce the number of these scans. The key observation made by Moon et al. is that large portions of the hypersurface described above do not contain transport features of interest, and exploit this to reduce the number of measurements taken. The hypersurface model is very general, and is not informed by the device architecture; instead, it is informed by measurements fitted with a Gaussian process.

There are two stages to the algorithm: the sampling stage, and the investigation phase. The sampling phase generates candidate coordinates of the hyperplane—that is, candidate voltages for each of the gates—which are worth investigating. The investigation phase then makes measurements at those parameters. The results from the investigation phase are pumped back into the sampling phase to inform the algorithm's decision of which area of the hypersurface it should sample next. If Coulomb peaks are detected by the investigation phase, further measurements are made in this region to detect whether that regime is a double dot or not. Those regions which are double dots are scored to indicate the double dot quality.

Critically, there are no heuristic rules such as those found in the Darulová paper. This is the approach's major advancement over any other in the field: rather than being based on a set of rules that, for instance, increment one voltage if a given condition is reached, the whole algorithm is probabilistic. This means that the approach should, in theory, work for any device architecture with only a few minor adjustments.

However, while it is agnostic to device architecture, it does have limitations, particularly in terms of scalability to array devices. The algorithm is designed to work with transport measurements since fast readout methods such as fast radio-frequency reflectometry have to themselves be tuned to operate in an arbitrary voltage space.

While it is a relatively fast algorithm, it could be further optimised using reflectometry readout to reduce the time spent making two-dimensional scans. However, the concept of a hypersurface does not exist in reflectometry readout. In transport measurements, the current drops to zero at given gate voltage configurations. Reflectometry returns measurements even when no current flows through the device, so the pinch-off detection on which this algorithm relies would have to be redesigned to work with reflectometry-based readout methods. An additional problem is due to dimensionality: as the number of dots in an array increases, the dimensionality of the voltage space increases too. This leads to an enormous increase in the time and computational power required to run a similar hypersurface-based algorithm scaled up to the dimensions required for the control of a multi-dot array device.

The earlier work of Kalantre et al. [70] has been implemented in real-time as part of an experiment by Zwolak et al. [87], with some minor modifications. Their CNN was also trained on simulated data, using the Thomas-Fermi approximation like the Kalantre et al. paper, but it appears they generated their own. To introduce some variability into the dataset, the data are simulated as if they were from a series of different devices with different architectures and characteristics. As with Kalantre, the output from the CNN is used to create a fitness function corresponding to the state of the device. Zwolak et al. added the slight modification in the addition of a penalty cost for tuning to non-double-dot voltage space. Despite the promise from the Kalantre paper, some problems arise when this algorithm is implemented in a real device. Thirteen of the 38 originally-initialised scans are terminated due to inflexibility in the algorithm; the CNN is trained on data corresponding to a quantum dot formed in the middle of the plunger gates, when in actuality a dot can be formed to the left, in the centre, or to the right of these gates. Any dot that does not form in the central space is mislabelled by the classifier. This error propagates through the algorithm, as these mislabelled dots go on to be identified as double

dots by the optimisation phase, meaning it terminates incorrectly at a single dot. In the remaining scans, those that started too far away from the double-dot parameter space generally did not work, failing 9 times out of 11. Of the remaining 14 runs, 12 were successful. The reason for the high failure rate is because the fitness function is simply too flat in regions far from those of interest for the optimisation routine to work effectively. In any machine learning algorithm, it is important that the fitness function has been carefully thought through to avoid this sort of problem.

While some progress has been made in using machine learning techniques for tuning quantum dots, the techniques hitherto proposed only tune up to double dots. Depending on the qubit architecture implemented, a double dot is only good for between one and two qubits.

Some work has been done on designing algorithms to tune arrays of quantum dots that could be the basis for multi-qubit processors. In mid-2019, Volk et al. tuned an eight-dot array using their so-called $N+1$ method [84]. In doing so, they highlighted many difficulties associated with tuning arrays of dots. In the stability diagram of a double dot, the charging lines can be easily associated with each dot based on the coupling of each line to each of the plunger gates. However, with multiple dots, associating a charging line with a specific dot quickly becomes non-trivial due to different dots having similar couplings to each of the scanned plunger gates.

In summary, their algorithmic approach is as follows. Firstly, determine the pinch-off characteristics of all the gates involved and check that individual dots can be formed. Next, tune a double dot and nearby sensing dot. Following this, the iterative stage is reached: the cross-capacitances of each gate and the electrochemical potential of each of the formed dots is measured (including the charge sensor); these values are then used to populate a cross-capacitance matrix (CCM). Virtual gates are then calculated using the CCM, and these are then used to form the adjacent dot. By

using virtual gates, the potentials of the existing dots are not disturbed and so they are kept in their existing regime. The CCM is then updated with new values from further cross-capacitance measurements, and further dots are added sequentially.

The approach put forward in this paper may be a good starting point for an ML-based algorithm for the coarse tuning of a quantum dot array. In its current form, the approach is slow and not optimised. Forming the virtual gates is done by scanning every gate in the whole device every time a new dot is added. To add a new dot, the virtual barrier gate is raised incrementally in the range determined by the initial transport measurements, and for each of these new voltages a full two-dimensional scan is taken to identify whether a new dot has been formed, the marker for which is a new charge transition line on the scan. This is 'optional' according to the paper; presumably, a one-dimensional scan in the correct voltage space could be substituted and the charging line identified with a Coulomb peak in this scan. The $N+1$ method has been used in further papers from the group [88].

Once virtual gates have been established, they only work in a small parameter space. If they are to be used for a large array, this step would have to be optimised and automated, perhaps using a method similar to that used by Mills et all [69] as discussed previously. However, in that approach, no attempt was made at optimising the processes using machine learning, and for a large array, this approach would be very slow.

A second concern with the use of virtual gates is that while they compensate for dot potentials, they do not automatically compensate for the effects of gates on inter-dot couplings. This issue has been tackled by Hsiao et al. [89], who have introduced virtual gates that not only correct for the plungers but also the tunnel barriers. In traditional virtual gates, the potential of each dot is compensated for, but using them changes parameters such as the tunnel coupling strength. This paper introduces a

method that incorporates this into virtual gates such that they also do not change the strength of the tunnel barriers. The key observation is that despite the non-linear (in fact, exponential) dependence of the tunnel coupling on the gate, the exponents themselves are linear combinations of the gate voltages. Thus, once the relevant parameters have been extracted from measurements, the authors attain virtual gates that compensate for the tunnel coupling between the dots in their array.

Chapter 3

Experimental and computational methods

In this chapter, I outline the details of the experimental setups used to take the measurements that make up this thesis, and explore some of the technical details of measuring spin qubit devices. The various setups I used had more in common than in difference; this chapter is a general overview of those similarities.

3.1 Overview of experimental setup

State-of-the-art experimental quantum computing laboratories are, for the most part, characterised by cryogenic temperatures and both dc and high-frequency electronic instrumentation. The device under test is known as a sample (Section 2.3.2). The purpose of our experiments is to measure how the sample reacts to our inputs, and eventually to tune various experimental parameters to the point that it behaves as a controllable two-level quantum system. A cartoon of a typical measurement setup is presented in Figure 3.1.1. In the following, I go over the main aspects of such a

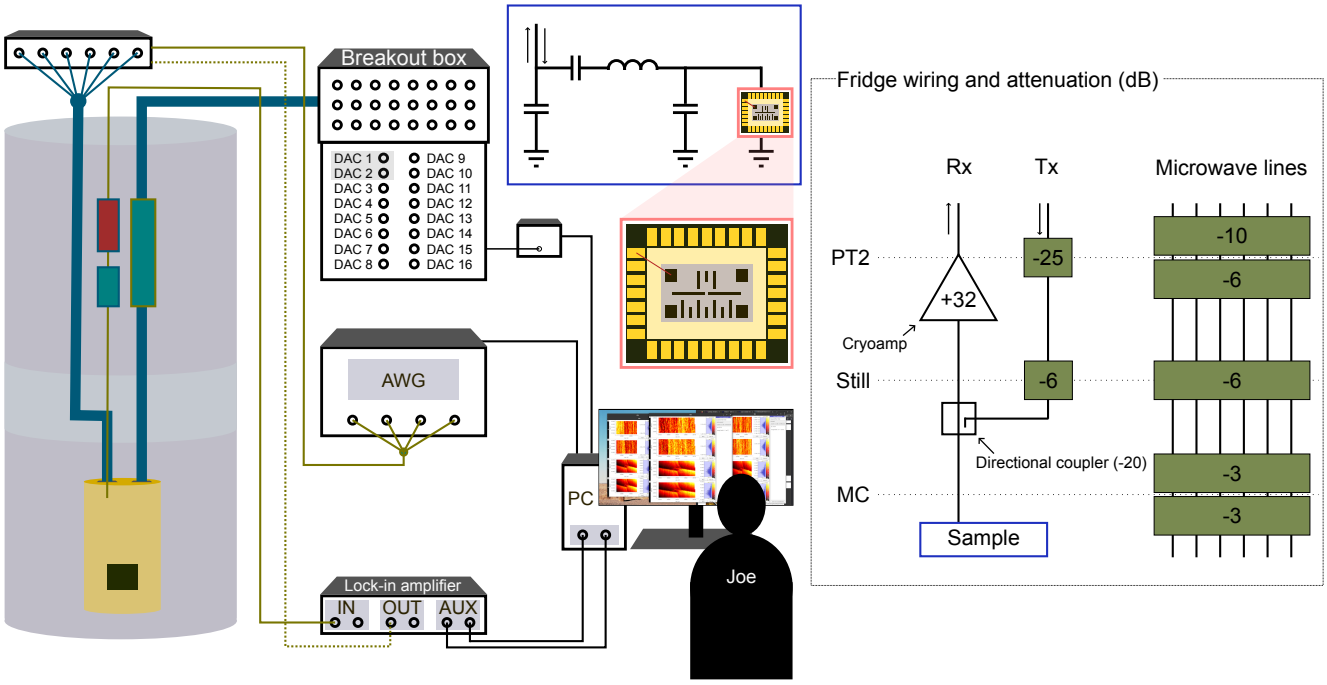


Figure 3.1.1: Cartoon of setup. The fridge, in grey, holds the sample in a puck (gold cylinder). Control lines are filtered depending on the frequency range they are required to transmit. A lock-in amplifier is connected to the sample via the reflectometry circuit, and demodulates the reflected signal. An example attenuation scheme for the high-frequency lines is given. The blue box contains a diagram of the reflectometry resonant circuit.

setup.

For the thermal energy of the electrons to be less than the dot charging energy, we need to cool the device to cryogenic temperatures, typically below 1 K. Most of the measurements in this thesis took place at a mixing chamber temperature around or below 50 mK. Achieving these temperatures requires the sample to be loaded in a dilution refrigerator, for which we use Oxford Instruments' Triton series.

Once the sample is loaded and cold, we need to control the sample and measure its state. Controlling the device consists of applying dc voltages to the metallic gates that shape the 2DEG/2DHG. From there, we apply ac voltages to some gates, and measure the state of the device using either dc transport or reflectometry (detailed in Section 3.2). We set these voltages through *control lines*, which are made up

of low-frequency lines (so-called *dc* lines) and high-frequency lines (so-called *fast* lines), with each type of line being appropriately filtered or attenuated to mitigate noise. For dc control, we use digital-to-analogue converters (DACs) to generate low-noise dc voltages between 0 and 4 V. Our DACs are the Delft-manufactured QT IVVI instruments. For spin manipulation and for fast measurement of the device charge state, we need ac control instruments, for which I primarily used the Quantum Machines OP-X and Tektronix AWG5204. These instruments allow arbitrary waveform generation with nanosecond-scale resolution.

To measure the state of our devices, we use radio-frequency (rf) reflectometry [90] via an rf lock-in amplifier. The high signal-to-noise ratio of rf reflectometry allows for microsecond-scale charge and spin state measurements. I detail some specifics in Section 3.2.

To control the hardware that sets and measures the device state, Barnaby van Straaten and I[1] wrote a piece of software unimaginatively titled `qgor`. The inspiration for the name, although it is hardly inspired, was the legacy control software `IGOR`. To fit in with the qool qrowd of quantum engineers, we replaced the first letter of the word with a `q`.

Some of the programs we wrote for `qgor` are used extensively throughout this thesis, so I detail them in Section 3.3. These tools are now extensively used by members of the group.

3.2 Reflectometry

Radio-frequency reflectometry is a measurement technique that allows for *fast readout*, namely measurements of transient events that cannot be observed in time-averaged

[1]With significant contributions from power users Sebastian Orbell and William Wombell, amongst others.

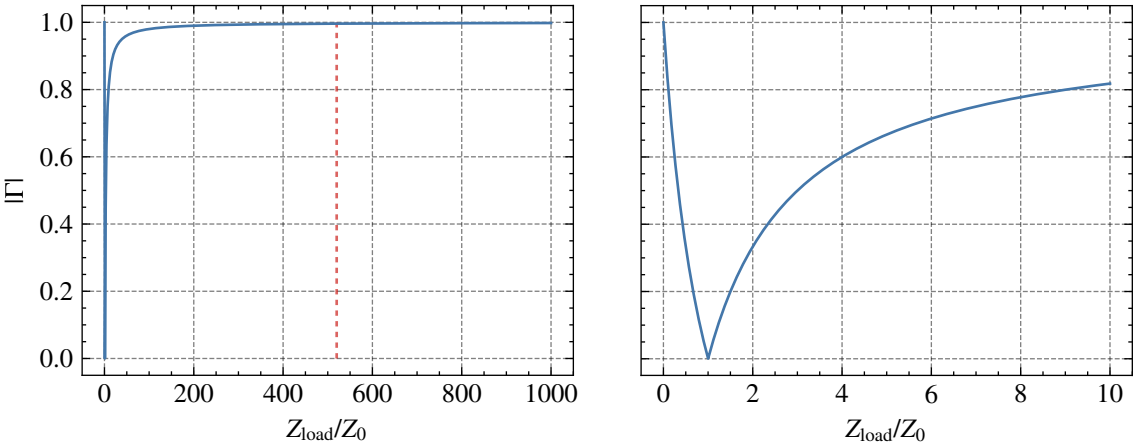


Figure 3.2.1: Reflection coefficient Γ as a function of Z_{load}. As Z_{load} increases beyond approximately ten times Z_0, the gradient of Γ as a function of Z_{load} drops significantly. The red dotted line on the left plot indicates the position of the resistance quantum $h/e^2 \approx 26 \times 10^3\,\Omega$.

measurements. Dc transport is not appropriate for such measurements due to the low bandwidth of the technique, which is typically limited by the low-pass RC filter created by the large input impedance of current amplifiers, a sample resistance in the kilo-ohm range, and the capacitance of the cables on the order of 1 nF. Reflectometry increases the bandwidth of quantum device measurements from the typical kilohertz range of dc transport measurements to the hundreds of megahertz range, or higher [90]. The high bandwidth of this measurement technique allows for rapid measurements of the charge and spin states of our devices, with noise suppressed by the use of radio-frequency lock-in amplification.

The operating principle of reflectometry is that an impedance mismatch in a transmission line will cause some of the incident signal to be reflected. Consider a load with impedance Z_{load} connected to a transmission line of impedance Z_0. The reflection coefficient, corresponding to the fraction of the signal power that is reflected at the impedance mismatch, is

$$\Gamma = \frac{Z_{\text{load}} - Z_0}{Z_{\text{load}} + Z_0}. \tag{3.2.1}$$

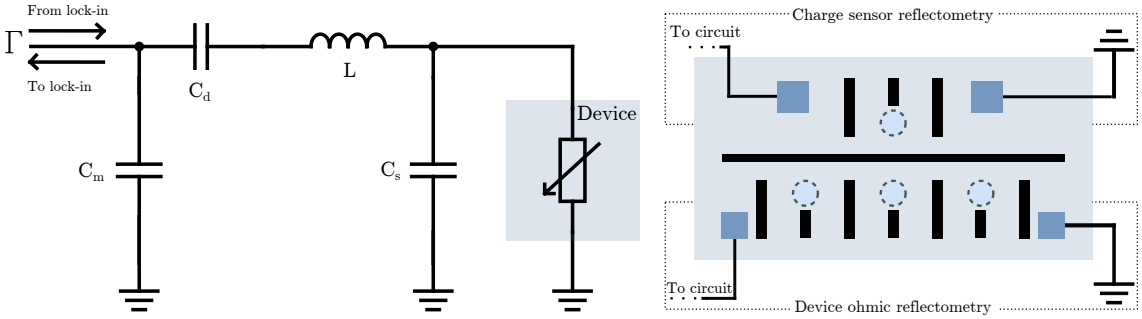


Figure 3.2.2: Reflectometry setup. Left: Schematic of the reflectometry circuit used at the Ares laboratory, Oxford. There are multiple of these circuits in parallel to allow for multiplexed readout. Right: Reflectometry configurations used in this thesis. Charge sensor reflectometry probes the impedance of a quantum dot charge sensor, which is capacitively coupled to the device array. Device ohmic reflectometry directly probes the impedance of the device array.

By measuring the returning signal's amplitude and phase after reflection from a load, we can record information about the load's impedance. If only life were that straightforward! Putting some numbers into Equation 3.2.1 shows that we need to modify this simple strategy before it can be useful for measuring quantum devices. Our Z_0 is fixed by the impedance of our transmission line, which is typically $50\,\Omega$. By comparison, a quantum dot device has an impedance on the order of the resistance quantum $Z_{\text{load}} \sim h/e^2 \approx 26\,\text{k}\Omega$ [90], or higher. This gives a ratio $Z_{\text{load}}/Z_0 \approx 520$, by which point the gradient of $|\Gamma|$ with respect to Z_{load} is almost zero (Fig. 3.2.1, left). Since we are trying to measure changes in the load's impedance, we need the absolute gradient of Γ to be as large as possible; the most sensitive region typically occurs in the regime $Z_{\text{load}} \approx Z_0$ (Fig 3.2.1, right). We therefore need to bring our load impedance to approximately $Z_{\text{load}} \approx 50\,\Omega$, which we do by means of an impedance matching circuit.

3.2.1 Impedance matching

To modify the impedance presented by a load, we can use a *matching circuit*. The matching circuit used in our laboratory, and therefore for the majority of the

measurements that follow, is schematically presented in Fig. 3.2.2. The circuit is an RLC resonator with an overall impedance of

$$Z_{\text{load}} = \frac{1}{i\omega C_d} + i\omega L + \frac{R_{\text{dev}}}{1 + i\omega C_s R_{\text{dev}}}, \tag{3.2.2}$$

where R_{dev} is the device resistance and the other parameters are as labelled in Fig. 3.2.2. This simplified model includes just the idealised versions of the circuit elements, removing for instance the capacitance and resistance of the inductor, and the parallel resistances of the capacitors. The exclusion of these parameters simplifies the analysis but does not significantly impact its conclusion.

The analysis in Appendix A shows that, close to resonance, the presented impedance of the matching circuit is

$$R_{\text{eff}} = \frac{L}{R_{\text{dev}} C_s}. \tag{3.2.3}$$

We use this formula to match the impedance of the circuit to the line impedance of $Z_0 = 50\,\Omega$. Considering Eq. (3.2.3), we can rearrange to show that the device resistance at which the matching circuit's impedance is $Z_0 = 50\,\Omega$ is

$$R_{\text{matching}} = \frac{L}{C_s Z_0}. \tag{3.2.4}$$

By using our knowledge of a device's resistance in a specific regime of interest, we can select the matching circuit's components such that the matching condition being met at that resistance, and therefore in the specific regime of interest. For the experiments in later chapters, I give the exact values; as a reference, we usually use inductors of around $2\,\mu\text{H}$, with $C_s \approx 0.5\,\text{pF}$ being provided entirely by the parasitic

capacitance of the device. The decoupling capacitor C_d (Fig. 3.2.2) is 92 pF.

Changes in the device's capacitance (modelled by C_s) and its resistance (R_{dev}) perturb the load's impedance from the approximate 50 Ω that is set by the choice of circuit parameters. As a result, we are in the regime where $|d\Gamma/dZ_{\text{load}}| > 0$ and we can obtain information about the device's resistance by measuring the change in reflected signal power and phase.

3.3 Measurements

Measurements of the charge state of a quantum dot are key to both the tuning procedure [4, 91] and qubit readout via spin-to-charge conversion [55, 92, 93, 94] Identifying the charge state directly, i.e. the number of charges on each dot, is not straightforward, because we only really know the number of charges on a quantum dot when we fully deplete it. In this thesis, I make extensive use of quantum dot charge sensors, probed via radio-frequency reflectometry.

3.3.1 Charge stability diagrams

Sweeping a dot's plunger gate voltage (gates V_2, V_4, Fig. 3.3.1) results in Coulomb peaks as the dot moves in and out of Coulomb blockade. On a single Coulomb peak, the number of charges on the dot fluctuates between n and $n + 1$. For a double quantum dot, sweeping both plunger gates results in a charge stability diagram, with geometric shapes representing voltage regions for which the charge states of the dots remains constant (Fig. 3.3.2 **a)**). Coulomb peaks along two different vectors within a two-dimensional measurement is an indicative feature of a double quantum dot.

There are several ways to probe the device to measure its state. In the following sections, I detail some of the most important techniques.

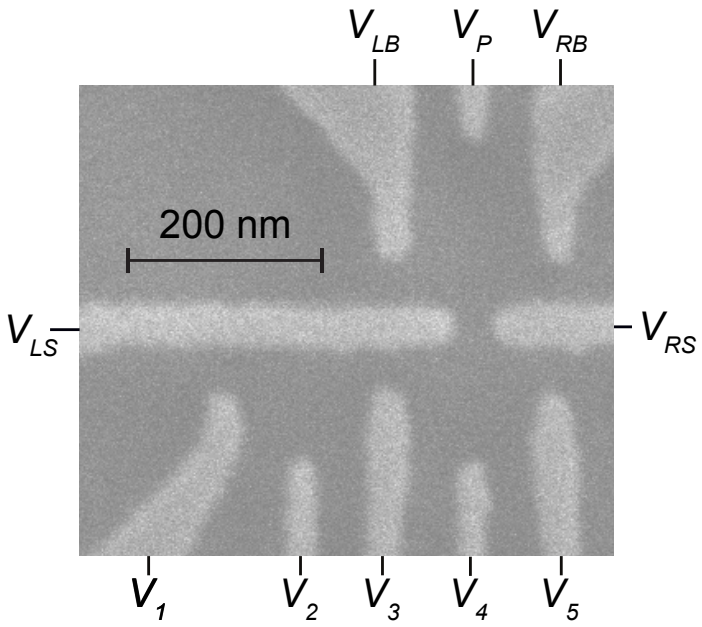


Figure 3.3.1: An example image of a quantum dot device, with a charge sensing dot on top, and a double dot array below. Odd-numbered device gates are barriers, while even-numbered device gates are plungers.

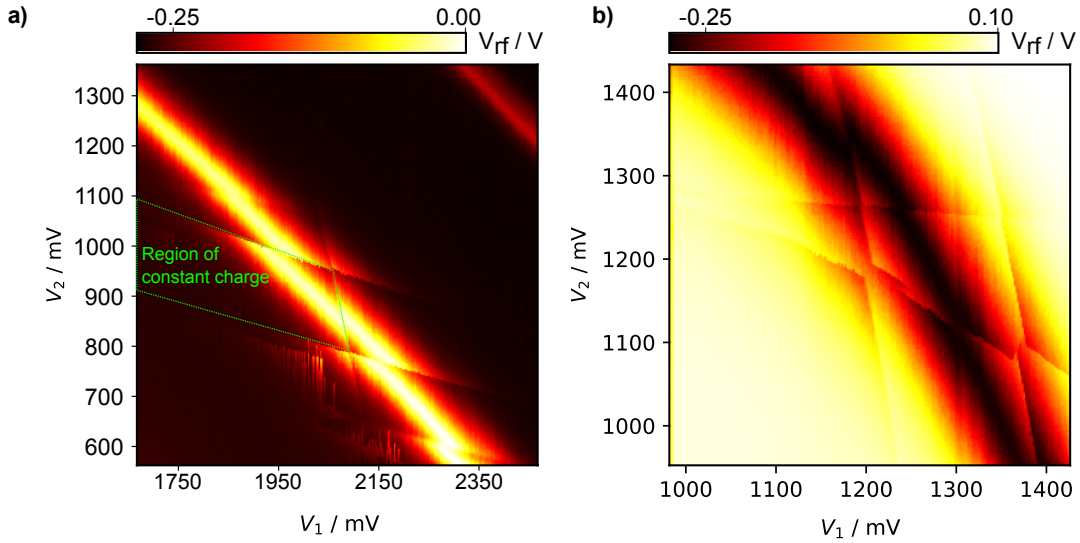


Figure 3.3.2: Some examples of stability diagrams, measured using a charge sensor. **a)** One region of constant charge is highlighted. Note also the large regions in which the sensor is not sensitive, resulting in the measurement containing purely noise (black regions). **b)** As in a). Note also the presence of a spurious third dot, visible as a horizontal charging line, suggesting the dot is located very close to gate V_2.

3.3.2 Transport

In transport measurements, we embed the device in a source measure unit. This instrument provides a voltage across the device and measures the resulting current through it. In Coulomb blockade, the current through the device is below the measurement noise floor due to the high presented impedance on the order of 100 MΩ. Within Coulomb peaks, when the dot occupation is fluctuating between n and $n+1$, we measure current. This technique forms the basis of most of the early literature in quantum dots [36], and is the most direct method for measuring changes in the device's charge state.

While transport measurements are relatively simple to set up and the results simple to interpret, the approach has limitations. The most pressing is aforementioned effective low-pass RC filter that blocks signal frequencies above a few kilohertz. Events happening faster than the millisecond timescale are therefore not directly measurable in a single pass using dc transport measurements.

Device ohmic reflectometry

In device ohmic reflectometry, we embed the ohmic contacts of the quantum dot array in the readout circuit (Fig. 3.2.2, bottom right). Rather than measuring the current through the device, we measure its impedance via the matching network as described in Section 3.2. This requires prior knowledge of the device's resistance in the regime of interest such that the presented impedance of the matching circuit is approximately $Z_{\text{load}} \approx Z_0$ in this regime (Eq. (3.2.4)). Radio-frequency reflectometry measurements have improved noise characteristics over dc transport due to a reduced contribution from $1/f$ noise [82], and the high bandwidth allows for increased measurement speed without loss of the high-frequency data components. Thus reflectometry measurements should allow us to measure, and therefore tune, our devices more

quickly; indeed, this is the subject of Chapter 4.

3.3.3 Charge sensing

Direct measurement of charge transitions via dc transport is limited by the fact that it requires a current to flow through the device. Charge rearrangements between dots do not result in a source-drain current, and therefore are not measurable directly [11]. The solution is to use a charge sensor, namely a device that is highly sensitive to changes in its electrostatic environment. The use of charge sensors has emerged widely in the literature, implemented via quantum point contacts [54, 58, 95, 96], proximal quantum dots [60, 76, 97, 98], and device gates [78, 79].

Quantum point contacts consist of a narrow, ballistic constriction of a conductance channel [99]. The conductance of this channel depends on its surrounding electrostatic potential, meaning it can be used to measure charge rearrangements in nearby quantum dots. They are (relatively) simple to fabricate and have a wide dynamic range, but trade this off for sensitivity. Quantum dot charge sensors, by comparison, have a charge sensitivity over an order of magnitude higher [77]; however, they have a narrower operating range, limited by the capacitive coupling between the gates used to operate the device and the measurement dot itself. This leads to significant regions of no sensitivity when measuring a device using a quantum dot charge sensor (Fig. 3.3.2).

For use as a charge sensor, a quantum dot is tuned to a regime where it is strongly capacitively coupled to the device array dots. When tuned to the side of a Coulomb peak, a quantum dot is in a regime of high transconductance, defined as $g_m = i_{\text{out}}/v_{\text{in}}$ for small signals. The operating principle is the same as in QPCs: small changes in its confining potential, as caused by charge rearrangements in nearby quantum dots, result in measurable changes to the dot's conductance, which is measurable via

radio-frequency reflectometry.

Quantum dot charge sensor reflectometry

In this configuration, a quantum dot charge sensor is embedded in the resonating impedance matching circuit via an ohmic contact (Fig. 3.2.2, top right). The sensing dot is biased to a regime of high transconductance (on the edge of a Coulomb peak), as in a transport-based sensing measurement. In the high-transconductance regime, we can identify device charge transitions in the output signal; however, this regime is lost when making changes to the voltages on any of the gates on the sample due to the capacitive coupling between these gates and the sensing dot (Fig. 3.3.2).

This approach has become ubiquitous for fast and sensitive measurements of quantum devices [77, 82, 100, 101, 102, 103]. It is also used for single-shot readout of spin qubits owing to its high sensitivity [104, 105]. As devices scale to multi-qubit architectures, the number of readout sensors per chip is likely to increase [1, 12], making their manual tuning and optimisation an increasingly time-consuming task. Multiple sensors may also be used in combination for improved measurement fidelity [59].

A benefit of charge sensor reflectometry is that we have control of the impedance of the sensor, and this impedance is decoupled from the operation of the device array. By changing the tunnel barriers of the sensing dot, we can change its impedance over several orders of magnitude, meaning we can measure the device state across its full operational range rather than only in the regime where the matching condition is met, as is the case in device ohmic reflectometry.

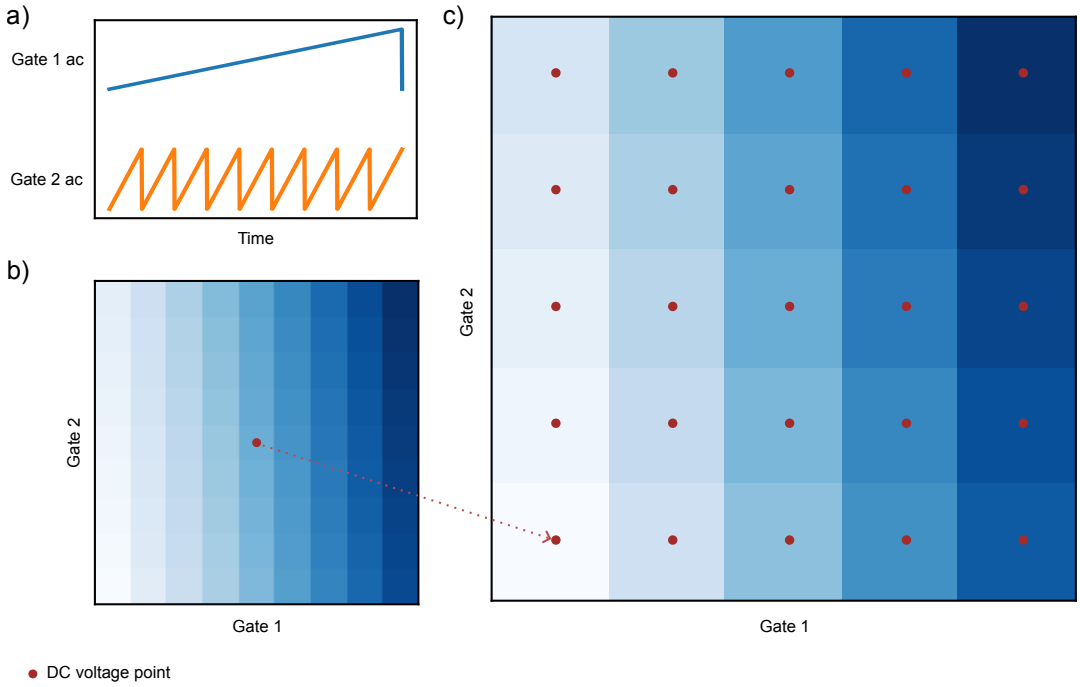


Figure 3.3.3: **a)** A cartoon of the voltage pulses applied by an arbitrary waveform generator to two device gates to map out a raster pattern that spans the charge stability of a double quantum dot. **b)** The pattern in a) maps out a raster pattern as shown. **c)** By stitching together the measurements in a)/b) whilst changing the underlying dc point, we can map out large stability diagrams of e.g. $1\,\mathrm{V} \times 1\,\mathrm{V}$ in times on the order of a second.

3.3.4 Fast measurements

Video-mode stability measurements

The wide bandwidth afforded by rf reflectometry means we can control the device with high-frequency control pulses and measure its response without losing information from the high-frequency components. One application of this is via so-called *video mode* measurements [91, 106], in which we map out charge stability diagrams by applying a voltage-space-filling scan (e.g. raster) over two device gates (typically plungers) using a high-frequency instrument such as an arbitrary waveform generator (AWG) (Fig. 3.3.3 **a)**). The speed of these measurements can reach well over 30 frames per second, allowing for real-time viewing of the device's charge transitions.

This is a highly entertaining experience.

Pulse-per-pixel measurements

The flexibility of AWGs means we can add further control pulses to the signals that map out the charge stability diagram, allowing us to apply a pulse to each point in the measurement. For instance, we can add a pulse that induces Pauli spin blockade (if the conditions are present) for every pixel in the stability diagram.

Large stability measurements with ac and dc

By taking video-mode type measurements at different gate voltage coordinates, we can stitch together large stability diagram measurements rapidly (Fig. 3.3.3 **c)**). This requires calibrating the amplitude of the ac signal relative to the dc voltage change, such that the dc jumps are equivalent to the amplitude of the ac pulse (next sub-section).

Attenuator compensation

To reduce the signal temperature of the ac signals, we use a series of attenuators at various stages of the fridge. The lower temperature reduces the effect of Johnson-Nyquist noise, but the attenuators also reduce the amplitude of the signal. To compensate for this effect, we calibrate the size of an ac pulse relative to a known dc voltage.

The procedure for this is as follows: we apply a square pulse of a known amplitude to a gate via the relevant high-frequency line. While this pulse is perturbing the voltage on that gate, we measure a stability diagram using two gates, one of which is the gate on we are applying the square pulse. We measure the dc voltage separation between transition lines, and compare this to the applied ac voltage out of the high-frequency instrument (e.g. AWG). The ratio of these voltages gives us the calibration scale,

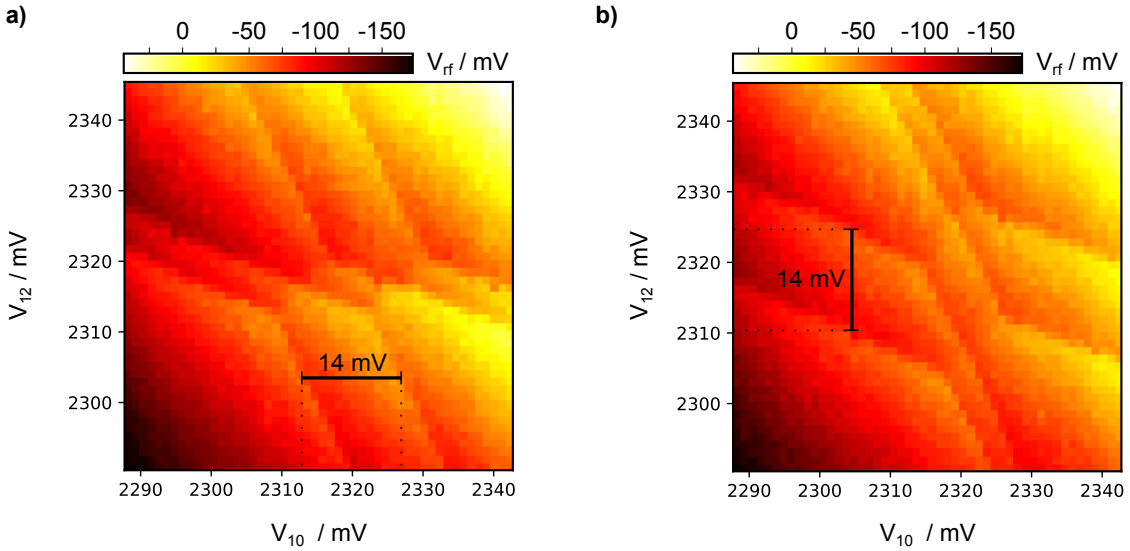


Figure 3.3.4: Stability diagram measured whilst applying a square pulse to **a)** V_{10} and **b)** V_{12}. By applying a square pulse of a known amplitude (here 230 mV) and measuring the distance between transitions in dc voltage space, we can obtain the conversion factor required to scale between the applied ac voltage and the dc equivalent at the device.

which corresponds to the attenuation due to the attenuators on the high-frequency lines.

Filter compensation

The bias tees used to combine the ac and dc signals act as a high-pass filter on the ac signal component, with a time constant of approximately 3.3 ms at room temperature. For a typical fast stability diagram measurement, the slow axis waveform has a periodicity of 10 ms; to apply the desired waveform to the device, we need to compensate the slow axis voltages (Fig. 3.3.5).

Passing a signal $s(t)$ through a first-order high-pass filter with time constant τ returns the filtered signal $s_{\text{filtered}}(t) = \mathcal{F}^{-1}[\mathcal{F}[s(t)](\omega) \cdot H_{HP}(\omega)]$, where $H_{HP}(\omega) = i\omega\tau/(1 + i\omega\tau)$, and $\mathcal{F}[\cdot]$ is the time- to frequency-domain Fourier transform. So that the waveform reaching the device gates is the intended $s(t)$, we apply the inverse function to the original waveform according to $s_{\text{comp}}(t) = \mathcal{F}^{-1}[\mathcal{F}[s(t)](\omega) \cdot H_{HP}^{-1}(\omega)]$.

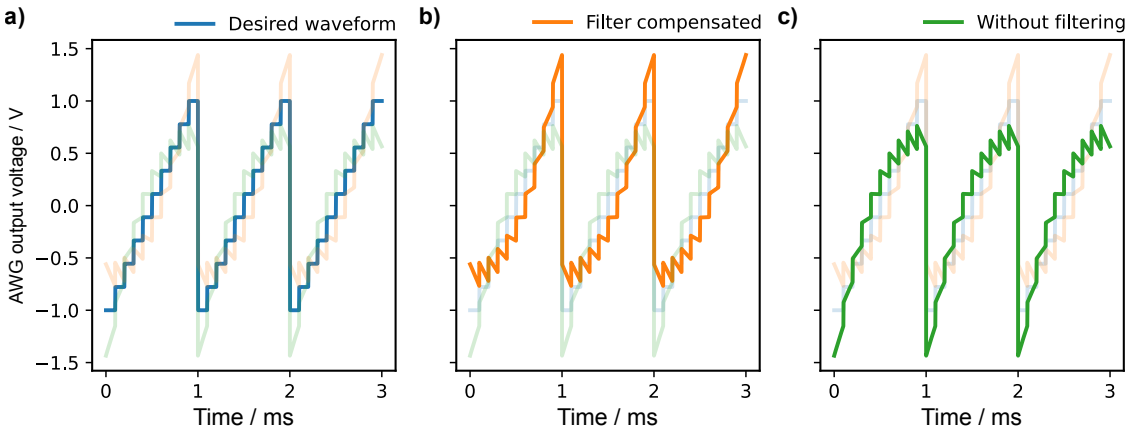


Figure 3.3.5: The effect of filter compensation for a filter with a 3 µs filter constant. **a)** The desired waveform, which is similar to those used in the video-mode measurements described in the main text. **b)** The pulse we send down to the fridge to recover the pulse in a) at the device. **c)** The pulse the device would see if we sent a) rather than b).

The resulting output from the AWG after filtering means the signal reaching the device gates is be exactly $s(t)$. A final consideration on this topic is that the high-pass filter removes any dc offset in a signal, and this cannot be compensated for by this filter compensation scheme.

3.3.5 Data acquisition

Lock-in detection

Lock-in amplifiers provide a means for measuring small changes in a signal's amplitude and phase, relative to a reference signal. They are particularly useful in noisy environments because they use homodyne detection to measure a frequency band around the reference frequency, removing all other (noisy) frequency components. In addition, the noise level at higher frequencies is usually smaller due to the $1/f$ noise spectral density, with the result that high-frequency measurements have a smaller noise contribution.

For the radio-frequency lock-in amplifier, we use a Zurich Instruments UHFLI ('ultra-high-frequency lock-in'). The reference signal is generated locally on the instrument and then split. This signal is sent through the reflectometry line in the fridge and enters the reflectometry circuit (Fig. 3.2.2). Here, the signal is reflected and returns to the lock-in where it undergoes homodyne demodulation with both the reference signal and a 90° phase-shifted version. The result is then low-pass filtered and the output is the in-phase (I) and out-of-phase/quadrature (Q) components of the demodulated signal.

Data acquisition card

To acquire the demodulated output of the lock-in amplifier, we connect its auxiliary outputs to an AlazarTech ATS9440 high-throughput digitiser. This card has four measurement channels, which typically we set up to measure device current, charge sensor current (if applicable), and the I and Q components of the integrated demodulated reflectometry signal.

3.4 Computational methods

3.4.1 Principal component analysis

The lock-in amplifier demodulates the returning rf measurement signal into its in-phase (I) and out-of-phase (Q) components. To extract the best signal-to-noise from this measurement, a typical approach is to tune the phase of the input signal such that the majority of the demodulated signal is returned in the in-phase component. This is the value that is then reported.

However, some signal remains in the out-of-phase component. We use principal component analysis (PCA) to optimally project the two-dimensional IQ data onto a

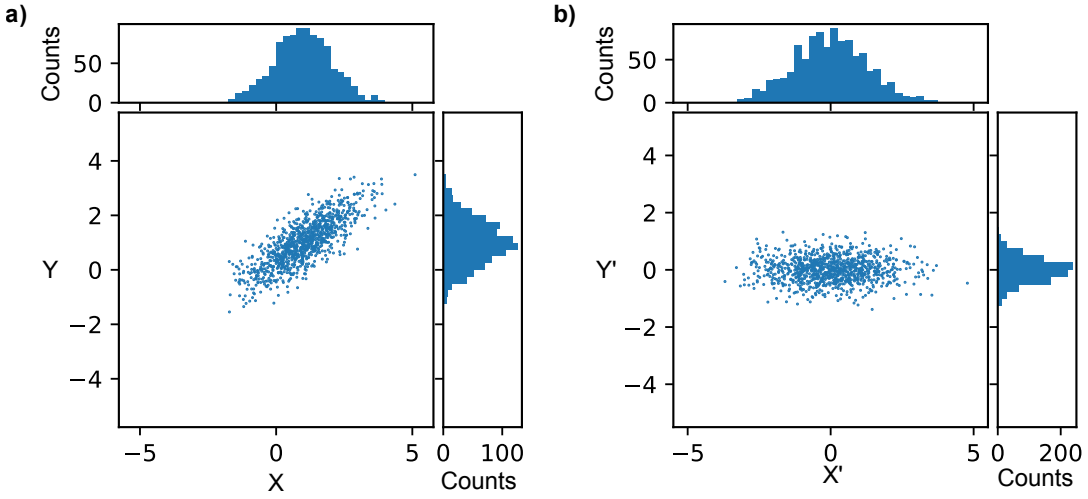


Figure 3.4.1: Principal component analysis. In **a)**, 1000 points from a random bivariate Gaussian are shown. Note the distribution is fairly evenly split between the two axes as indicated by the width of the histograms, meaning if we just choose to measure one of these axes, we lose the variance in the other. In **b)**, after PCA is applied, the data has zero mean and the variance is optimally projected onto one axis (the X' axis). The variance in the Y' axis is, in experimental settings, primarily noise.

single dimension with maximum variance.

Mathematical formalism of PCA

Given a dataset with two feature dimensions, x and y, we begin by constructing the covariance matrix,

$$S = \begin{bmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y,y) \end{bmatrix} \tag{3.4.1}$$

where $\text{cov}(x,y)$ represents the covariance of x and y and $\text{cov}(x,x) = \text{var}(x)$.

The covariance between variables x and y is defined as

$$\text{cov}(x,y) = \mathbb{E}[(x-\mu_x)(y-\mu_y)] \tag{3.4.2}$$

where $\mathbb{E}$ is the expected value and μ_x, μ_y are the means of x and y, respectively.

Covariance matrix eigendecomposition

The eigenvectors and corresponding eigenvalues of the covariance matrix S represent the distribution of the dataset's variance in a new orthogonal basis composed of the eigenvectors. Each eigenvalue represents the variance of the data when projected onto its corresponding eigenvector.

The eigenvector with the largest eigenvalue, $\vec{v}_1$, is known as the first principal component. We project our data onto $\vec{v}_1$ to reduce the dimensionality of the data whilst capturing the majority of the variance (and therefore information) within it (Fig. 3.4.1). This projection is given by

$$\vec{z} = \vec{v}_1^T \vec{X}, \tag{3.4.3}$$

where $\vec{X}$ is the data vector containing x and y and $\vec{z}$ is the projected value.

For our measurements, the x and y dimensions are represented by the I (in-phase) and Q (out-of-phase) components of the reflected rf signal. The corresponding eigenvector is known as the first principal component of the data.

The eigenvector is a vector in IQ space with an equation $y = mx + c$. We subtract the mean from the data so the best fit line, which is exactly equivalent to the eigenvector with the highest corresponding eigenvalue, will also have zero mean in both dimensions, meaning its equation simplifies to $y = mx$. The gradient is therefore $m = y/x$, representing an angle of $\arctan y/x$. The phase of a data point in IQ space is also $\phi = \arctan \frac{y}{x}$[90], so the angle of the best fit line will be at the same angle as that represented by the phase. Projecting the data onto the first principal component is therefore identical to optimally tuning the phase for a given

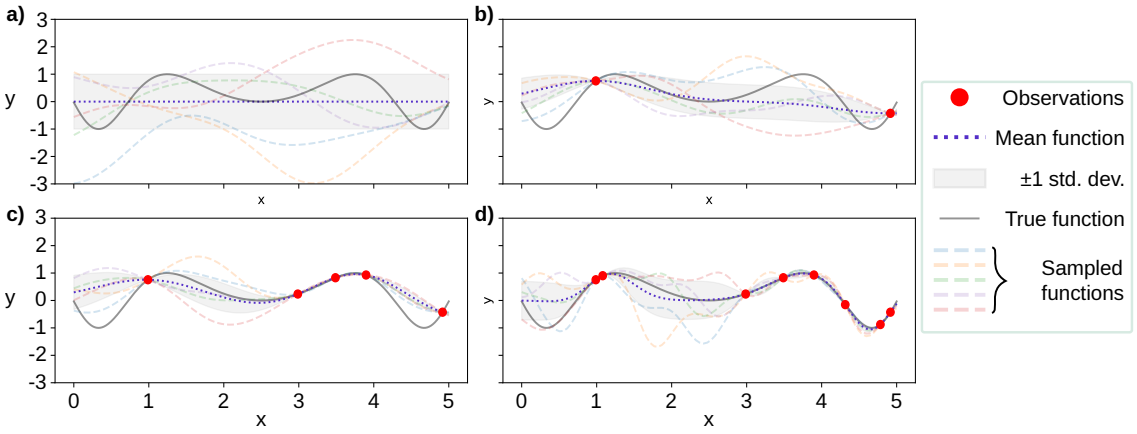


Figure 3.4.2: A Gaussian process learning the underlying ground truth with increasing accuracy as it is conditioned **a)** zero, **b)** two, **c)** five, and **d)** eight observations. As more observations are added, the mean function models the true function with increasing accuracy. Example functions from the Gaussian process are shown.

radio-frequency measurement.

3.4.2 Gaussian processes

Gaussian processes (GPs) are a widely-used method for predicting the functional relationship between observed data points [107]. They are of particular use due to their inherent modelling of the degree of uncertainty in their predictions. For a set of observations, a Gaussian process defines a distribution over functions that could have generated the observations.

Mathematical formalism of GPs

A Gaussian process is a collection of random variables, with any finite number of these variables having a joint Gaussian distribution. A GP can be completely defined by its mean function $\mu(x)$ and a covariance function, also known as a kernel, $k(x, x')$. Formally, a GP is written as

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')), \tag{3.4.4}$$

with the covariance function $k(x, x')$ defined as $\mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$, as per Equation (3.4.2). This covariance function represents the degree of similarity between points x and x'.

To begin with, it is typical to set the mean function to equal zero, representing the fact that we do not have any prior beliefs about the modelled function. If this is not the case, whatever prior knowledge we have available can be used generate a prior mean function. The kernel defines the covariance between any pair of values x and x'. Based on the expected behaviour of the modelled function, we can select a kernel function that will appropriately match our data. A widely-used kernel is the radial basis function, defined as

$$k(x, x') = \sigma_f^2 \exp(-\frac{(x - x')^2}{2l^2}), \tag{3.4.5}$$

where σ_f^2 is the variance parameter and l is the length-scale parameter. These hyperparameters undergo their own optimisation routines such that they best explain the data, leading to more accurate modelling.

If there is strong covariance between x and x', $f(x) \approx f(x')$. By using kernels that return outputs close to 1 for input values that are close together, the output of the GP will be smooth.

The mean function is the mean of the possible functions that generate the observations. At input values where we have an observation, the uncertainty is low (equivalent to the uncertainty we have in our measurement process) and the mean is well-defined. At input values close to those where we have an observation, the uncertainty increases due to the fact that we no longer know the exact output value for that input.

Posterior distribution

The use of a Gaussian process comes from its ability to predict the value of a function for an input that has not been measured. For a set of training data $(\mathbf{X}_1, \mathbf{y}_1)$, we want to predict the output value $\mathbf{y}_2$ at a new input value $\mathbf{X}_2$ which is not within the training set. We do this by computing the posterior distribution $p(\mathbf{y}_2|\mathbf{y}_1, \mathbf{X}_1, \mathbf{X}_2)$. Due to the formulation of the Gaussian process, $\mathbf{y}_1$ and $\mathbf{y}_2$ are jointly Gaussian and their joint distribution is given by

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}_1, \mathbf{X}_1) + \sigma_n^2 I & K(\mathbf{X}_1, \mathbf{X}_2) \\ K(\mathbf{X}_2, \mathbf{X}_1) & K(\mathbf{X}_2, \mathbf{X}_2) \end{bmatrix} \right) \tag{3.4.6}$$

,

where $\mu_1 = \bar{\mathbf{X}}_1$, $\mu_2 = \bar{\mathbf{X}}_2$, and σ_n^2 is the noise variance, representing the noise inherent in the measurement process.

In practice, Gaussian processes are used to infer the underlying function that generated observed data points. As additional observations are added, the GP learns the underlying function with increasingly greater accuracy (Fig. 3.4.2), and the uncertainty in predicted values reduces. Gaussian processes are useful for black-box modelling of functions because they can take on many forms. Knowledge of the approximate forms of the ground truth function can help to create a more useful kernel for the model. For example, if modelling periodic data, a kernel that models this periodicity will help the GP fit the observations more accurately [108].

3.4.3 Bayesian optimisation with Gaussian processes

If we are interested in optimising a function (the *objective function*), the most naive approach, known as a brute-force approach, is to measure the value of that function for all input values. This guarantees that we will find the true optimum. However,

sometimes the function we wish to optimise is slow or expensive to compute. In this case, we can model the objective function using Gaussian processes (GPs), and use Bayesian optimisation to efficiently find the optimum.

Bayesian optimisation is an efficient method for optimising objective functions that are noisy and expensive to sample. Under this scheme, we build a surrogate of the objective function using a Gaussian process and uses this model to make decisions about which input values to sample next. To do this, we create an *acquisition function*, which makes use of the probabilistic distribution of the GP to balance between exploration of the objective function and exploitation of our knowledge of it. Explorative measurements are those that sample the function where the uncertainty in its real value is high; exploitative measurements are those that sample the function where the predicted mean function is high. The maximum of the acquisition function at a given iteration gives the next point to sample.

A typical acquisition function is the upper confidence bound, and this function neatly and transparently demonstrates the concept of exploration-exploitation trade-off. This function is defined to be

$$\alpha_{\mathrm{UCB}}(x) = \mu(x) + \kappa\sigma(x), \tag{3.4.7}$$

where $\mu(x)$ is the predicted mean of the objective function at input value x, $\sigma(x)$ is the predicted standard deviation at x, and κ is a hyperparameter that sets the weighting of this trade-off. If κ is large, the uncertainty term is more heavily weighted, so the maximum value of the acquisition function (and therefore the next input value to be sampled by the optimiser) occurs at a point of high uncertainty. Towards the end of an optimisation run, when we are more confident in our knowledge of the objective function, the value of κ is reduced to lean towards the exploitation of this

knowledge. The maximum of the acquisition function in the low-κ regime will occur where the mean function of the surrogate is high.

The Bayesian optimisation routine is as follows:

1. **Initialisation**: We begin with a prior distribution over the objective function with specified mean and kernel functions.
2. **Evaluate initial points**: Next, we sample the objective function at a set of initial values.
3. **Condition GP on initial values**: We use the observations made in Step 2 to update the posterior distribution of the GP.
4. **Optimise acquisition function**: We compute the acquisition function for all possible input values within our domain using the GP's mean and covariance. The maximum of this function gives us the input value to sample next in the objective function.
5. **Sample the objective**: We evaluate the objective function at the input generated by Step 4.
6. **Iterate**: We use the measurement from Step 5 to update the posterior distribution of the GP, and iterate this sample and update until we meet a predetermined stopping criterion, such as convergence of the acquisition function or a maximum number of function evaluations.

Bayesian optimisation failure modes

Bayesian optimisation is a robust optimisation approach, but there are some potential failure modes that must be considered to ensure optimal performance. As with all optimisation routines, Bayesian optimisation can suffer from premature convergence to local optima, which can occur when the balance between exploration and exploitation

is not managed correctly. Incorrect hyperparameters within the Gaussian process will also lead to poor performance [109]. For instance, if the variance within the GP is too low, the value of exploration will be diminished, which may lead to convergence on local optima. Equally if the length-scale parameter of the GP is incorrect, this will misguide the acquisition function. A large length scale will generate low returns for exploring input values that are close to inputs that have been measured.

Chapter 4

Tuning a double quantum dot using radio-frequency reflectometry

The work in this chapter was originally published in Ref. [91]. The work I present in this chapter was done in collaboration with Barnaby van Straaten, with assistance from group post docs Florian Vigneau and Federico Fedele.

The first stage of operating a spin qubit is to tune the device to a quantum dot regime. The exact nature of this regime depends on the type of qubit you are after, but several implementations can be hosted in a double quantum dot, making it a natural unit cell.

In this chapter, I present an algorithm that automatically tunes a double quantum dot by leveraging high-speed radio-frequency (rf) measurements. We use the sheer speed of reflectometry to generate significant amounts of data, and pair this with a score function based on the Fourier transform to identify the periodicity of charge loading events that signifies the presence of a double quantum dot.

Problem statement

Tuning procedures for quantum dots in heterostructure devices are typically based around transport measurements. However, transport is slow and has a limited bandwidth, creating a bottleneck in the tuning process. A tuning algorithm based around reflectometry measurements, which are typically set up in qubit experiments anyway, would bypass this bottleneck without requiring additional experimental setup or hardware.

4.1 Introduction

Many groups now use rf reflectometry for high-fidelity, single-shot spin readout [104, 105, 110, 111, 112]. Reflectometry is the final measurement protocol of spin qubit experiments; however, until the work presented in this chapter, groups were using transport (dc) measurements for the automated tuning of their devices [4, 66, 69, 87]. Transport-based tuning solutions are not compatible with many propositions for scalable device architectures [32, 113], and are slowed by the RC filter formed by the transport measurement setup. The small number of existing algorithms for the automatic coarse tuning of a quantum dot suffered from various problems. The earliest, Ref. [66], relied on fitting a tetragon to current measurements. This is slow and has poor noise resistance. In Ref. [86], Darulová *et al.* automatically characterise and tune several quantum dot devices, but do report or show the quality of these double dots nor the tuning time. They also rely on knowledge about the device gained from previous manual experience.

Our algorithm is demonstrated on a relatively simple device that could, in principle, be tuned using transport measurements. The power of the algorithm lies not in tuning such a device, but in tuning those in which their architecture prevents transport-based tuning, and is therefore a key step in the path to large arrays of automatically-tuned

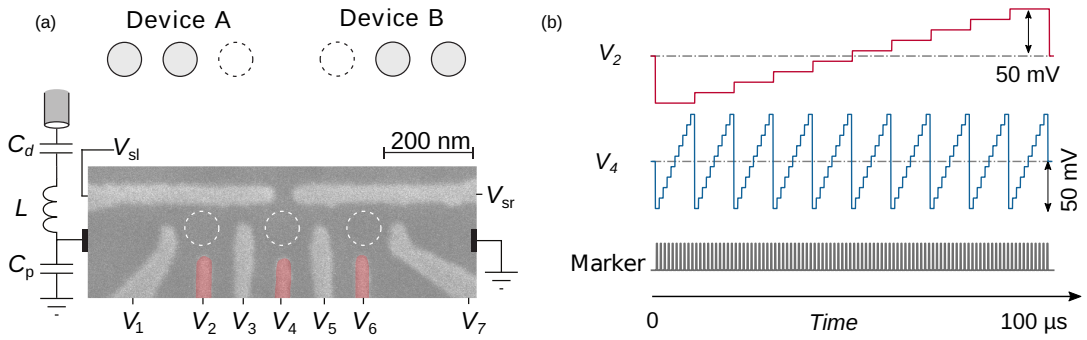


Figure 4.1.1: Device and measurements. **a)** Microscopy image of a device nominally identical to the one used in this chapter. The gates highlighted in pink have bias tees connected to allow for ac and dc voltages to be summed. We operate the triple dot array as two double-dot arrays as labelled. **b)** Voltages sent from the AWG to device A form a raster pattern that maps out the charge stability diagram. For device B, we sent the signal labelled for V_2 to V_6.

spin qubits.

The algorithm is built around two-dimensional charge stability measurements which are acquired using rf reflectometry over a millisecond timescale. We calculate a score for each of these measurements, looking for periodicity in both axes, which is representative of a double dot due to the presence of Coulomb peaks. We optimise this score function over the device gate voltages, searching for configurations that lead to strong double dot features. By using a Gaussian process to model the location of features of interest, we reduce the amount of time spent measuring in noisy, featureless regions and thereby speed up the search process.

4.1.1 Experiment

Device

We demonstrate the algorithm in a Ge/SiGe depletion-mode device (Fig. 4.1.1 **a)**) similar to that described in Ref. [47]. The measurements were performed at a mixing chamber temperature of approximately 50 mK.

The architecture of the device allows for up to three device dots to be formed and controlled. A pair of splitter gates, V_{sl} and V_{sr}, separated the device array from gates that could have been used to create a sensing dot. These splitter gates were set to a constant voltage of 2 V throughout the experiment. From here, you may forget about the sensor dot gate architecture and these splitter gates.

Seven device gates, V_1 to V_7, were used to control the array. We operated the array as two independent double dots, labelled A (left) and B (right) (Fig. 4.1.1 **a)**). Device A was controlled by gates V_1 to V_5, with device B being controlled by gates V_3 to V_7. Bias tees were connected to gates V_2, V_4 and V_6 (*plunger gates*) to allow the summation of ac signals with the dc bias points.

Ohmic radio-frequency reflectometry

One of the ohmic contacts of the device was embedded in a matching network, the impedance of which was probed using rf reflectometry. We used an L-matching network with a 92 pF decoupling capacitor (C_d) and a 2.2 µH inductor (L). The parasitic capacitances of the device and sample board's printed circuit are labelled C_p.

The reflectometry circuit was probed with an ac tone at close to the resonant frequency, at the point of maximum contrast between the open device and closed device. We obtained the maximum contrast by measuring the demodulated rf signal as a function of frequency with the device in the open state, and then in the closed state. By selecting the frequency at which the absolute difference in the signal between these states was maximised, we obtained the highest contrast. The power of the rf tone was fixed at −90 dBm at the sample after various attenuation stages to reduce the signal noise temperature.

Measurements

We acquired the algorithm's core two-dimensional charge stability measurements by applying a raster pattern to the plunger gates of each device (gates V_2 and V_4 for device A; gates V_4 and V_6 for device B) using an arbitrary waveform generator (AWG) (Fig. 4.1.1 **b)**). After filtering and attenuation, the amplitude of the raster scans equated to approximately $100\,\mathrm{mV} \times 100\,\mathrm{mV}$ at the device. The raster pattern of each axis is composed of a series of summed step functions, with plateaus at regular intervals corresponding to the measurement points. We used scans with a resolution of 100 by 100 pixels. Each pixel was integrated for a total measurement time of $1\,\mu\mathrm{s}$, meaning a full $100\,\mathrm{mV} \times 100\,\mathrm{mV}$ charge stability diagram took approximately $10\,\mathrm{ms}$ to measure.

Filter compensation

We compensate the waveforms for the effects of the high-pass filters due to the bias tees that combine the dc and ac signals as outlined in Section 3.3.4.

Principal component analysis

The demodulated in-phase and out-of-phase (I and Q) signals were combined and optimally projected into a single dimension using principal component analysis (PCA) as detailed in Section 3.4.1. We processed the measurements taken in the exploratory ramps using PCA, but those scored by the Fourier-based score function did not need this processing because the score function is able to jointly evaluate both components of the demodulated signal.

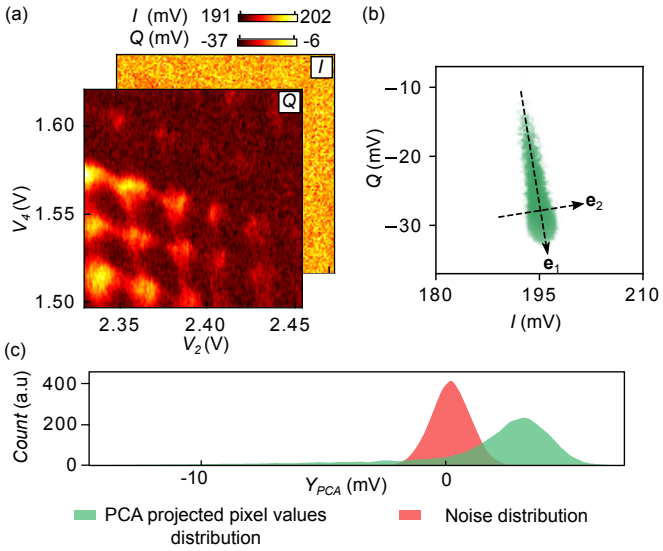


Figure 4.1.2: **a)** The I and Q components of the demodulated signal, showing most of the data in the Q component. **b)** A scatter plot of the data from a) show that most of the variance is over the Q (y) axis, but there is some variance in the I (x) axis. Using PCA, we project the data onto a line of best fit that captures the maximum possible variance in a single dimension ($\boldsymbol{e}_1$). **c)** Histograms of the noise distribution, collected as described in Section 4.3.3, and the PCA-projected data from b).

4.2 The rf hypervolume

With low gate voltages, the conduction channel of the device is populated with charge carriers, so the impedance is relatively low. This is known as the *open* regime. When the gate voltages are high, the conduction channel is pinched off and the impedance is high enough that any current is below the measurement noise floor. This is known as the *closed* regime.

In transport measurements, we can consider a hypersurface that separates the open and closed regimes. In a single gate voltage sweep, this surface manifests as the sharp drop in current as the device enters the pinch-off regime. Scaling this up to two gates with no crosstalk, we would see a rectangular hypersurface, inside which the device is in the open state and outside which the device is closed. Introducing crosstalk skews the shape of the hypersurface, but the concept remains. It is possible

in this way to picture a high-dimensional hypersurface representing the transition between an open and closed device. In our quantum devices, the features of interest (quantum dots) lie near this transport hypersurface. This concept forms the basis of transport-based tuning algorithms [4].

Transitioning to reflectometry-based auto-tuning introduces an additional complication which means the hypersurface idea needs adjustment. Equation 3.2.4 shows us that there is a combination of circuit parameters that brings the overall readout circuit to an impedance of $Z_0 = 50\,\Omega$; at this point, the circuit may be said to have *met the matching condition*. We can consider the matching condition as a window; within this window, the circuit impedance is approximately $50\,\Omega$, and we can measure information about the sample's impedance. Outside this window, the circuit impedance is not close to $50\,\Omega$, and the gradient of the reflected power (or phase) with respect to the device resistance drops to approximately zero. Here, we cannot measure the device state at all, because changes in its resistance do not change the demodulated reflectometry voltage. For this experiment, the circuit parameters are chosen such that the device resistance at which the matching condition is met occurs at the few-hole double-dot regime.

4.3 Algorithm

The idea of a quantum dot tuning algorithm is simple: change the gate voltages of your device, measure its state, and make a decision as to whether that state is what you want. If it is what you want, you can terminate there. If it is not, you must continue searching.

There are therefore three components:

1. a strategy to navigate through device voltage space;

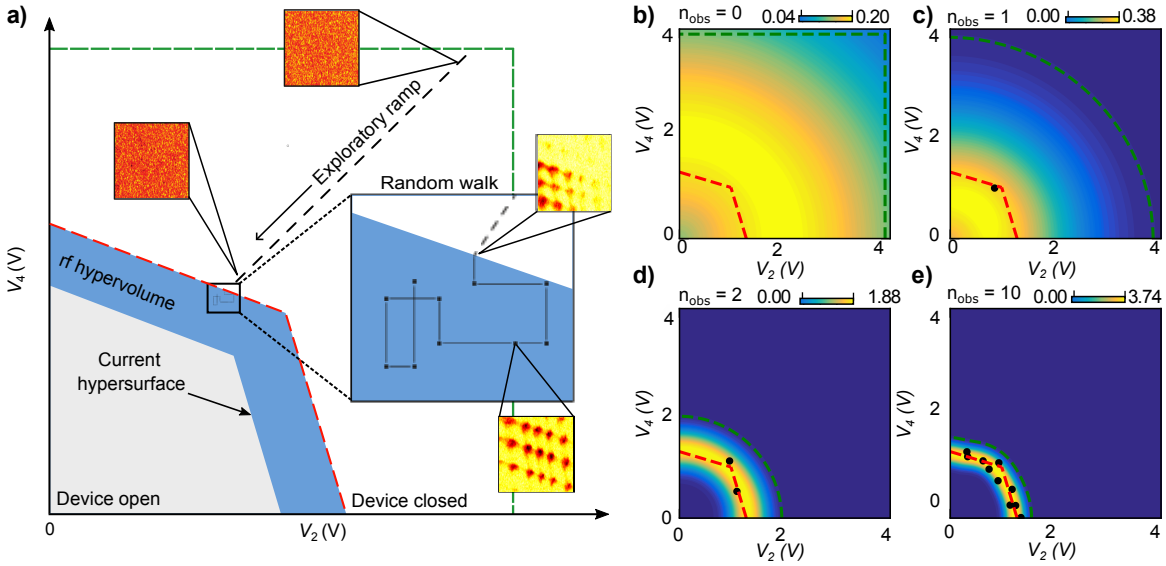


Figure 4.3.1: **a)** A pictorial representation of the algorithm. The first exploratory gate ramps starts at the green dashed line, equal to 4 V on all gates, and moves inwards towards the origin. Along this ramp, we check whether the measurements contain noise or pixel values that suggest we have hit the outer edge of the rf hypervolume (red dashed line). Within the hypervolume, we perform a random walk, scoring the features with the score function. **b) - e)** As measurements identify the location of the rf hypervolume, the Gaussian process model of its surface is updated. Subsequent ramps' starting locations are chosen to be somewhere on the manifold representing three standard deviations away from the Gaussian process mean function.

2. a measurement protocol;

3. a scoring function for the results of your measurements.

4.3.1 Overview

In this subsection, I give a very brief overview of the algorithm. The details are minimal but are given in the following sections. A cartoon of the algorithm is given in Fig. 4.3.1, and a flowchart is given in Fig. 4.3.2.

We begin with the gates at a location determined by the Gaussian process (Section 4.3.2), designed so that this is outside (i.e. at larger gate voltages than) the rf hypervolume. We then generate a vector that gives us a direction in which to ramp the gate voltages (towards $\vec{V}_{\text{gates}} = \vec{0}$). As we step along this exploratory ramp,

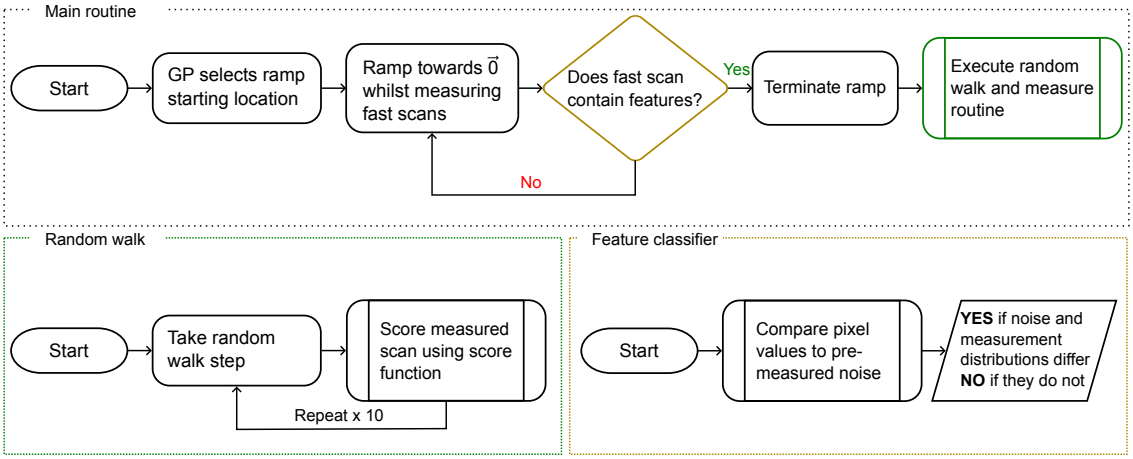


Figure 4.3.2: A flowchart representing the key steps of the algorithm.

we take two-dimensional measurements (as per Section 4.1.1) at regular intervals of 50 mV. We feed each measurement into a classifier that determines whether we are measuring noise or device features. If we measure noise, we continue along the ramp; if the classifier determines that we are measuring features, the ramp is terminated here, and we know that the ramp has reached the rf hypervolume. Once we are within the rf hypervolume, we perform a random walk through the device's gate voltage space, searching for quantum dot features. Each of the measurements taken along this random walk are then scored by a score function that evaluates the likelihood that 'good' double dot features are present in the data.

We use a few tricks from the machine learning toolbox to improve the naive algorithm. Firstly, the in-phase and out-of-phase components of the data measured in the 2D measurements are optimally combined using principal component analysis (Section 3.4.1). To guide the selection of vectors for the exploratory ramps, we model the position of the rf hypervolume with a Gaussian process (Section 4.3.2). I detail the implementation of the noise/feature classifier in Section 4.3.3, and that of the score function in Section 4.3.4.

4.3.2 Gaussian process

I introduced Gaussian processes in Section 3.4.2. Here, I will detail how we used them to improve the efficiency of this algorithm.

If we knew exactly where the rf hypervolume was in device space, the exploration section of this algorithm would be quite straightforward. As it happens, we do not know exactly where it is to start with due to device variability, but we can classify a given measurement as having been taken inside it using the feature/noise classifier (Section 4.3.3). Our exploratory ramps terminate only when they reach the hypervolume's surface. We use the voltage coordinates of these termination points to update a Gaussian process that models the outer surface of the rf hypervolume (Fig. 4.3.1).

We use the Gaussian process model of this surface to inform the starting positions of subsequent exploratory ramps. We start the exploratory ramps on a voltage surface equal to three standard deviations outside the mean function of the Gaussian process, thereby reducing the time spent moving through gate voltage space in which there are no device features. The Gaussian process begins with a Gaussian centred at 2 V, with a standard deviation of 2/3 V such that three standard deviations is equal to 2 V, meaning the ramp initialisation surface is equal to 4 V for all gates. As measurements are made, we update the Gaussian process with the known voltage locations of the rf hypersurface. These measurements confirm its location with a low uncertainty. Our uncertainty in the position of the rf hypersurface between these voltage locations, at unsampled gate voltage coordinates, is modelled by the Gaussian process, along with the most probable location of the hypersurface itself (given by the Gaussian process's mean function). Starting our exploratory ramps from three standard deviations away from the mean function theoretically results in 99.7% of our measurements starting outside the rf hypervolume.

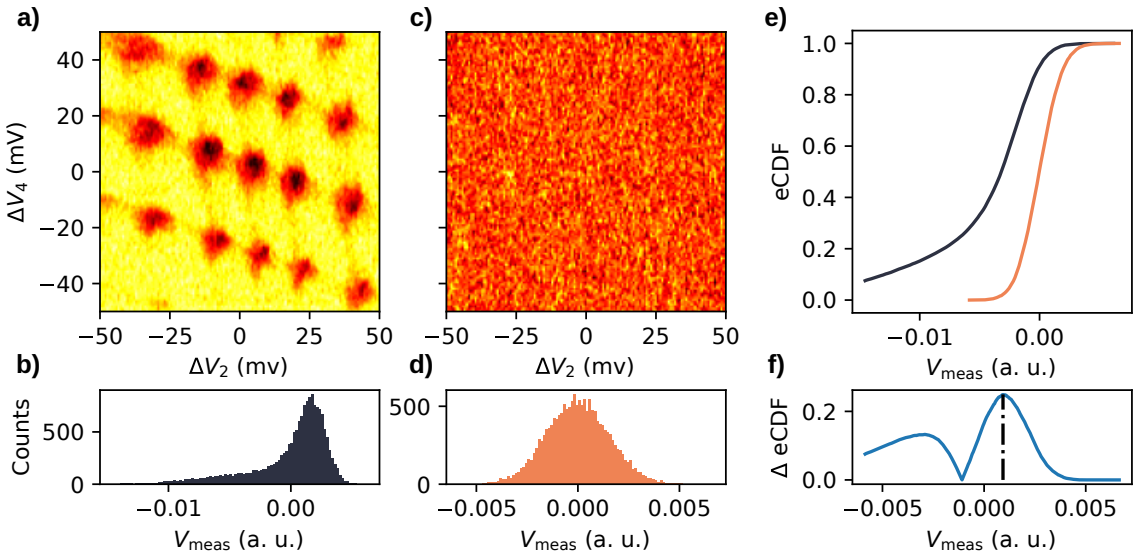


Figure 4.3.3: **a)** An example measurement of device features, with **b)** the associated histogram. Likewise **c)** is a measurement containing just noise in the pinch-off region, with **d)** its associated pixel value histogram. **e)** The empirical cumulative distribution function for the feature measurement (black) and noise measurement (orange). **f)** The difference in the eCDF, the largest value of which is the Kolmogorov-Smirnov statistic. This statistic is used to determine whether the distributions are different.

4.3.3 Classifying features against noise

Outside the rf hypervolume, the 2D measurements contain only noise (Fig. 4.3.3 **c)**). Inside the rf hypervolume, due to the parameters of the readout circuit, we measure device features of interest (Fig. 4.3.3 **a)**). Only when we are in the rf hypervolume is it necessary to check the score of the measurement; scoring every measurement, including noise, could lead to spurious scores and would certainly slow down the algorithm.

To determine whether a given measurement was taken inside the rf hypervolume, we compare the distribution of data in the measurement to a pre-measured noise distribution (Fig. 4.3.3 **e)**, **f)**). If the measurement data sample is statistically different to this noise distribution, we classify the measurement location as being inside the hypervolume.

To generate the noise benchmark distribution, we set all gate voltages to their maximum values (4 V, limited by the range of the DACs). In this voltage configuration, the device is definitely in pinch-off, meaning we are not measuring device features. We took ten two-dimensional scans of the type described in Section 4.1.1 and combined the data into a single sample of size $N_{\text{measurements}} * \text{pixels}_x * \text{pixels}_y = 10 * 100 * 100 = 100,000$.

We classify a measurement as being noise or not by comparing its pixel value distribution to the noise benchmark distribution using the two-sample Kolmogorov-Smirnov (K-S) test [114] (Fig. 4.3.3 **f)**). The K-S test determines the likelihood that two samples are drawn from the same distribution. By comparing the distribution of pixels in a measurement to the noise benchmark distribution, we can measure the likelihood that the measurement contains device features rather than noise.

We use a critical p-value of 10^{-3}. This means, on average, we falsely reject the null hypothesis (a false positive; here, the classification of noise as signal) one time for every 1,000 scans.

4.3.4 Double-dot feature score function

Measurements that are determined to be inside the rf hypervolume are scored based on the quality of their double dot features. Those that show 'good' double dot features, namely periodicity in two approximately orthogonal directions, we wish to give a high score; those showing other features, such as single or open dot features, we wish to designate a low score.

Score functions for classifying dot features exist in the literature, with implementations based on neural networks [71, 115, 116], Hough transforms [117], and pattern matching [83]. However, we found these approaches too noise-sensitive, too slow, or both. Our approach is centred around the use of the fast Fourier transform (FFT)

to identify periodicity in the data (Fig. 4.3.4 **b)**, **d)**).

We first take a measurement of resolution $n \times m$ (in the experiment, $n = m = 100$), resulting in $\vec{I}$ and $\vec{Q}$ from the in- and out-of-phase demodulation. We then combine the $\vec{I}$ and $\vec{Q}$ data to create a complex dataset $\vec{Z} = \vec{I} + i\vec{Q}$. The next step is to normalise the data to create $\vec{\mathcal{Z}} := (\vec{Z} - \bar{Z})/\sigma_z$, where $\bar{Z} = \bar{I} + i\bar{Q}$ and $\sigma_z^2 = \mathrm{var}[\vec{I}] + \mathrm{var}[\vec{Q}]$. The amplitude of the peaks in the Fourier transform is related to the amplitude of the peaks in $\vec{Z}$, which in turn is related to the impedance of the device during the measurement. The normalisation removes this dependence on the matching condition, allowing the score function to quantify purely the quality of the dot features in a given measurement.

After normalisation, we calculate the FFT of $\vec{\mathcal{Z}}$ using the discrete-time Fourier transform (DTFT) by

$$\mathrm{DTFT}[\vec{\mathcal{Z}}]_{\delta\gamma} = \frac{1}{NM} \sum_{a=1}^{N} \sum_{b=1}^{M} e^{-2\pi i(\nu_\delta^x a + \nu_\gamma^y b)} \mathcal{Z}_{ab}, \tag{4.3.1}$$

where $\vec{\nu}^x$ and $\vec{\nu}^y$ are vectors containing the frequencies that we search for in the measurement. Based on the approximate lever arms of the device and the voltage range of the measurements, we used $\vec{\nu}^x = \vec{\nu}^y = [0, 12]$ with a resolution of 100. In other words, we looked for the presence of 100 frequencies between zero and 12 in each scan. From the output of Eq. (4.3.1), $\mathrm{DTFT}[\vec{\mathcal{Z}}] \in \mathbb{C}^{100\times100}$, we obtain the score

$$s := \min\left(\max|\mathrm{FT}[\vec{\mathcal{Z}}]|_{\delta<\gamma}, \max|\mathrm{FT}[\vec{\mathcal{Z}}]|_{\delta>\gamma}\right). \tag{4.3.2}$$

The two elements inside the parentheses in Eq. (4.3.2) give the highest peaks on each side of the line $\nu_x = \nu_y$, which splits the data $\mathrm{DTFT}[\vec{\mathcal{Z}}]$ along its diagonal (red dashed line, (Fig. 4.3.4 **b)**, **d)**). A peak in $\mathrm{DTFT}[\vec{\mathcal{Z}}]$ at coordinates $\vec{\nu} = (\nu_x, \nu_y)$ is

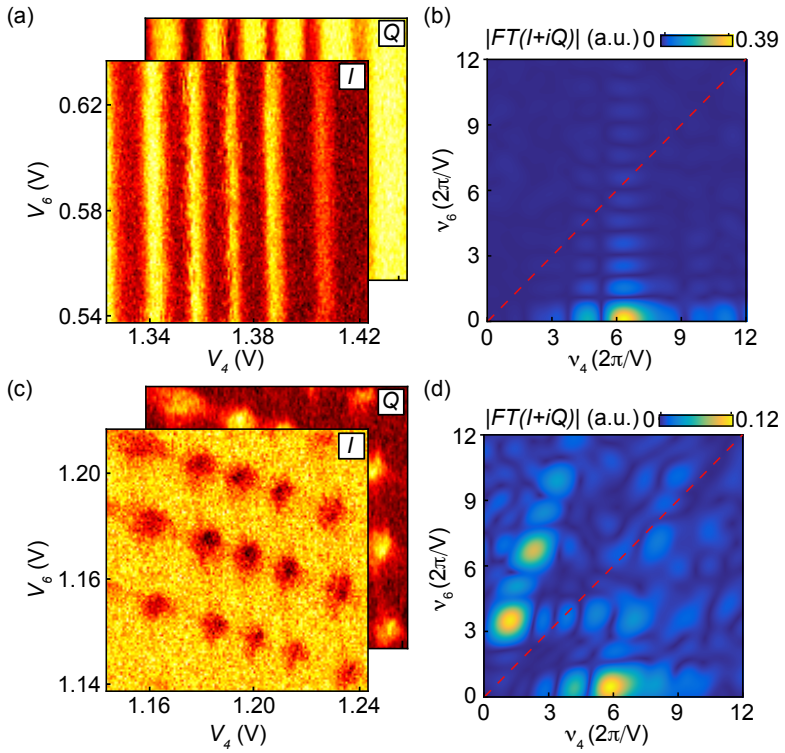


Figure 4.3.4: The FFT-based score function. **a)** A measurement of a single-dot regime in I and Q, and **b)** the corresponding FFT. Subplots **c)** and **d)** are in the same format. The red dotted line in (b) and (d) represents $\nu_x = \nu_y$. Measurement (a) received a score of 0.01, whereas measurement (c) received a score of 0.13.

due to a periodicity in $\vec{\mathcal{Z}}$, where $|\vec{\nu}|$ gives the number of periods and $\vec{\nu}$ is the vector normal to the oscillations. Thus peaks in DTFT$[\vec{\mathcal{Z}}]$ for which $(\nu_x > \nu_y)$ represent oscillations primarily along the x axis, and $(\nu_x < \nu_y)$ represent oscillations primarily along the y axis. Since we are looking for oscillations in both axes, we look for maximal peaks on either side of this dividing line. The score assigned to a given measurement is equal to the height of the smaller of these two maxima. This choice is motivated by the fact that we are specifically looking for double dots; strong single-dot features would have a large maximum on one side of the line $\nu_x = \nu_y$, but not on the other, and therefore receive low scores under this scheme (Fig. 4.3.4 **a**, **b)**).

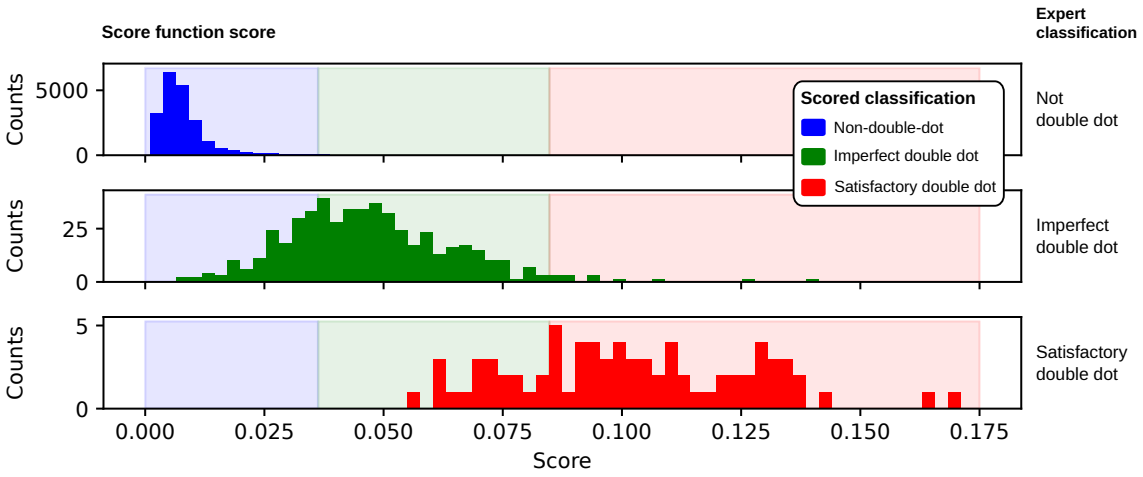


Figure 4.4.1: Score function score histograms compared to expert classification. Note that the y-axis scales are different on each histogram due to the imbalance in the class sizes. Each histogram represents scores given to measurements that were determined to be in the indicated class by human experts. The background colours represent the values that define each group such that the scores given by the score function and the labels generated by the experts are in optimal agreement. The score function was designed to have a low false positive rate, meaning each group tends to overlap more with lower-scoring classes than with higher.

4.4 Results

The results discussed in this section are from 24 hour-long tuning sessions in which the algorithm classified 20,000 measurements from within the rf hypervolume. 13 of these runs controlled device A; the other 11 were for tuning device B.

4.4.1 Score function validation

An important point yet discussed is whether or not the score function agrees with the human perception of a 'good' double dot. To verify that it does, two human experts were given the joyous task of classifying the measurements of these runs into one of three categories: no double dot, imperfect double dot, and satisfactory double dot. These experts had no knowledge of the score assigned to each measurement by the score function.

The scores are presented with their human-generated associated labels in Fig. 4.4.1. The algorithm gives us a continuous score (within the limits of floating-point arithmetic), whereas the humans were asked to classify the measurements into one of three classes. To quantify the degree of agreement between the algorithm and the experts, we generate thresholds for the algorithm score that define the boundaries between the three aforementioned classes.

Scores below $s = 0.0362$ are classified as no double dot; those between this value and $s = 0.0848$ are classified as an imperfect double dot; those scoring above this value are classified as a satisfactory double dot. These thresholds were calculated to optimise the degree of similarity between the two classification datasets. We then compared the human classification with the algorithm's classification using a confusion matrix (Table 4.1).

The score function was designed with an emphasis on low false positive rates, motivated by the high measurement throughput. We know that double dot features exist in the device, and we know that if we search for long enough, our algorithm will identify them. By designing the score function to have a low false positive rate, we can be confident that a high-scoring measurement contains double dot features. The confusion matrix (Table 4.1) firstly shows that the score function models an accurate representation of the experts' opinion, because the highest-scoring entry in each row i and column j is at $i = j$. Between Table 4.1 and Fig. 4.4.1, the data show that a high score corresponds to a high probability of the measurement containing double dot features.

4.4.2 Double-dot tuning

We demonstrated the performance of our algorithm by tuning both double-dot sub-arrays (A and B).

Table 4.1: Confusion matrix for all feature measurements from devices A and B

Devices A & B (%)	Expert Classification		
	Non-double dot	Imperfect double dot	Satisfactory double dot
score < 0.04	96.95	0.91	0.00
$0.04 \leq$ score < 0.08	0.25	1.42	0.08
score ≥ 0.08	0.01	0.11	0.29

Score improvement over time

The cumulative highest score shows that the algorithm finds double dots of increasing quality over the course of each tuning run (Fig. 4.4.2). To demonstrate the utility of the machine learning components of the algorithm, we ran a reduced version that did not include the Gaussian process, noise classifier, and random walk within the rf hypervolume. Instead of the noise classifier terminating the ramp within the rf hypervolume, every scan along the exploratory ramp was scored by the score function, with the ramp terminating once one of the gate voltages hit 0 V.

We also tested the utility of a random walk, whereby a random gate coordinate was selected and the resulting measurement scored. The full version of the algorithm outperforms both the reduced version and the random walk in terms of time efficiency. These three versions of the algorithm would, given enough time, converge on the same scores, since the underlying scoring system remains the same. The only fundamental change is in how the device voltage space is explored. The result that the full algorithm finds higher-quality double dot features more quickly than the reduced versions shows that the additional features reduce the amount of time spent outside the region of interest.

Example regimes

Fig. 4.4.3 shows the highest-scoring measurements from each of the 24 runs across devices A and B. All clearly show double-dot features, and would be suitable starting

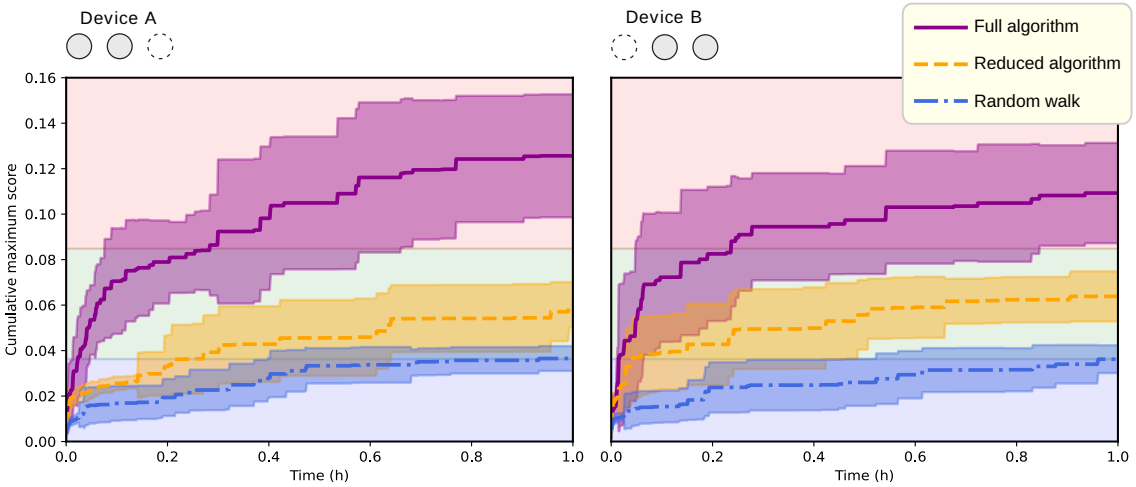


Figure 4.4.2: Improvement of maximum score over time for both devices. The horizontal bands represent the approximate classifications obtained in the main text, with the top (red) band representing satisfactory double dots.

points in the search for a qubit regime.

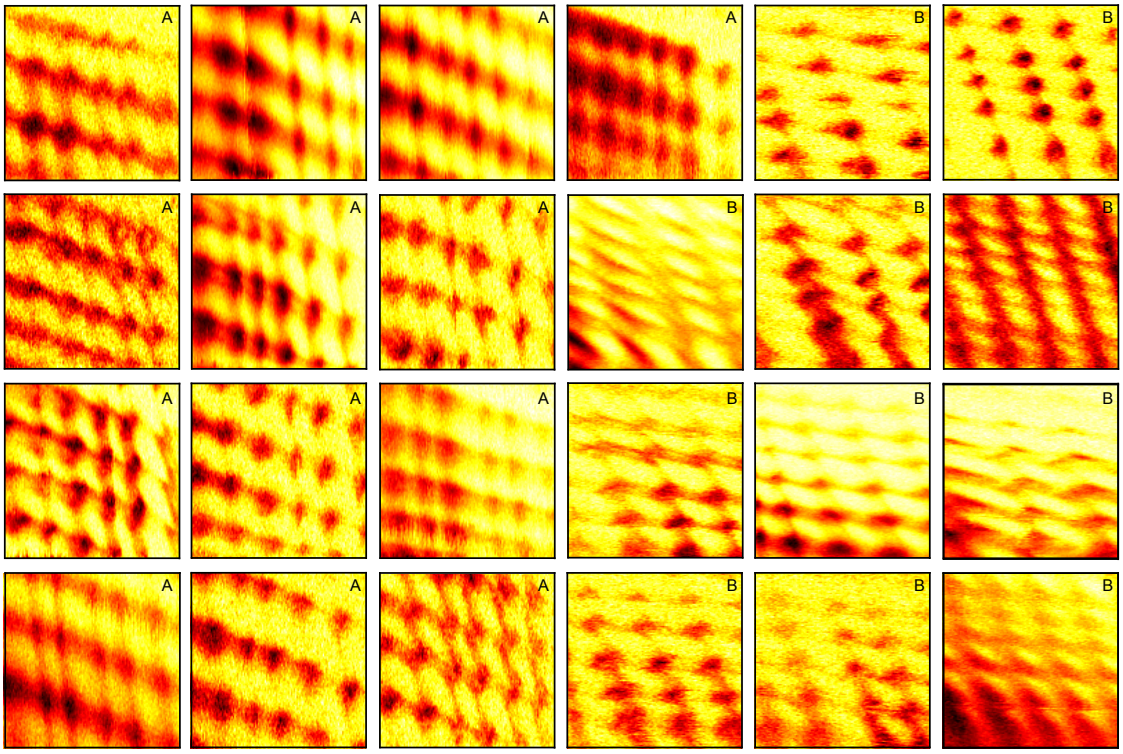


Figure 4.4.3: The highest-scoring measurement from each of the 24 runs taken over the tuning runs described in the main. The device measured for each scan is given in the top right.

4.5 Conclusions

The algorithm effectively tunes double dots in our device. By recording the score along with each measurement, the algorithm can be used alongside human experts, whereby they can select regimes that look promising for further characterisation. Our contribution represents the fastest reported automated tuning of a double dot device, and the first to do so using purely reflectometry measurements.

The time efficiency of the algorithm could be further improved with some fairly straightforward additions. Rather than randomly selecting the direction of the exploratory ramps, a decision-making algorithm such as Bayesian optimisation could be used to direct these ramps towards regions of high score, or to explore regions where the underlying score landscape is not known.

While fast algorithms are neat and useful from a scalability perspective, the most important feature of an algorithm is that the output is what was desired. I am therefore of the view that the most significant improvements to any tuning algorithm, including this one, may be brought about by ensuring the score function agrees with our knowledge of what makes a good double dot. This is not an easy task. Further advances in computer vision may be key to this, but using a supervised algorithm requires labelling huge amounts of data with a consistent scoring framework, which means using just a small number of people who we all agree know very well what a 'good' double dot means. I have consistently wrapped 'good' in quotation marks to highlight that this is a subjective term.

To objectively measure the success of tuning algorithms such as this one, it would be of great benefit to the field to develop standardised tests of efficacy. This is not straightforward, because the quality of devices varies so widely. The double dots in Fig. 4.4.3 were as good as we could get in this device, but may not receive a passing mark for more modern devices. However, standardised tests have proved of great utility in the machine learning community, with authors able to cite directly the accuracy or speed of their new architecture on well-known test datasets. This lack of standardisation makes it difficult to make direct comparisons to existing algorithms, such as Refs. [4, 86].

We here demonstrated the fastest automated tuning of a double quantum dot demonstrated in the literature. This is a good start, and this algorithm could serve as the starting point for an algorithm designed to tune larger arrays of quantum dots. This may require a redesigned score function, and possibly some heuristic methods for selecting the gate voltages. Larger arrays are naturally defined by more gates, meaning the dimensionality of the problem increases. Searching the full gate space using ramps may no longer be an efficient method; for instance, it may be more

effective to tune an array in sub-units of double dots rather than the whole array at once. This is an improvement on the earlier work of Ref. [4], not just due to the speed-up but because now we do not have to use a transport setup to tune the device before switching to an rf setup to measure it; these are now integrated under a single measurement protocol.

Chapter 5

Charge sensor compensation

The work in this chapter was first published in Ref. [118]. This chapter covers similar content, but with some additional details for clarity and completeness. I again worked on this project with Barnaby van Straaten, with some assistance from Federico Fedele.

We have demonstrated that it is possible to tune a double quantum dot using purely rf reflectometry measurements [91] (Chapter 4). In that work, the reflectometry circuit contained the device itself, bonded via its ohmic pads. However, single-shot spin measurements generally require a charge sensor such as a quantum point contact [49] or quantum dot [77], rather than the ohmic rf approach. Unfortunately, the use of a charge sensor in a tuning algorithm is complicated by the capacitive coupling between the device gates and the charge sensor. In this chapter, I present a solution that compensates for this crosstalk. The result is an algorithm that is beneficial for tuning as well as device characterisation and measurement.

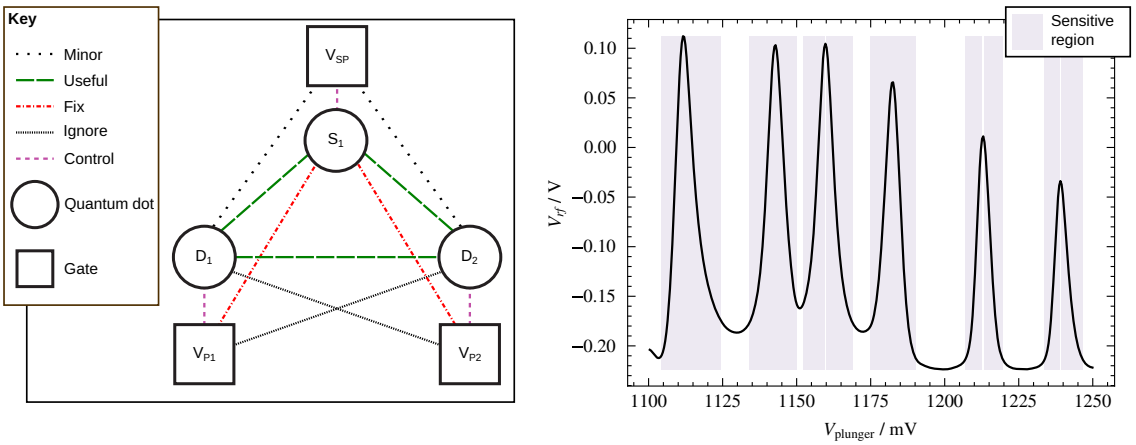


Figure 5.0.1: **Left panel**: Diagram highlighting the various types of capacitive coupling in a double dot system with a charge sensor. V_{SP} is the sensor's plunger gate, while V_{P_i} is the plunger gate for dot i. Because we only move the charge sensor plunger over a few millivolts, the effect of the capacitance indicated by the black dashed lines is minor. The capacitances indicated in green are useful for charge sensing and identifying triple points. The red dense dash indicates the crosstalk that is compensated for by the algorithm in this chapter. The crosstalk between device gates and other device dots, in black dense dash, is unwanted but beyond the scope of this chapter. The control capacitances are between gates and dots that those gates are designed to control. **Right panel**: Demodulated reflectometry signal when sweeping the voltage applied to a quantum dot's plunger gate. When transport is energetically allowed, we observe Coulomb peaks. In between these peaks, we observe blockade. The highlighted sections identify regions where the dot has a high transconductance.

5.1 Introduction

The high sensitivities provided by rf quantum dot charge sensing rely on the sensing dot being tuned to the side of a Coulomb peak (Fig. 5.0.1, right panel). Here, the transconductance is very large, meaning a small change in the dot's electrostatic environment leads to a large change in the demodulated output voltage. This electrostatic change can be induced by charge rearrangements in the nearby quantum dot array; indeed, this is the target of our measurements. However, the crosstalk between the device gates and the sensor dot means changes in these gate voltages also change the sensing dot's confining potential, and much more significantly than single charge transitions. As a result, measurements made using a quantum dot

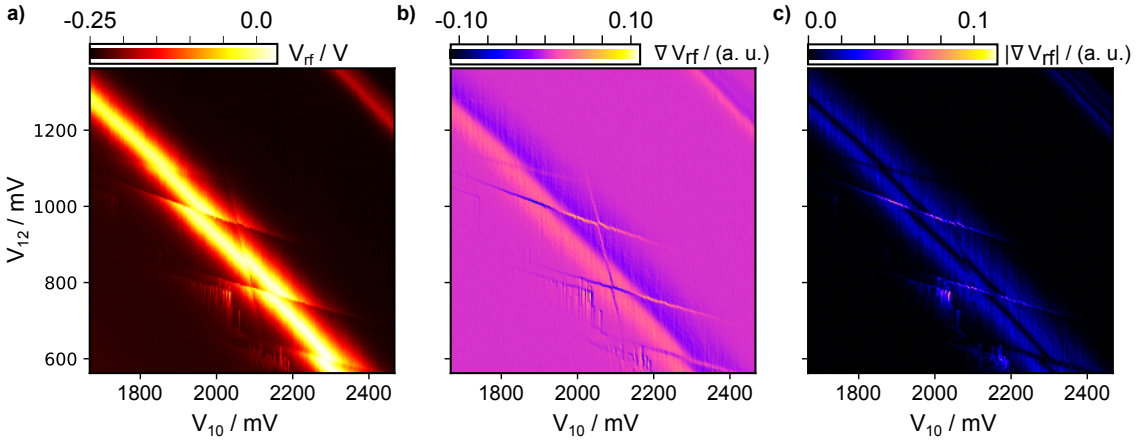


Figure 5.1.1: **a)** A typical charge sensing measurement of the double quantum dot device used in this chapter. Device details are given in Section 5.1.2. **b)** The gradient of a), which shows the device transitions more clearly. **c)** The absolute gradient of a), showing large regions of zero gradient in the sensing peaks. In these regions, the sensor is insensitive to charge rearrangements in the device.

charge sensor typically have alternating regions of high and low (to zero) sensitivity (Fig. 5.0.1, right panel; Fig. 5.1.1), which complicates the tuning procedure (whether automated or manual).

Techniques based on digital feedback have been developed to compensate for this crosstalk effect [76, 98]. These approaches rely on maintaining a constant current flowing through the sensor dot, but do not necessarily maintain a high transconductance, using available Coulomb peak flanks irrespective of their conductance gradient. These methods are thus not able to keep the readout contrast, i.e. the sensor's sensitivity, constant as the device plunger gate voltages are swept through ranges over 100 mV × 100 mV (depending on the device-specific lever arms).

In addition, methods based on control theory rely on the presence of several suitable sensing peaks. If a given peak no longer reaches the readout current (or voltage) setpoint, the plunger gate voltage will be swept until a new peak is found that does meet this criterion. However, many suitable Coulomb peaks are not always available (see Figure 5.2.2 **b**). Closed-loop control requires fast and constant measurement of

the sensor current, which may not be possible with standard laboratory equipment.

In this chapter, I present an automated approach that finds and maintains the high sensitivity of an rf sensor dot while sweeping across large ranges of a device's gate voltage space. Our algorithm employs Bayesian optimisation to tune the rf sensing dot to a regime where it is highly sensitive. Once the optimal readout contrast is found, a set of a specific type of virtual gate [67, 83, 84, 119], which each use the sensor dot's plunger gate to provide crosstalk compensation and are therefore called *compensated gates*, are used to maintain the readout sensitivity in fast $100\,\mathrm{mV} \times 100\,\mathrm{mV}$ measurement windows. This ac-based compensation is combined with a dc-based compensation algorithm to allow for the rf sensing of a stability diagram over $1\,\mathrm{V} \times 1\,\mathrm{V}$, in which we can distinguish more than 25×25 charge transitions, using a single sensing peak. Our compensation algorithm can be repeatedly recalibrated to arbitrarily extend this range, thus overcoming the limitation of the linear approximation that supports this approach (Section 5.1.1).

Problem statement

Changing device gate voltages over a small voltage ranges can blockade the sensing dot and drop the sensitivity to zero. When changing the voltages over larger ranges, the effect can be to deform the measurement dot entirely, either by opening the conductance channel and removing the confining potential or by suppressing the dot entirely.

To tune a device using charge sensing measurements without losing sensitivity, it is necessary to have a method to compensate for the effect of gate crosstalk. Ideally this method would be automated and effective globally in device gate space.

5.1.1 Problem analysis: constant capacitance model

In this section, I break down the problem in the context of the constant capacitance model, which is a widely-adopted circuit model of a quantum dot array.

In the constant capacitance model, the potential on the each of the charge nodes is given by

$$\vec{V}_D = \mathbf{C}_{DD}^{-1}(\vec{Q}_\mathrm{D} - \mathbf{C}_{Dg}\vec{V}_g), \tag{5.1.1}$$

where $\mathbf{C}_{DD}$ is a matrix containing the node-to-node capacitances, $\vec{Q}_D$ is a vector containing the charge associated with each node, $\mathbf{C}_{Dg}$ contains the node-to-gate capacitances, and $\vec{V}_g$ contains the voltages on each gate. To compensate for the effect of changing gate voltages on the sensor potential, the change in $\vec{V}_D$ should be zero for those elements relating to the charge sensor. Let us look at a sub-vector of $\vec{V}_D$, $\vec{V}_C$, that only contains nodes relating to the charge sensor. Assuming that the number of charges stays constant,

$$\begin{aligned}
\Delta\vec{V}_C &= \mathbf{C}_{DD}^{-1}(\vec{Q}_\mathrm{D} - \mathbf{C}_{Dg}\vec{V}_{g_2}) - \mathbf{C}_{DD}^{-1}(\vec{Q}_\mathrm{D} - \mathbf{C}_{Dg}\vec{V}_{g_1}) && (5.1.2)\\
&= \mathbf{C}_{DD}^{-1}(\mathbf{C}_{Dg}\Delta\vec{V}_g) && (5.1.3)\\
&= \vec{0}\,(\text{ideally}) && (5.1.4)
\end{aligned}$$

With knowledge of $\mathbf{C}_{DD}$ and $\mathbf{C}_{Dg}$, we can find a combination of gates that perfectly compensate the sensor. In a simulation where we have defined these matrices, this is very straightforward. In reality, however, this is not straightforward at all.

For compensating a quantum dot charge sensor, we can consider three potentials: the

two tunnel barriers and the dot itself. To keep all three of these constant, we need knowledge of the capacitive coupling between each gate and those nodes, and how the nodes are coupled to each other. To measure how gates affect the nodes of interest, we need some metric that we can measure the change in. For the dot potential, we can measure changes in the position of a Coulomb peak as a function of gate voltage. The effect of gate voltage on tunnel barriers has been examined in Ref. [89], but this approach requires a charge sensor to measure the gradient of tunnel rates as a function of gate voltages. Fundamentally, we cannot separate the three nodes of interest; the tunnel barriers are hidden from us and our only measurable unknown is a combination of the three nodes represented by the position of the Coulomb peak. The result is an overdetermined problem in which we have three constraints and one unknown, which has infinitely many solutions.

This makes it impossible to find virtual device gates that entirely compensate the sensor. To get around this, we separate the problem into two parts. We can get the majority of the compensation effect by just using the sensor's plunger gate to maintain the potential of the sensing dot, forming compensated gates. This forms the main compensation part of the algorithm. We separately tune the tunnel barriers to their optimal point in a subroutine. Rather than actively compensating them, we developed a fast algorithm that we can run whenever we need to update the barrier voltages.

Compensated gates

Compensated gates are a specific type of virtual gate, meaning a combination of real gates that primarily act on the device parameters. For compensated gates, we include a contribution from the sensor plunger gate that compensates for the effect of the device gates on the sensor's potential. Changing the voltage on such a compensated gate has the form

$$\Delta V_{\text{compensated}} = \Delta V_g - (\gamma \Delta V_g) V_P, \tag{5.1.5}$$

where V_g represents a device gate that is virtualised in $V_{\text{compensated}}$, V_P is the plunger gate of the sensor which we are compensating, and γ is a coupling ratio that will be defined in the following sections.

5.1.2 Experiments

In the experiments that follow, I demonstrate why the operating range of the algorithm is limited, due to the underlying assumption that the capacitance matrix that models the device is constant. I then detail techniques for optimising the state of the quantum dot charge sensor, and for maintaining this optimal tuning over device gate voltage space.

Device

These experiments were performed in an electrostatically-defined Ge/SiGe quantum dot (QD) fabricated in a similar manner to that described in Ref. [47]. A microscopy image of a nominally-identical device is provided in Fig. 5.1.2. Gates $V_1 - V_5$ are used to define quantum dots in the device array, while gates $V_{\text{LB}}, V_{\text{P}}$, and V_{RB} define the charge sensor. Gates V_{LS} and V_{RS} separate the sensing dot from the device dots, and were both set to 1.8 V for this experiment. An ohmic of the charge sensor is embedded in an L-matching network with a 92 pF decoupling capacitor C_{d}, 2.7 µH inductor L, and a parallel capacitance given by the total parasitic capacitance of the device C_{p} (Fig. 5.1.2) [101]. We used a Zurich Instruments UHFLI to drive the circuit at a power of −90 dBm and demodulate the reflected signal into the in-phase and quadrature components.

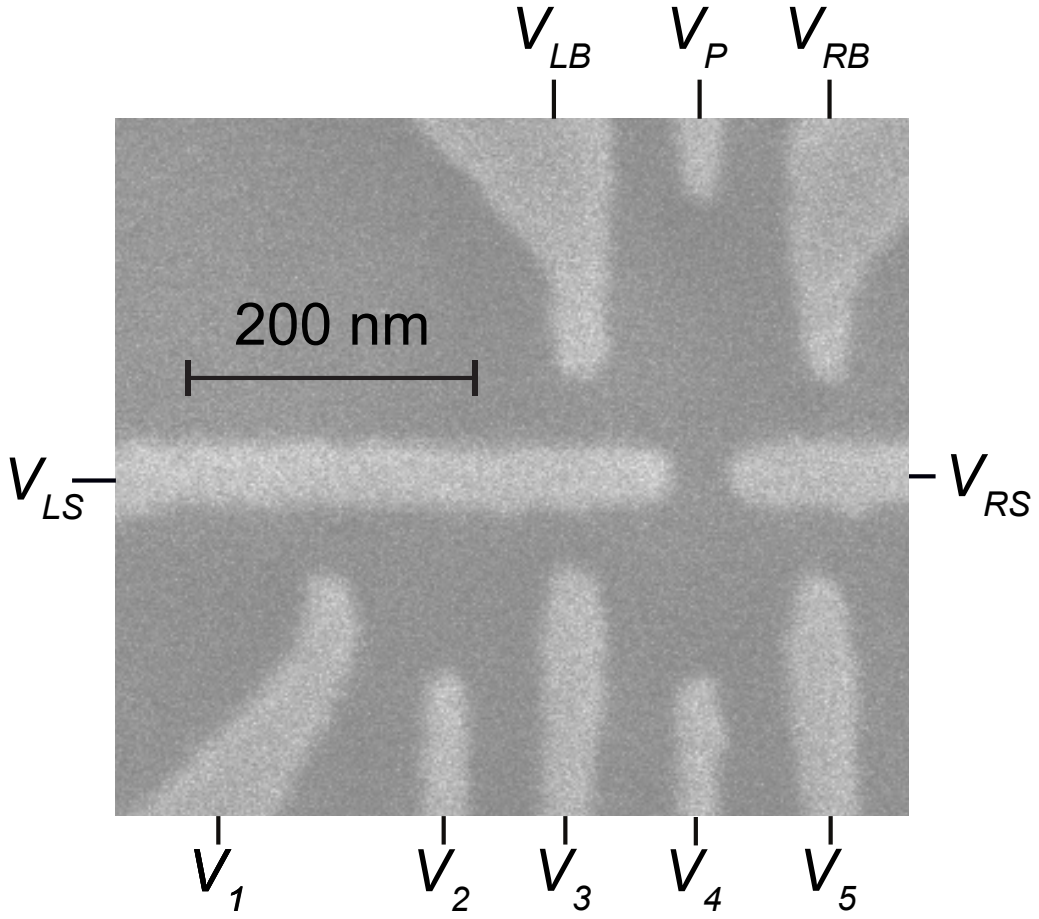


Figure 5.1.2: **a**) A scanning electron microscopy image of a device nominally identical to the one used in this experiment. The gate electrodes connected to microwave lines via a bias tee are colour-coded in pink. **b**) Demodulated signal from the rf-QD obtained by PCA when the device is in a double dot regime. For a given point in gate voltage space, we perform this 2D measurement by ramping gates V_2 and V_4 in a raster pattern with an arbitrary waveform generator.

Measurements

This algorithm primarily relies on fast two-dimensional (2D) measurements taken by ramping the voltage on gates V_2 and V_4 in a raster pattern using an arbitrary waveform generator (AWG), as described in Section 3.3.4 [91, 106, 120]. The output waveforms used to generate these 2D measurement windows are compensated to minimise the effect of distortion introduced by the high-pass filters as described in Section 3.3.4. We measured the demodulated in-phase (I) and out-of-phase (Q) components of the reflectometry signal from the rf-QD sensor for each pixel with

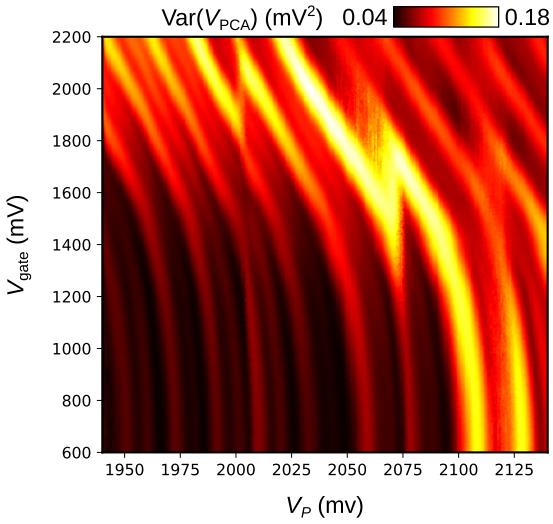


Figure 5.1.3: Coulomb peaks in a charge sensor controlled by V_P. The coupling between the sensor dot and V_{gate} is not constant, which we see by the curvature in this plot. Thus the *constant* in constant capacitance model is only locally applicable.

an integration time of 1 µs. As before, we optimally capture both the I and Q components of the rf-QD signal using principal component analysis (PCA). The signal of the rf-QD demodulated in this way is labelled V_{PCA} and corresponds to the first principal component of the measured data.

5.1.3 Non-constant capacitance

Before getting into the algorithm proper, I want to highlight that this algorithm is a linear solution to a non-linear problem. The constant capacitance model is simplistic and does not fully capture a quantum dot's behaviour. The coupling between any given device gate and the sensing dot is not a constant value. Sweeping the sensor plunger V_P and another device gate shows a non-linear relationship (Fig. 5.1.3). This is likely due to the physical position of the quantum dot changing as the gates change the potential landscape, resulting in a change in the capacitive coupling between any given gate and the quantum dot.

Fully capturing this behaviour would require a very complex model and training

it would be effectively like training a digital twin. As an alternative, my design philosophy for this algorithm is that it is accurate locally and fast to re-calibrate. Thus, while a single calibration cannot decouple the device gates from the sensor across the operating gate voltage space, the calibration procedure can simply be run again once the limits of the constant capacitance approximation are reached.

5.2 Algorithm

As a result of the analysis in Section 5.1.1, we split the algorithm into two parts. The first part handles the tuning of the sensor gate's tunnel barriers, and does so by using Bayesian optimisation to maximise the state contrast (*visibility*) in the charge sensor. The second part implements the compensated gates introduced in Section 5.1.1. A flowchart of the key parts of the algorithm is given in Fig. 5.2.1.

5.2.1 Optimally tuning charge sensing dot barriers

One benefit of embedding the charge sensor, rather than the device array, in the reflectometry circuit is that we can adjust the impedance of the sensing dot to improve the quality of the impedance match, without affecting the regime of the device array. To find the optimal barrier thickness, we define a score function based on the charge state contrast measured by the charge sensor, and optimise this score function using Bayesian optimisation.

Variance-based score function for sensor visibility

At optimal impedance matching, we expect to have the highest possible contrast between charge states, because the gradient of the reflection coefficient $|\Gamma|$ with respect to the impedance of the reflectometry circuit Z_{load} is maximised. To optimise the sensor's charge state contrast, or *visibility*, we use a black-box approach based on

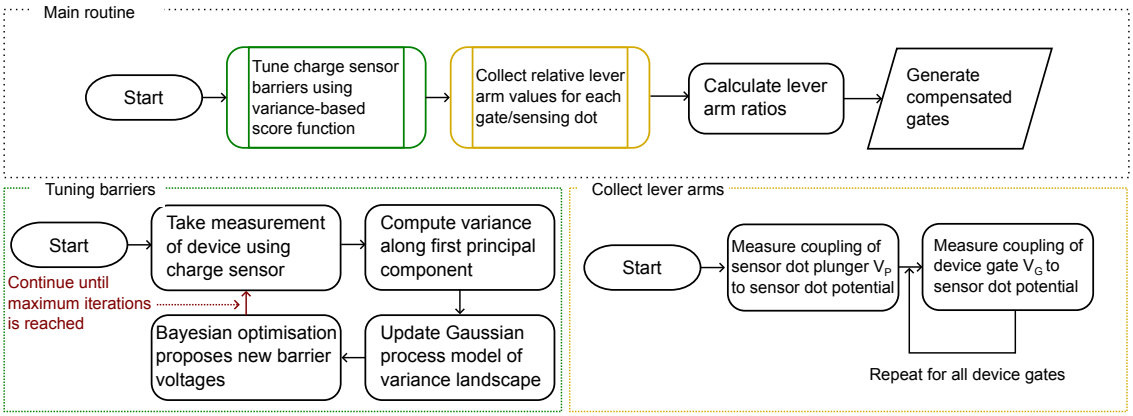


Figure 5.2.1: A flowchart for the key components of the compensation algorithm. By using the compensated gates generated by the algorithm, we can explore the device voltage space using the compensated charge sensor across a voltage range that depends on the extent of the non-linearity of the lever arms.

this expectation. Device charge stability diagrams measured via large (and therefore high-contrast) charge sensing peaks will contain data with a high variance; conversely, those measured via small charge sensing peaks, or while the sensor is in Coulomb blockade, will have small variances. Thus we use the variance of the fast 2D charge stability diagrams as a metric for how well tuned the charge sensor barriers are.

To evaluate the score function, we initially take a fast 2D measurement as described in Section 5.1.2 and determine the variance of the data after projection onto the first principal component, $\mathrm{Var}(V_{\mathrm{PCA}})$. The variance is a natural choice for a score function, and can be used to optimise not just the position of the Coulomb peak but also many other experimental parameters such as rf frequency and power (Section 5.3.1). The variance of V_{PCA} is conceptually the same as considering the variance of the demodulated I component in a measurement where the phase is perfectly tuned (Section 3.4.1). If we only had access to the I component of the demodulated data, the score function could be evaluated by taking the variance in the I component. In this case, the phase would need to be optimised as a parameter in the subsequent Bayesian optimisation.

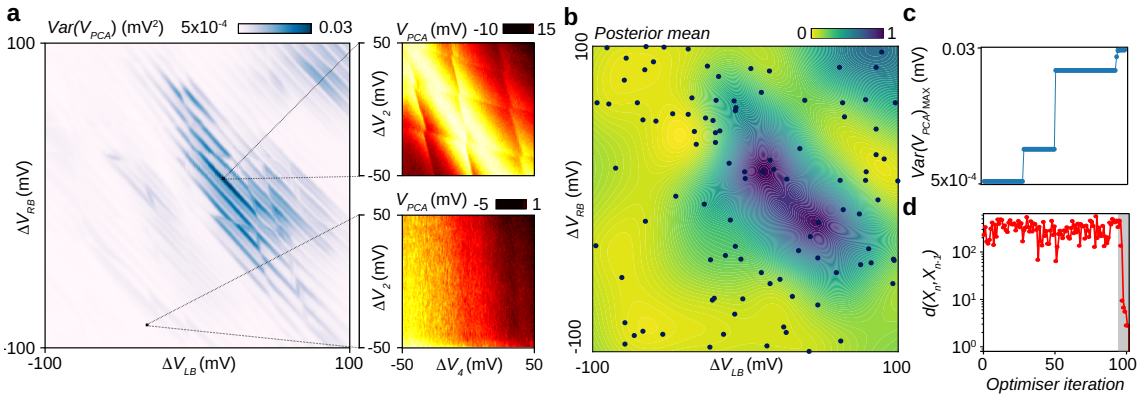


Figure 5.2.2: **a**) An example of the underlying score function over which we run the barrier optimisation routine, representing the variance of $\mathrm{V_{PCA}}$ for a 200 mV × 200 mV window in the gate voltage space given by the V_{LB} and V_{RB}. Each pixel in this scan represents the variance of $\mathrm{V_{PCA}}$ in a fast 2D measurement. The insets shows examples of these measurements corresponding to two pixels with high and low variance. In regions where the rf-QD is sensitive, the variance is high (upper inset). In regions where the rf-QD is not sensitive, the variance is low as the measurement is primarily noise (lower inset). **b**) The posterior of a Gaussian process over a set of values of $\mathrm{Var}(V_{\mathrm{PCA}})$ from **a**, marked by black points and chosen by the optimiser. The optimiser aims to find the maximum value of $\mathrm{Var}(V_{\mathrm{PCA}})$ in the gate voltage window considered. **c**) The maximum value of $\mathrm{Var}(V_{\mathrm{PCA}})$ that was found by the optimiser as a function of the number of iterations. As the optimisation progresses, the optimal value of $\mathrm{Var}(V_{\mathrm{PCA}})$ increases. **d**) Distance between sequential gate voltage coordinates chosen by the optimiser. This distance decreases significantly for the last few iterations of the optimiser (grey area).

Bayesian optimisation of rf-QD sensor barrier gates

We aim to optimise our readout sensitivity by controlling the transparency of the rf-QD's tunnel barriers through gates V_{LB} and V_{RB}. Our Bayesian optimiser proposes barrier gate voltages at which to evaluate the score function. In Fig. 5.2.2 **a**, we show an example of the gate voltage parameter space over which we optimise the barriers. Each pixel in this figure panel is the variance of $\mathrm{V_{PCA}}$ for a fast 2D measurement. To find the optimal barrier voltage, we could perform this measurement and take the values of V_{LB} and V_{RB} that result in $\max[\mathrm{Var}(V_{\mathrm{PCA}})]$. However, this is a very time-consuming approach, considering this measurement is made up of 500 × 500 fast 2D measurements, with each separated by at least one voltage change on a

DAC. This introduces a timing overhead due to communication with the DAC. In our optimisation approach, V_{LB} and V_{RB} are chosen by the optimiser. Initially, these voltages are chosen to explore the domain of the score function, building the Gaussian process model (Fig. 5.2.2 **b**). Later iterations exploit the information in the model to converge on the global maximum (Fig. 5.2.2 **c**, **d**). A sharp Coulomb peak, optimal for sensing, will be thus found at the coordinates in gate voltage space which correspond to the global maximum of $\mathrm{Var}(V_{\mathrm{PCA}})$. Depending on the barrier voltage range considered, we found that optimising the barriers in this way requires approximately 100 fast 2D measurements.

Our Bayesian approach is motivated by the fact that the score function landscape displayed in Fig. 5.2.2 **a** defeats standard optimisers such as Nelder-Mead or BFGS. We hypothesise that this is due to the complex, periodic nature of the optimisation landscape given by Coulomb peaks in the rf-QD.

We use a Bayesian optimiser with a Matérn 5/2 kernel and length scale constrained to be greater than the Coulomb peak spacing. This forces the model to only capture the large-scale variations in the score function, without the short-scale periodic variation. We found that making the optimiser blind to the score function's periodicity in this way allowed it to efficiently find the global maximum. The intuition is as follows. In the region where the tunnel barriers are well tuned, we measure tall Coulomb peaks, but we also measure Coulomb blockade. The long length scale of the optimiser kernel forces the uncertainty to be large in this area, but the mean function is also very high compared to the rest of the parameter landscape. In the final iterations of the optimiser, it exploits this knowledge and seeks to find the true maximum, which is within this region where the mean function is relatively large.

5.2.2 Measuring gate-to-dot coupling

By executing the Bayesian optimisation routine, we identify an optimal Coulomb peak for sensing. This Coulomb peak, present in the rf-QD sensor, is affected by the device gate voltages due to crosstalk. We aim to create compensated (virtual) gates for the quantum dot device that allows us to move through the devices' gate voltage space while keeping the rf-QD Coulomb peak fixed in the 2D measurement window. We achieve this by compensating each device gate with a contribution from V_{P}. The resulting virtual gates $V_{1-5}^{'}$ necessarily require V_{P} to act in the opposite direction to the device gate.

These compensated gates require knowledge of the proportional effect of each gate V_{1-5} and V_P on the position of the charge sensing peak. The strength of the compensating contribution from V_{P} in each compensated gate $V_i^{'}$ is given by a coefficient γ_i, where $\gamma_i = \alpha_i/\alpha_P$ and α_P is the strength of the effect of V_{P} on the position of the Coulomb peak, which we use as a proxy for the potential of the sensing dot. The values of γ_i quantify the relative strength of each device gate compared to the strength of V_{P} on the sensor dot's potential. We would expect that γ_i is thus less than 1 since the plunger gate is the most strongly coupled to the sensor dot of any of the gates of interest.

I here present two methods for obtaining the coupling ratios γ_i. The first is based on the finite difference gradient of the sensor's Coulomb peak in one-dimensional measurements. The second is based on the position of the Coulomb peak in the 2D measurements described in Section 5.1.2.

Method based on two-dimensional measurements

In this approach, we measure the position of the Coulomb peak in a 2D measurement and measure how this position depends on gate voltages V_{1-5} and V_{P} (Fig. 5.2.3 **a,**

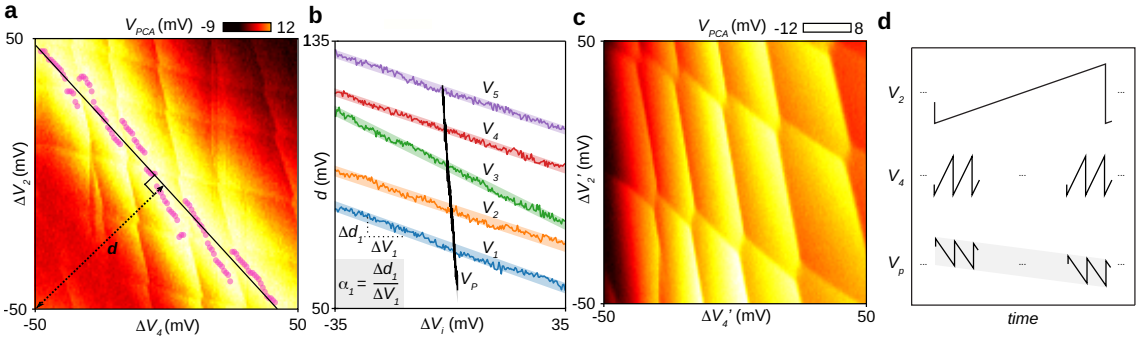


Figure 5.2.3: **a**) A 2D measurement window. Purple points identify the rf-QD Coulomb peak maximum following the method described in the main text, with $\boldsymbol{d}$ as in Eq. (5.2.1). The solid line is a regression fit through these points. The dashed line indicates d, the perpendicular distance of the rf-QD Coulomb peak to the lower left corner of the 2D measurement window. **b**) Values of the Coulomb peak offset, d, as a function of changes in gate voltage for each of the gates of interest. Note the curves have been vertically offset to aid the reader. The sensor's plunger gate, V_{P}, has the strongest gradient, as expected. **c**) Example of a 2D measurement window after ac compensation. The uncompensated version can be found in Fig. 1 **b**. **d**) The top two waveforms are schematics of the voltages applied to device gates to perform a 2D measurement window. The bottom waveform is the ac compensation voltage applied to V_{P}.

b). This position (d) is defined to be the shortest distance from the lower left corner of the 2D measurement to a linear regression fit of the rf-QD Coulomb peak. The procedure for fitting this line is now described.

Initially, two linear models are generated from the measurement: the first contains the pixels with the maximum value in each column, and the second with the maximum value in each row. We then generate a line fit for both sets and keep the set with the fit that has the largest r^2 value. This model selection routine stabilises the following part of the algorithm. The selected peak line now lies across the top of the Coulomb peak in the charge sensor and is modelled by the equation $ax + by + c = 0$ (solid black line, Fig. 5.2.3 **a**). To get a measure of the position of the peak line, we calculate the shortest distance from it to the origin of the measurement (x_0, y_0) (dashed black line, Fig. 5.2.3 **a**) and name this distance d. The point on the peak line closest to the origin is at (m, n) and the line from here to the origin is perpendicular to the

peak line. The distance from (x_0, y_0) to (m, n), is

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}, \tag{5.2.1}$$

and since $(x_0, y_0) = (0, 0)$ in pixel space,

$$d = \frac{|c|}{\sqrt{a^2 + b^2}}. \tag{5.2.2}$$

We quantify the strength of the effect of the gate voltages on d as $\alpha_i = \Delta d_i / \Delta V_i$, where Δd_i is the change in d when the device gate voltage V_i is changed by a given ΔV. We can choose a gate voltage range ΔV_i within which the values of α_i are constant, i.e. there is a linear relationship between d and V_i.

Method based on finite differences gradient

We can obtain the relative coupling strengths of each gate on the potential of the rf-QD by measuring the gradient of a Coulomb peak as a function of each gate's voltage. A Coulomb peak is approximately Lorentzian and therefore does not have a constant gradient. However, for a local compensation approach, we can approximate the gradient using finite differences.

We initially tune to the side of a Coulomb peak in the rf-QD sensor. For each gate, we measure at two gate voltages, $V_{\text{gate}} - \delta V_{\text{in}}/2$ and $V_{\text{gate}} + \delta V_{\text{in}}/2$, and calculate the finite difference gradient by $\delta_h[V_{\text{gate}}] = \delta V_{\text{out}} / \delta V_{\text{in}}$. At each of the two gate voltages, we measure N samples and calculate the gradient based on the averages of these samples (Fig. 5.2.4).

In this method, the finite differences gradient directly provides the measure α_i for the strength of the gate i. This approach is useful for situations in which there is no

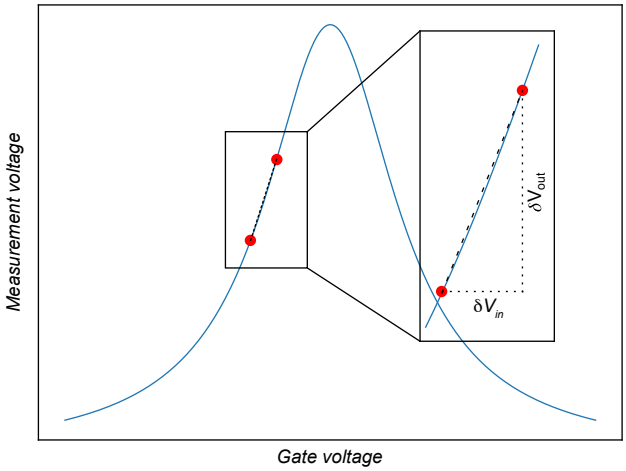


Figure 5.2.4: Demonstration of the finite differences method for obtaining the gradient of the sensor peak. N samples are taken at each of the two positions indicated by the red points, and the average of each of these two sets is used to calculate the gradient $\delta V_{\text{out}}/\delta V_{\text{in}}$

fast measurement setup, or when the full Coulomb peak does not fit in the width of the 2D measurement window.

This method sometimes does not work accurately due to device charge transitions interrupting the Coulomb peak in the charge sensor. In this case, it may be necessary to measure the Coulomb peak for several values of the gate and perform linear regression on the data to extract the gradient, much like in the other method. However, this significantly slows down the procedure due to the overhead from communicating with the DAC used to set the voltage on each gate.

5.2.3 Compensating for crosstalk

The strength of the compensating contribution from V_{P} in each compensated gate V_i' is given by the coefficient γ_i, where $\gamma_i = \alpha_i/\alpha_P$, and α_i is the strength of the effect of V_{i} on d (P denotes the sensor plunger). The values of γ_i quantify the relative strength of each device gate compared to the strength of V_{P} on d.

Dc compensation

As we perform a long-range device gate sweep, we navigate the device's stability diagram. The position d of the rf-QD Coulomb peak in the measurement window changes as a result of this sweep. To compensate for a change in voltage ΔV on device gate i, the compensated device gate $V_i^{'}$ must increment the voltage on the sensor plunger gate V_{P} by a value opposite in sign and scaled by the relative strength of the compensating contribution, $-\gamma_i \Delta V$. This compensated device gate has a unit vector defined by

$$\hat{V}_i^{\prime} \propto (\hat{V}_i - \gamma_i \hat{V}_{\mathrm{P}}). \tag{5.2.3}$$

To generalise this concept, we first define an uncompensated virtual gate, which has an arbitrary direction in gate voltage space spanned by $\{V_1, V_2, V_3, V_4, V_5, V_P\}$:

$$\hat{V}_{\mathrm{arb}} \propto [\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, 0]^T, \tag{5.2.4}$$

where ϵ_i represents the contribution of gate i to the virtual gate.

The corresponding compensated virtual gate is constructed by including the contributions to V_P:

$$\hat{V}_{\mathrm{arb}}^{\prime} \propto \left[\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, -\sum_{i=1}^{5} \gamma_i \epsilon_i\right]^T. \tag{5.2.5}$$

Ac compensation

Within a measurement window, we observe the rf-QD Coulomb peak due to the presence of crosstalk. To compensate for this crosstalk, and maintain the high-transconductance edge of the peak throughout the measurement, we apply a com-

pensating waveform to V_P. This third waveform is a linear combination of the two waveforms being used to generate the plunger fast sweeps (Fig. 5.2.3 **c, d**):

$$V_{\mathrm{P}}(t) = -\begin{bmatrix}\gamma_2 \\ \gamma_4\end{bmatrix} \cdot \begin{bmatrix}V_2(t) \\ V_4(t)\end{bmatrix} \tag{5.2.6}$$

where $V_{\mathrm{P}}(t), V_2(t), V_4(t)$ are the ac components of the voltages applied to each of the three gates. V_{P} compensates for the cross-capacitive coupling between the device gates and the rf-QD. In this way, we remove the gradients corresponding to the rf-QD Coulomb peak from the 2D measurement window background (Fig. 5.1.2 **d**). This ac compensation approach is readily applicable to arbitrary pulses involving any number of device gates. Eq. (5.2.6) is thus generalised by including additional time-dependent components scaled by the relevant γ value.

5.3 Results

5.3.1 Validation of variance-based noise function

In this section, I demonstrate why the variance of a two-dimensional measurement with a large number of pixels is insensitive to various sources of experimental noise, and is therefore a reliable metric. This analysis is based on the variance of the variance-based score function, which is frankly rather confusing. For the sake of clarity, I will refer to the variance-based score function as *the metric*.

Pixel values in our measurements are made up of the true signal and a noise contribution,

$$M = S + N, \tag{5.3.1}$$

where M is the measured value, S is the signal contribution and N is the noise contribution. The noise is drawn from a zero-mean Gaussian distribution with variance σ_N^2:

$$N \sim \mathcal{N}(\mu = 0, \sigma_N^2). \tag{5.3.2}$$

The variance of our entire two-dimensional measurement, which is the value of the metric, is given by

$$\sigma_M^2 = \frac{1}{n}\sum_i m_i^2 - \frac{1}{n^2}\left(\sum_i m_i\right)^2, \tag{5.3.3}$$

where n is the number of pixels in our measurement M and i indexes each pixel. This formulation makes the assumption that the probability of each value in M is equal.

By analysing the propagation of errors (Appendix B), we can characterise the noise tolerance of the metric. The result of this analysis is that

$$\sigma_T^2 = \frac{4\sigma_N^2}{n}\sigma_M^2, \tag{5.3.4}$$

where σ_T^2 is the variance of the metric, σ_N^2 is the variance of the noise function N (from Eq. (5.3.1)) and n is the number of pixels in the measurement.

We can verify this result by simulation. To simulate the charge sensing peak, we created a Lorentzian function centred around the diagonal of our simulated measurement and added noise. By repeatedly sampling the variance of this function we obtained the variance of the variance, which gives the y-axes in Fig. 5.3.1. We then varied the measurement noise scale and the number of points in the measurement.

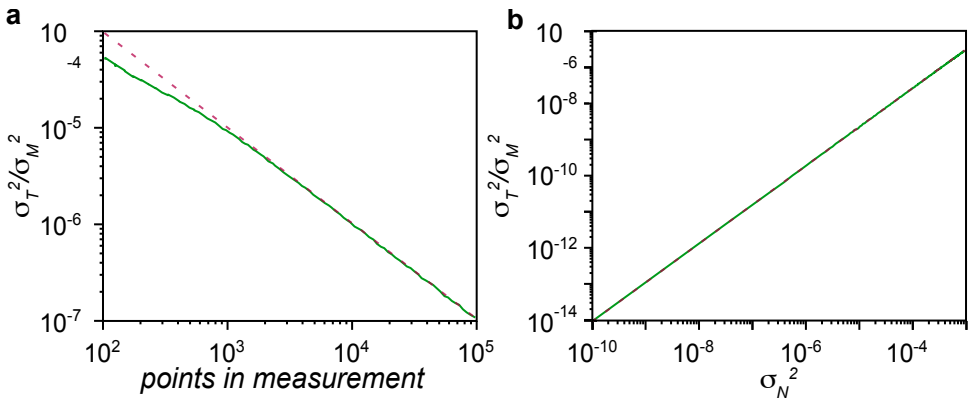


Figure 5.3.1: Variance of the metric, σ_T^2, as a function of **a)** number of points in the measurement and **b)** measurement noise scale. There were 10,000 points in the measurements for **b**.

The results agree with those expected due to Eq. (5.3.4), showing that the variance of this metric is small for a large number of pixels.

The result is that the noise sensitivity of the variance metric is extremely low, especially when a large number of points are used. Our measurements contain $128 \times 128 = 16,384$ pixels. The noise of the metric scales with n^{-2}, where n is the number of pixels, and linearly with the noise inherent in the measurement.

Tuning experimental parameters

This metric is not restricted to tuning the tunnel barriers of the charge sensor. In fact, it is useful for tuning any experimental parameter to optimise the reflectometry measurement signal-to-noise ratio.

As an example, consider a measurement that is designed to optimise the frequency of the reflectometry signal. If we sweep the frequency while watching the outcome of successive fast $100\,\mathrm{mV} \times 100\,\mathrm{mV}$ measurements, we will observe purely noise when the frequency does not align with the measurement circuit's resonating frequency, then signal when it is aligned, and then back to noise when the frequency is too high (Fig. 5.3.2). The frequency is optimal when the variance is maximised.

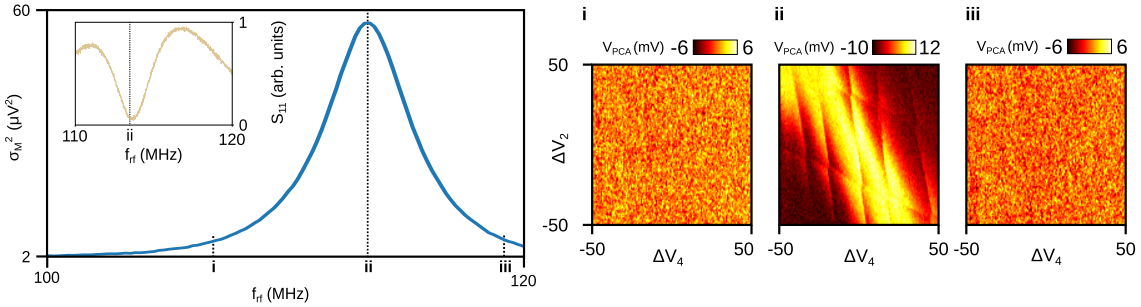


Figure 5.3.2: Variance of a 100 by 100 mV measurement at a fixed point in voltage space as a function of reflectometry frequency. As the frequency approaches the resonant frequency of the readout circuit, the visibility of the charge sensor increases, leading to an increase in variance. The inset shows the cavity S_{11} response as a function of frequency for comparison. Subfigures **i**, **ii**, and **iii** show examples of these measurements at different rf frequencies.

5.3.2 Compensated gate performance

A perfectly-compensated gate V_g' would have an associated $\alpha_g = 0$, meaning the Coulomb peak offset d does not change as the compensated gate voltage V' is modified. We therefore measure the performance of our dc compensation algorithm by calculating the standard deviation of d, σ_d, as a function of gate voltage sweeps for three cases. In the first case, we generate uncompensated virtual gate sweep directions as per Eq. (5.2.4). These virtual gate sweep directions are constructed as an arbitrary linear combination of gates V_1 to V_5. In the second case, we generate compensated virtual gates as per Eq. (5.2.5). In this case, the virtual gate sweep directions are constructed as in the uncompensated case, but a compensation term is included. Finally, we test the performance of each of the five compensated device gates as per Eq. (5.2.3). In this case, each compensated gate voltage sweep is that of a single device gate and V_P.

For this performance test, a Coulomb peak was first selected by the Bayesian optimisation routine described in Section 5.2.1. The same peak was used for each of the benchmarking runs. To produce the arbitrary virtual gate directions, we generated

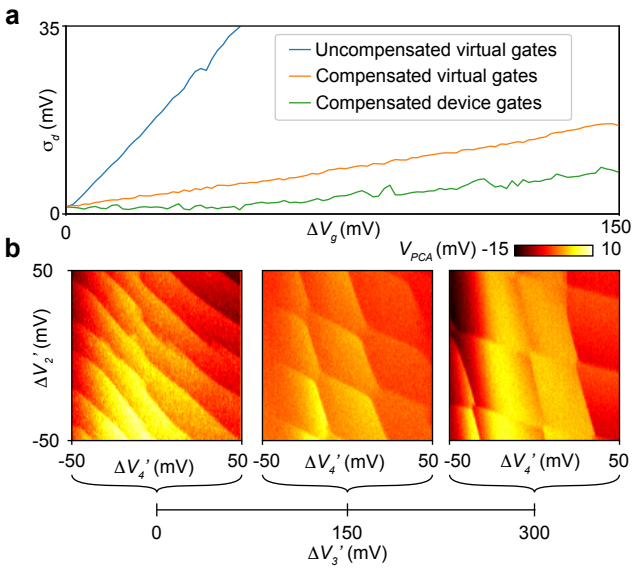


Figure 5.3.3: **a**) Standard deviation of a Coulomb peak's position quantified by d, σ_d, as a function of the length of the gate voltage sweep, ΔV_g, for uncompensated virtual gates (as per Eq. (5.2.4)), and for a different combination of gates in the compensated case (as per Eqs. 5.2.3, 5.2.5). **b**) Three 2D measurement windows for different gate voltage configurations. These configurations are accessed by changing the value of the compensated middle barrier gate voltage, V_3', by $\Delta V_3'$. Our compensation algorithm allows us to choose a Coulomb peak of our sensor dot and use it to explore the device gate voltage space going from a single dot to a double dot regime. Ac compensation is applied for each of these 2D measurement windows.

100 random vectors from the standard-normal distribution $\vec{\epsilon}_1, \ldots, \vec{\epsilon}_{100} \in \mathbb{R}^5$. These vectors were then used to create both uncompensated virtual gates and compensated virtual gates.

Fig. 5.3.3 **a** shows σ_d as a function of the length of the gate voltage sweep, ΔV_g, for the three different sets of gate sweeps described above.

Without dc compensation, at $\Delta V_g \approx 50\,\mathrm{mV}$ the Coulomb peak is no longer visible within the measurement window and σ_d saturates at 35 mV. Also, as we further increase ΔV_g, a different Coulomb peak becomes visible in the measurement window. With dc compensation, we observe that the algorithm drastically reduces σ_d over a gate voltage range of 150 mV for both compensated device gates and compensated virtual gates. Ramping compensated device gates has a smaller effect on σ_d than

ramping compensated virtual gates. This might have to do with non-linearities in the effect of gate voltages that are accentuated in their combination. Our compensation algorithm applied to V_3 allows us to measure device operation regimes from single to double dot for a given Coulomb peak of the sensor dot (Fig. 5.3.3 **b**).

Large, fast stability diagram measurements

By combining both ac and dc compensation algorithms, we are able to produce large stability diagrams with clear charge transitions by maintaining the chosen Coulomb peak's offset for the sensor dot fixed. We demonstrate this by measuring 1000 mV x 1000 mV stability diagrams made up of 10 x 10 2D measurement windows. In Fig. 5.3.4 (**a**), it is evident that when no compensation is applied, the visibility of charge transitions in the stability diagram varies significantly. The effect of crosstalk manifests in seven Coulomb peak transitions of the sensor dot. With dc compensation applied, a Coulomb peak chosen by the Bayesian optimisation routine is used and its offset is kept fixed. The sensor's sensitivity remains thus constant across different 2D measurement windows (Fig. 5.3.4 **b**). Still, we observe a clear change in the sensor's sensitivity within individual 2D measurement windows. Using both dc and ac compensation, the sensor's sensitivity is kept mostly constant across the full stability diagram (Fig. 5.3.4 **c**). This shows the extent of our compensation algorithm's capabilities, with charge transitions being visible for almost a volt in each device plunger direction.

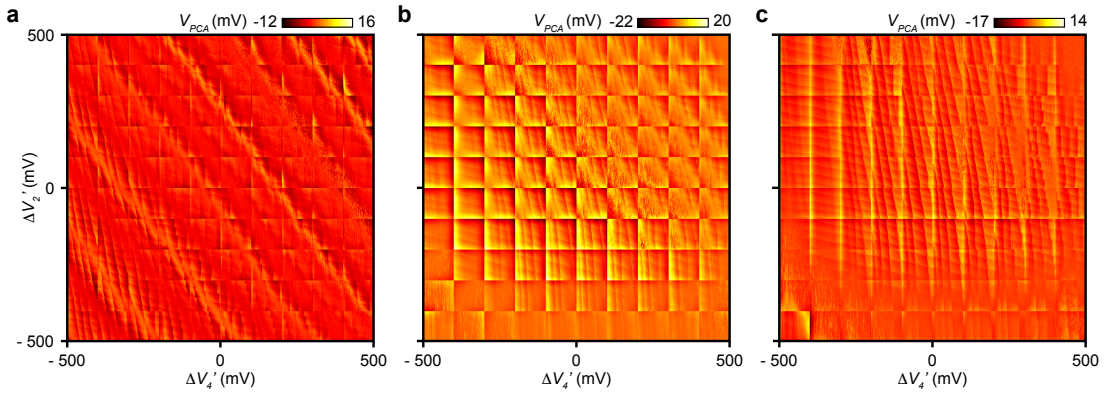


Figure 5.3.4: 1000 mV x 1000 mV mosaic of 2D measurement windows for three different cases: **a)** without ac or dc compensation, **b)** just dc compensation for V_2 and V_4, and **c)** both dc and ac compensation for the same gates. The dc compensation maintains the sensor Coulomb peak offset, d, fixed and the ac compensation makes charge transitions significantly clearer. The edges of the 2D measurement windows are still noticeable in panel c due to imperfections in the waveform compensation used to correct the distortion introduced by filters.

5.4 Conclusions

We demonstrate how a quantum dot charge sensor can be quickly tuned and compensated for both ac and dc sweeps of a device's gate voltages. A combination of the algorithms presented results in a 1 V $\times$ 1 V stability diagram measured while maintaining an optimal transconductance of the sensor dot found by a Bayesian optimiser.

The gate voltage range within which this compensation approach can be applied is limited by the failure of the linearity assumption underlying the definition of virtual gates. The effectiveness of the compensation therefore reduces as the device gate voltages are swept far away from the gate voltage location at which the compensated gates are calibrated. In our approach, recalibrating the compensated gates takes approximately ten seconds, allowing for the expansion of the gate voltage range in which compensation can be applied. The effect of charge switches and gate voltage drifts that change the sensitivity of the rf-QD can be overcome by integrating a closed-

loop feedback system for small measurement windows, such as those demonstrated in Refs. [76, 98].

The approach we present in this work naturally scales to device architectures with multiple sensors. The ac compensation can also be used for any type of fast 2D measurement, such as those involving interlaced pulse sequences. The algorithm is fully automated and requires no human intervention, and is therefore suitable for use in combination with other automated tuning procedures. This may combine well with proposals to combine data from multiple charge sensors [59].

This algorithm proved useful for several further experiments due to its ease of use and flexibility. However, there are several problems with it. The largest is that this is an open-loop solution, based on a linear approximation of a non-linear system. Thus incorrect measurement of the γ values will result in poor performance. In general I found the methods I proposed to be noise-robust and accurate.

Similar approaches to creating compensated virtual gates exist in the literature, but with no comment on how these compensated gates were generated [121]. In this work, the authors present a single set of calibrated compensated virtual gates, likely because they create the gates manually. This is very time-consuming, and does not allow for exploration of the device gate voltage space using the charge sensor, since inherent non-linearities in the system will reduce the accuracy of the virtual gates as the gate voltages are changed from the calibration position. As with the previous chapter, developing a bench-marking procedure through which to test the objective quality of compensation algorithms is an important, but missing, piece of work. This is hard in practice because of the differences in types of charge sensor, device architecture, and measurement protocol. I expand on this concept in Chapter 8.

5.4.1 Further work

Further work on this subject could look at integrating feedback into the algorithm, such that the open-loop errors are handled. My proposition is similar to how star guidescope trackers operate. There is an open-loop element that does the majority of the work, with an additional feedback loop over the top to tidy up any errors in this part of the algorithm. This would allow finer-tuning of the feedback loop's parameters. This requires fairly complex experimental setup, because some of the compensation needs to be dc, and thus go through the dc lines; some of the compensation is faster and requires transmission through the high-frequency lines. Coordinating this between different instruments is also not straightforward.

Hader *et al.* have also created an algorithm for evaluating the quality of charge sensing peaks [122]. In this work, they look to minimise the noise present in charge sensing measurements. Their routine is significantly more involved than my approach, including parameter estimation, therefore not lending itself to the tuning end of the operating spectrum. It would be interesting to compare their approach, which includes noise models for estimating the quality of peaks, to my variance-based approach. If the approach of Ref. [122] turns out to be significantly better for selecting peaks, it could easily be integrated into the rest of the algorithm presented here.

For larger arrays, it may be of interest to speed up the algorithm. I propose a method similar to the one I gave in the main text based on the finite differences gradient of the charge sensor Coulomb peak. To directly obtain the ratio of strengths between the sensor gate and a given device gate, sinusoidal signals are sent to each of these gates with a π phase difference. The amplitude of the sensor drive is fixed, while the amplitude of the device gate drive is swept. At the point where the contributions from both gates are the same, there will be a minima in the variance of the reflectometry

signal from the sensor. Alternatively, rather than attempting to cancel out the applied signal, one could measure the effect and perform a Fourier transform. The amplitude of the peak at the drive frequency would give the α values.

Chapter 6

Towards automated fine tuning of quantum dot devices in semiconductor heterostructures

In this chapter, I discuss some of the techniques that I developed to be used in a fully-automated qubit tuning pipeline for electrostatically-defined quantum dots using radio-frequency charge sensor measurements. This chapter contains several ideas for further investigation. I worked on this problem in parallel with another student in my group, Jonas Schuff, who published his excellent results to ArXiv around the time of my writing this chapter [123]. That work used dc transport measurements in a nanowire device, whereas my contributions in this chapter use reflectometry charge sensor measurements in a silicon/germanium heterostructure device. The work on this chapter also expands on that in Ref. [83], but without the reliance on fitting models.

6.1 Introduction

We control and measure singlet-triplet qubits (and other spin qubits) by tunneling spins between dots in a double dot array. By adjusting the relative energy levels of the two quantum dots, known as the detuning (ϵ), we can shuttle charges between them. Points in voltage space at which the energy levels of the two dots are equal ($\epsilon = 0$) are known as inter-dot transitions. In the literature, experimentalists usually seek to operate qubits at the transition between the $(2, 0)$ and $(1, 1)$ states [61]. Depending on the implementation, using this actual transition may improve the qubit's noise performance [2], but qubits have been encoded in quantum dots containing more than a single hole [2, 124] or electron [125] with good results.

Regardless of the number of spins in the dot, identifying the voltage location of an inter-dot transition is the first step in the fine-tuning process. In Chapter 4, we identified double dot features using a score function based on the Fourier transform. This score function identifies the large features of a double dot, but is not helpful in identifying the exact location of inter-dot transitions. For a fully-automated qubit tuning pipeline, we need an automated procedure for identifying this feature. In Ref. [67], Tim Botzem *et al.* propose fitting a simple physical model of the system to data. Similarly, in Ref. [83], the authors fit an inter-dot transition by template matching a geometric model, using the Hough transform and related processing tools. The package `QTT` [126] has a function based on similar concepts to these approaches. A common issue with all fitting methods is that you can always fit a model to some data, regardless of what features the data contains. The fit may be nonsensical, but this is hard to quantify. Additionally, these methods are slow, taking several seconds per measurement. They are useful for extracting parameters from a measurement of an inter-dot transition, but not particularly helpful for quickly identifying them in a tuning algorithm.

At an inter-dot transition, the target is to tune the system into a regime in which we observe Pauli spin blockade (PSB), since this phenomenon allows us to read out the spin state of the system via spin-to-charge conversion [56, 103]. In principle, PSB should occur at transitions between $(N_{\text{even}}, N_{\text{even}}) \iff (N_{\text{odd}}, N_{\text{odd}})$ charge states. In practice, PSB may not be present in all of these transitions. The confinement potential of the dot array needs to be tuned such that the various tunnel couplings allow for PSB to occur.

In this chapter, I present a toolbox of methods that, with some further integration, should allow for a fully-automated procedure to identify PSB in a gate-defined heterostructure device using only fast radio-frequency reflectometry charge sensor measurements. These tools comprise methods for identifying inter-dot transitions, maintaining these inter-dot transitions as other device parameters are changed, and methods for improving signal-to-noise ratios for spin measurements. In combination with the algorithms from the previous chapters, this workflow should allow for the full tuning of a device from grounded to PSB using only reflectometry, which would be a significant step forward for a fast (and therefore scalable) tuning process.

6.1.1 Experiment

I took the measurements in this chapter on a double quantum dot dot defined in a silicon/germanium heterostructure, similar to those used in Chapters 4 and 5 and nominally identical to that used in Ref. [43]. I measured in the double dot regime defined by barrier gates V_9, V_{11} and V_{13}, with plunger gates V_{10} and V_{12}. I applied fast pulses where relevant to V_{10} to control the right quantum dot and V_{13} to control the left. An image of a similar device with the gates labelled as they were set up in this experiment is given in Fig. 6.2.1 **a)**.

I measured the device using a charge sensor defined by barrier gates V_5 and V_7, and

plunger gate V_6. I probed the charge sensor via radio-frequency reflectometry at a frequency of 114.9 MHz. Gate V_8 split the sensor from the main device. The setup is effectively identical to that shown in the previous chapters.

6.2 Inter-dot transition identification

I begin by suggesting an approach for identifying the exact location of an inter-dot transition within the double dot stability diagram. This is a simple procedure for the human experimenter, but has proven to be a challenging process to automate. My suggestion begins with the statement that four different charge states exist in the voltage space around an inter-dot transition. Searching for the presence of four different charge states is therefore a natural way to identify an inter-dot transition.

6.2.1 Stability diagram histogram fitting

In this section, I assess the viability of identifying inter-dot transitions using Gaussian mixture models.

Histograms of charge stability diagrams exhibit a different number of modes depending on how many charge states are present. By using accurate charge sensor compensation (Chapter 5, Section 5.2.3), the number of modes in the histogram is equal to the number of charge states (Fig. 6.2.1 **d)**).

There are several ways to count the number of modes in a histogram, and all are fairly unreliable. The pixel values within each charge stability region, if perfectly compensated, should present as Gaussian. One approach would be to fit a Gaussian mixture model (GMM). However, GMM fitting algorithms require the parameter k, which is the number of Gaussians in the mixture. Further algorithms exist to assess which value of k best represents the data, but these involve fitting multiple GMMs,

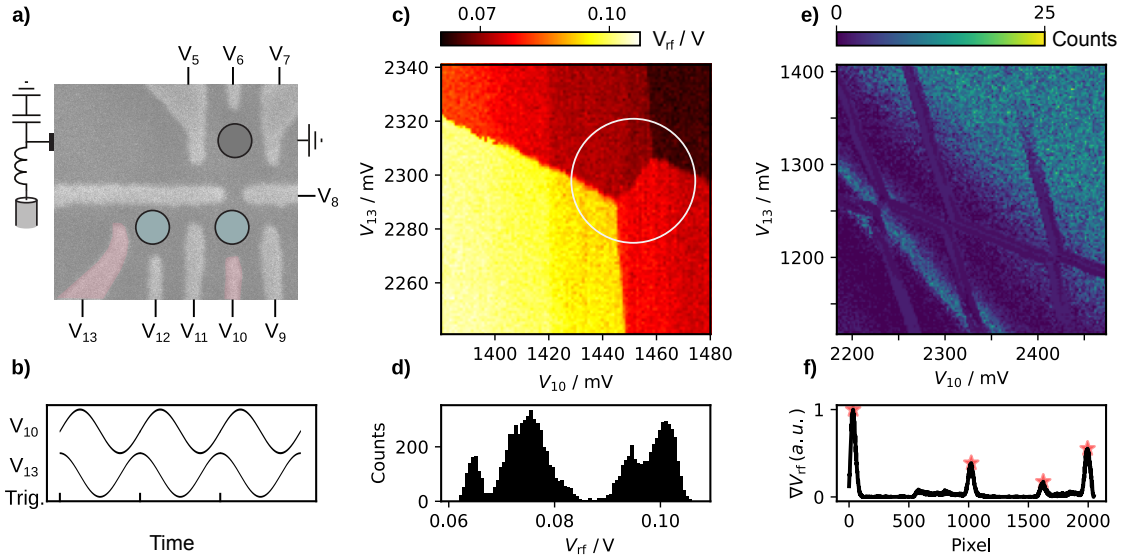


Figure 6.2.1: **a)** An image of a similar device to that used in these experiments. The array dots are in blue and the sensor dot is in grey. Fast gates are marked with purple highlighting. Note that in the real device, the splitter gate did not contain a gap and consisted of a single, continuous antenna. **b)** Sinusoidal and cosinusoidal pulses are applied to the labelled gates to create the circular pulse as demonstrated in c). **c)** A charge stability diagram taken using the charge sensor compensation algorithms developed in Chapter 5. The white circle represents the trace taken by applying a sinusoidal signal to V_{10} and a cosinusoidal signal to V_{13}, as described in the main text. **d)** The histogram of the data in c) shows four peaks. **e)** A map of the device's charge stability diagram, with each pixel corresponding to the number of peaks in the derivative of the measured time trace taken whilst pulsing with the circular pulse as indicated in c). **f)** The derivative of the time trace measured whilst pulsing around an inter-dot transition similar to the white circle in c).

each time increasing the number of components, and using an error estimator such as the Akaike information criterion to determine the most likely k value. This method does not scale well since we need to try all values of k up to n where n is the number of dots in the array.

Since

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}, \tag{6.2.1}$$

this scales on the order of n^2. If we know k, we can use mature GMM algorithms to

accurately fit the data. However, our goal is to calculate k, so GMMs are not the right solution.

6.2.2 Circular measurements for transition identification

The two-dimensional stability diagram is useful because we are humans looking at a two-dimensional computer screen. For the future of automated tuning, and with arrays of more than two dots, measuring the full stability diagram will become a bottleneck. We can significantly reduce the number of individual measurements required and the complexity of the pulses by measuring the device as we apply a sinusoidal pulse to one dot's plunger gate and a cosinusoidal pulse to the other's. The resulting pulse traces out a circle in the underlying charge stability diagram (Fig. 6.2.1 **a)**). A waveform is also applied to the charge sensor's plunger gate following the process outlined in the ac compensation scheme from Chapter 5.

I generated this measurement by applying a sinusoidal pulse to V_{13} and a cosinusoidal pulse to V_{10}. I programmed the Alazar (acquisition) card to measure for the same length of time as a single period of this circular trace, such that one measurement corresponds to one complete loop.

The radius of the circular pulse can be scaled by changing the amplitude of the output channels on the signal generator. Measuring a smaller circle is akin to zooming in on the features around a point in dc voltage space. Depending on the radius of the circle (semi-amplitude of the pulses), we can search for features of differing length scales in the device's underlying charge stability landscape. It is important to note that the frequency of the pulses must be slower than the tunneling rates of the transitions. Since the pulses are single frequencies, they they do not require filter compensation. The only hardware limitation is that they are above the high-pass filter cutoff frequency.

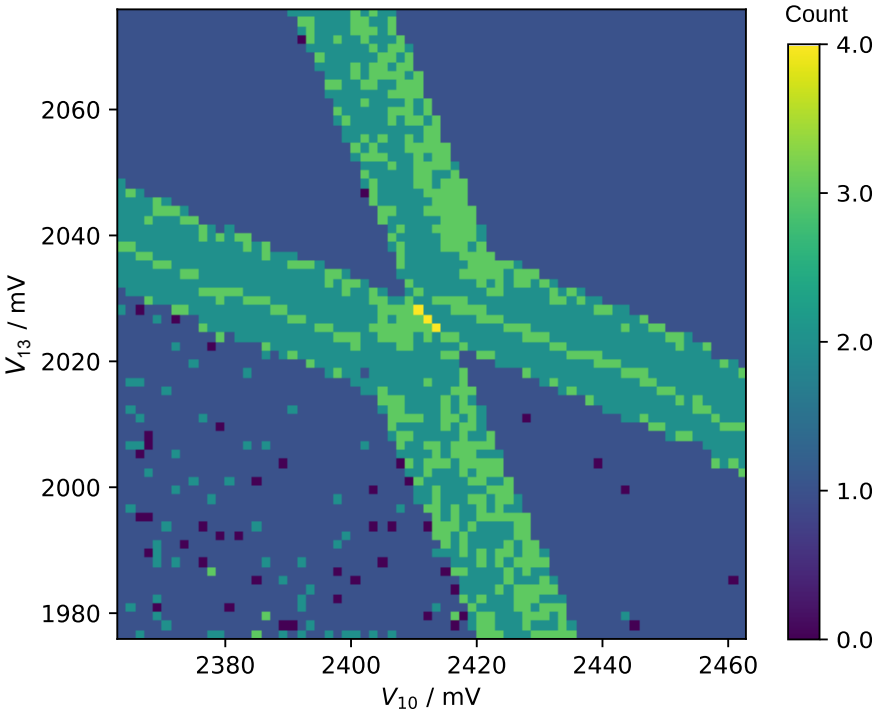


Figure 6.2.2: Counts of the number of peaks in the gradient of the time trace taken whilst pulsing in a circular manner as described in the main text. The approach finds the centre of the inter-dot transition accurately (counts = 4).

Counting transitions via derivative peaks

We can use these measurements to identify the number of charge transitions inside the area traced out by the circular pulse. M approach identifies the number of voltage states by counting the peaks present in the derivative of the circular measurement.

As the circle pulse crosses a charge transition, the sensor responds to the reconfigured charges and we measure a change in sensor output. Rather than looking at the histogram of the data, we can simply count the number of reconfigurations in the sensor and take this to be the number of charge transitions in the measured region. The absolute gradient of the measured time trace accurately captures this signal. We can count the number of peaks, N_{peaks}, in this derivative time trace using the `scipy.signal.find_peaks` peak-finding algorithm (Fig. 6.2.1 **f)**).

To count N_{peaks}, I take the direct time trace, averaged over 50 circle measurements,

and smooth the data using a Gaussian blur to remove some of the high-frequency noise. I then take the numerical absolute gradient and normalise the result, and then search for the peaks in this processed signal.

6.2.3 Identifying transitions

By counting the number of peaks in the gradient of the circular measurement, we can clearly identify charge transitions (Fig. 6.2.1 **c)**). However, the result is very noisy because the `find_peaks` algorithm has several parameters that need optimising. Measurements of pure noise still return a score because noise peaks still look like peaks to a peak-finding algorithm when the hyperparameters have not been well tuned. This is the cause of the noise in Fig. 6.2.1 **c)**. Note that $\max(N_{\text{peaks}})$ is approximately 25, when in principle the maximum should be four. However, also note that in regions where the charge transitions are present, the method is reliable. In hindsight, the reason for this noise is the normalisation of the derivative trace. Even in a signal containing pure noise, normalising this signal results in values between 0 and 1, making a peak height threshold useless. The peak detection worked using a threshold and a minimum prominence, but these parameters need more accurate tuning for a more reliable implementation of this approach.

In summary, applying a circular pulse with a radius calibrated to the voltage length scales of the device, we can identify inter-dot transitions by searching the voltage space for regions where the peak count is equal to four (Fig. 6.2.2). Calibrating the voltage length scales can be done by using a method similar to that described in Section 3.3.4.

6.2.4 Future experimental direction

This algorithm provides the basis for a method to follow transitions by finding regions in voltage space where $N_{\text{peaks}} = 2$. We can use this to follow a given transition until the algorithm identifies an inter-dot transition by $N_{\text{peaks}} = 4$. If we want to accurately identify the exact voltage locations of each triple point, we can search the space for which $N_{\text{peaks}} = 4$ using a circular pulse of a smaller radius, searching for the point at which $N_{\text{peaks}} = 3$. This corresponds to one of the two triple points that make up an inter-dot transition.

6.3 Maintaining inter-dot position

After identifying a suitable transition, we want to change the experiment's parameters whilst searching for a feature that identifies a qubit, such as oscillations between the two qubit states. The first step in this procedure is to search for Pauli spin blockade. In this section, I develop two ideas for maintaining the inter-dot transition at a fixed point in gate voltage space whilst changing other gate values, such that we can change the experiment's parameters (e.g. tunnel rates) whilst continuing to measure a single transition.

Varying the device's gate voltages to induce changes in the experiment's parameters moves the dc point relative to our 2D measurement window due to crosstalk between all gates and all dots in the array. I propose to use the position of the transition lines to quickly re-centre the inter-dot transition during sweeps on the other device gates.

6.3.1 Control using charge transition points

After the procedure outlined in Section 6.2, the device is tuned to an inter-dot charge transition. To begin with, we need to quantify the position of the inter-dot transition

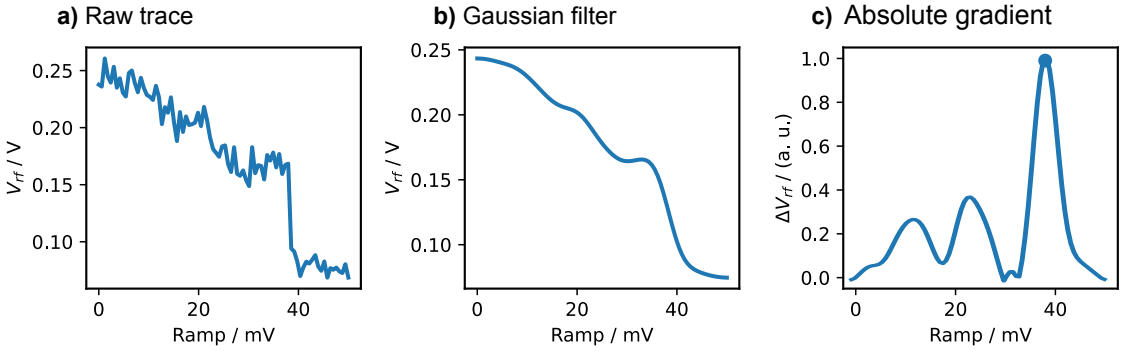


Figure 6.3.1: Processing steps for the transition identification. **a)** is the raw trace. The trace is **b)** filtered via a Gaussian (low-pass) filter before **c)** we extract the normalised absolute gradient. The tallest peak in this signal, identified using a peak-finding software routine and identified on the plot by a blue marker, corresponds to the voltage location of the charge transition.

such that we can re-centre it after changing gate voltages that define our double dot.

Identifying transitions

To make the visualisation of this process more straightforward, I start with a stability diagram measurement, but note that the following algorithm does not require the measurement of a full stability diagram, just the four traces described shortly. The size of the measurement is device-dependent; we require at least one transition in each dot. For this device, I worked with measurements of $50\,\mathrm{mV} \times 50\,\mathrm{mV}$. I select two locations within gate voltage space, P_1 and P_2 (Fig. 6.3.2 **a)**). From each of these points, I extract two traces from the 2D measurement: one horizontally and one vertically, resulting in four traces (Fig. 6.3.2 **a)**, **b)**).

Next, I smooth the traces by applying a Gaussian (low-pass) filter (Fig. 6.3.1 **b)**). I then take the absolute gradients of the processed traces and search for the tallest peak within them (Fig. 6.3.1 **c)**), once more using `scipy`'s `find_peaks` algorithm. After this, I normalise ∇V_{rf} such that it contains values between 0 and 1.

Fig. 6.3.2 **d)** shows the results of performing the trace peak-finding routine for ten sets of P_1 and P_2. This shows that this algorithm accurately identifies the transitions

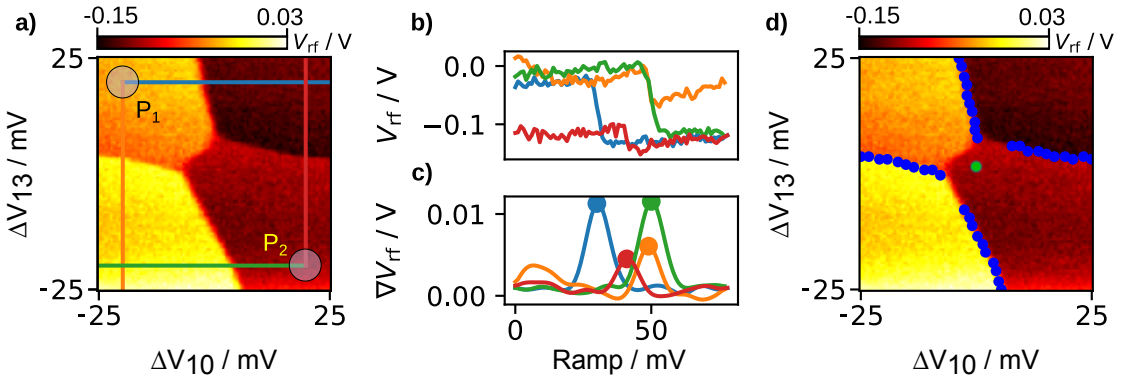


Figure 6.3.2: **a)** A charge transition with four traces highlighted, selected by taking orthogonal traces from points P_1 and P_2. **b)** The traces from a), colours matched. **c)** The absolute gradient of the traces in b), with peaks identified by the peak finding algorithm highlighted as circular blobs. **d)** The peak locations for ten sets of P_1 and P_2. The blue dots are peak locations (as in c)), and the green dot is the mean position of the blue dots.

within a stability diagram measurement.

Centring the transition

By the geometry of the charge stability diagram, the inter-dot transition lies at the geometric mean of the reservoir-dot transition lines. Thus taking the mean of the peaks in the traces gives us an approximate location of the inter-dot transition (Fig 6.3.2 **d)**, green dot). We can approximately centre the transition using this method, and search the area around this voltage coordinate for qubit or PSB features.

Standalone trace measurements

The approach as discussed so far relies on the fast 2D measurements that are ubiquitous in this thesis. For this algorithm, measuring the full 2D voltage space is not strictly necessary. For larger arrays, taking the full measurement is slow and counterproductive, since we cannot easily visualise measurements in dimensions greater than two. However, the trace methodology still applies, and we can measure the transition locations using ray traced measurements. Similar approaches exist in

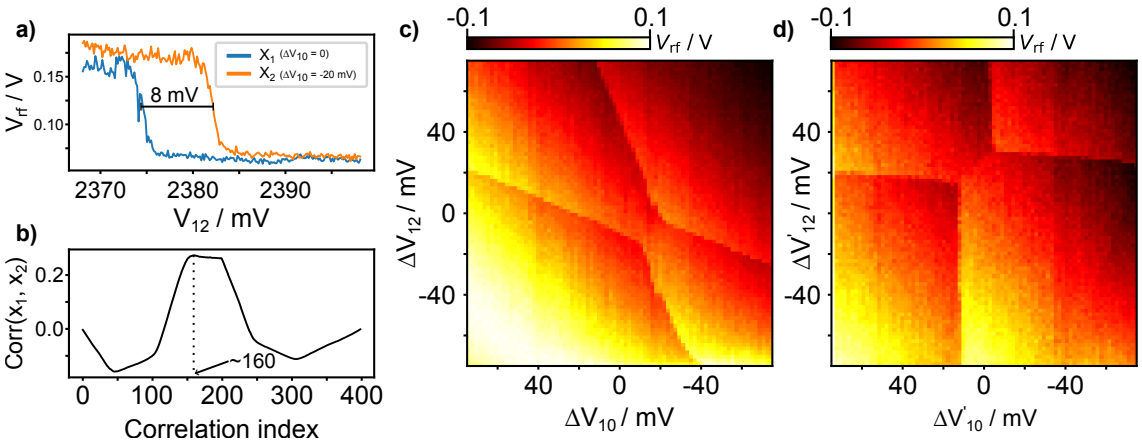


Figure 6.3.3: **a)** Traces taken by sweeping V_{12} over 30 mV, with the orange trace taken after moving V_{10} by -20 mV. **b)** The correlation of the two traces in a), with the `arg max` indicated by the blue dotted line. The argument is approximately 160; of a total of 400 pixels, this demonstrates the principle of this method. The x_2 trace is shifted by 8 mV after a 20 mV change in V_{10}. $8/20 = 160/400 = 0.4$, which is the α value representing the relative coupling of these gates. **c)** A stability diagram measurement taken using gates V_{10} and V_{12}; c.f. **d)**, taken using $V_{10}^{'}$ and $V_{12}^{'}$. These virtual gates were generated by taking the mean of 100 α values generated by 100 traces as described in Section 6.3.2.

the literature [127].

6.3.2 Virtual gates

Virtual gates are used widely in the literature to target specific device parameters [1, 83, 84, 89]. Several methods are used to extract the information required for their generation, including neural networks [128], model fitting [129], and classical signal processing [130]. From my discussions with members of other laboratories, it is also very common to manually define the virtual gates. In this chapter, I use the term virtual gate in its more traditional manner, meaning combinations of device gates that aim to target just a single device parameter, rather than the charge sensor compensation type.

As we saw in Chapter 5, the problem with virtual gates and similar concepts is that they are a linear solution to a non-linear problem. The operating range of virtual

gates is limited by this simplification. Since we cannot solve that issue, a good approach is to have a fast, reliable method that we can recalibrate easily.

I propose a technique similar to the idea that forms the basis of the charge sensor compensation algorithm in Chapter 5. For simplicity, I will discuss this in the case of creating two virtual plunger gates from the real plunger gates V_{10} and V_{12}. To create the virtual $V_{10}^{'}$ ($V_{12}^{'}$), we need to combine the real gate V_{10} (V_{12}) with a contribution from V_{12} (V_{10}) in the opposing direction, scaled by a coupling factor α. To calculate α for each case, we need to measure how much a change in V_{12} (V_{10}) affects the potential under V_{10} (V_{12}). The target is for $V_{10}^{'}$ ($V_{12}^{'}$) to not change the potential under $V_{12}^{'}$ ($V_{10}^{'}$).

Correlation between traces

For further simplicity, I will now only discuss generating the virtual plunger $V_{12}^{'}$ with a contribution from V_{10}. I begin by finding and centring a transition under V_{12}.

I take two traces of 30 mV in V_{12}, named x_1 and x_2, with x_2 measured after having incremented the value of V_{10} by $\Delta V = -20\,\text{mV}$. The value of ΔV depends on the lever arms of each gate but is on typically on the order of 10 mV. Due to this change, the transition in V_{12} is slightly higher (lower) in V_{12} voltage space since we have incremented V_{10} by a negative (positive) number (Fig. 6.3.3 **a)**). I calculate the voltage lag in the signals by the correlation $\text{Corr}(x_1, x_2)$. The `arg max` of this correlation result gives the lag in pixels; I convert it into a voltage lag V_{lag} (Fig. 6.3.3 **b)**) such that the ratio $V_{\text{lag}}/\Delta V$ is dimensionless. This ratio is the measured value of α for this gate pair. In the case of Fig. 6.3.3, the `arg max` in the correlation trace is 160 out of a total of 400 pixels, corresponding to a fractional lag of 0.4. We can confirm visually that this is accurate by observing that the -20 mV shift in V_{10} caused x_2 to lag x_1 by approximately 8 mV, which returns the same absolute ratio of

0.4.

Accuracy

In the noiseless (simulated) case, this method works with perfect accuracy every time. However, our experiment is rather noisy. By averaging values of α obtained via the correlation method, we can extract a more accurate coupling ratio than if we just made a single measurement. To calculate how many averages are necessary to obtain a satisfactory result, I took $1,000$ measurements of α between gates V_{10} and V_{12} to create $V_{12}^{'} = V_{12} - \alpha V_{10}$. The histogram of these results shows that they are distributed approximately normally around the mean (Fig. 6.3.4 **a)**).

From my dataset of $1,000$ measurements, I sampled n values for $n \in [1, 1 \times 10^3]$, generating 30 samples $\vec{\alpha}$ of size n for each n in this set. The mean, $\bar{\alpha}$, and standard deviation, std($\vec{\alpha}$), of these sub-samples as a function of n are given in Fig. 6.3.4 **b)**. It shows that with few measurements, the resulting α is inaccurate due to measurement noise. The standard deviation begins to saturate at sample size $n \approx 100$ (Fig 6.3.4 **c)**), which represents the lower bound for the number of averages before we are confident in the result. Fitting the standard deviation shows it has a functional form of $n^{-1/2}$, as expected by the central limit theorem.

I create virtual plunger gates using this method, with the result that the transitions are approximately orthogonal to each other when the virtual gates are swept (Fig. 6.3.3 **d)**). This serves as evidence that the virtual gates are targeting only the potential of a single dot each.

Using fast measurements

Taking over 100 traces using the DAC is a fairly slow approach, but does result in an accurate value for α. However, we can speed up the process using fast measurements.

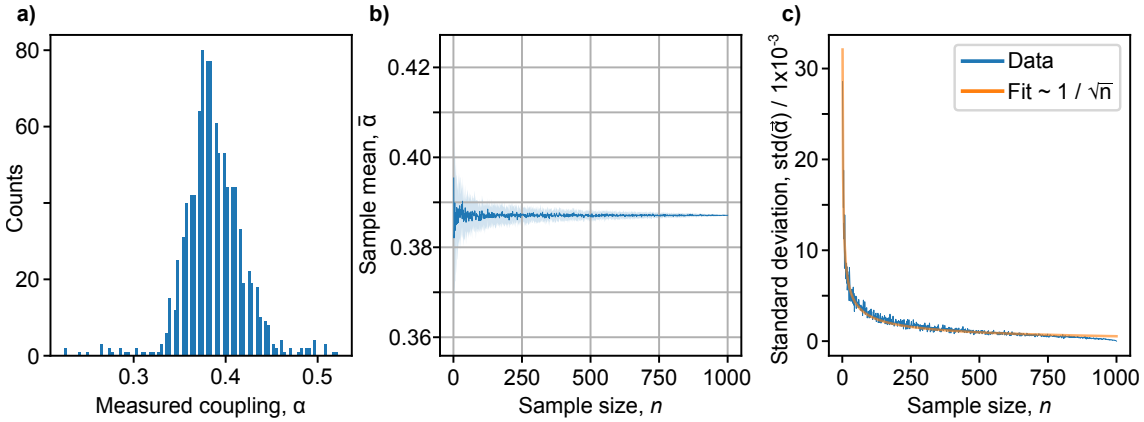


Figure 6.3.4: **a)** Histogram of 1,000 values of α measured over a single transition. The discretisation is due to the nature of the correlation function. **b)** The mean $\pm$ one standard deviation (highlighted) as a function of n, where n represents the size of a sub-sample from the dataset plotted in a). **c)** The standard deviation from b) as a function of n, with a fit of the form $B + A/\sqrt{n}$.

We can take these measurements by pulsing across the transition using an arbitrary waveform generator (AWG). The fast approach offers significant speedup; a full trace taken using the AWG takes approximately the same length of time as a single point using the DAC. However, it is only useful for gates that are connected to high-frequency lines, which is typically a limiting factor for this method. For gates that are wired for high-frequency control, this approach allows us to generate 100 measurements of the coupling strength in approximately 30 ms. This number of averages results in an accurate virtual gate almost instantaneously (Fig. 6.3.5).

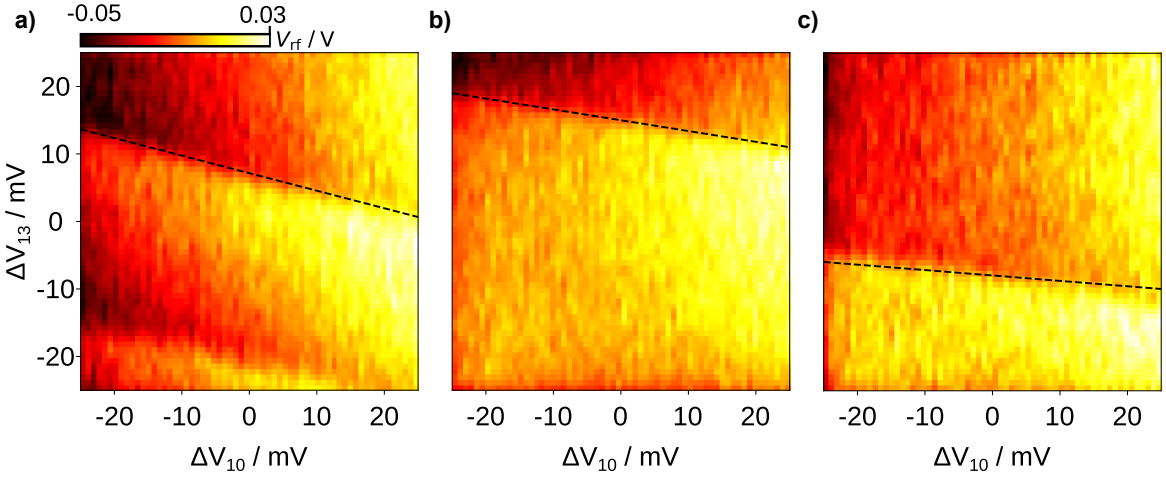


Figure 6.3.5: Transitions under V_{13} measured with the fast two-dimensional pulses as in the previous chapters. For each plot, I extract the virtual gate angle using the correlation method described in Section 6.3.2. The dashed black line represents the extracted gradient of the transition line dV_{13}/dV_{10}. This gradient is equal to the value of α, which is then used to create a virtual gate.

6.4 Identifying Pauli spin blockade

The signature of Pauli spin blockade (PSB) is to see a triangular shadow extending the $(1,1)$ (or equivalent) state into the $(2,0)$ upon the application of the appropriate pulse scheme (Fig. 6.4.1 **a)**).

6.4.1 PSB measurement sequence

The PSB measurement begins in the $(2,0)$ state. We then pulse back deep into the $(2,0)$ state to ensure we initialise with a singlet. We then pulse along the detuning axis, across the inter-dot transition and into the $(1,1)$ charge state, where the singlet and triplet spin states mix. We then pulse quickly back into $(2,0)$. If the spins are in the singlet state at the time of the return pulse, then we do not see spin blockade. However, if the spins are in the triplet configuration, we observe a spin filtering effect and tunnelling is blocked until the triplet state decays to a singlet.

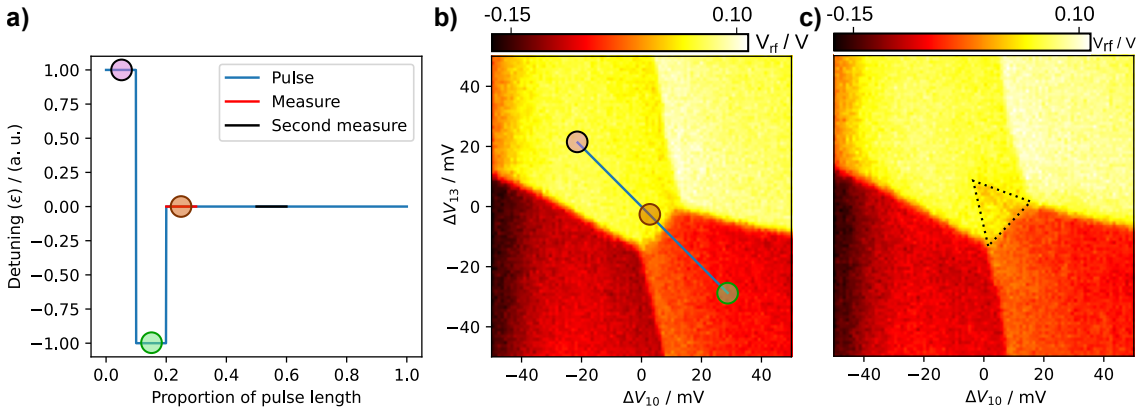


Figure 6.4.1: The Pauli spin blockade signature. **a)** The pulse scheme applied along the detuning axis, perpendicular to the inter-dot transition, that produces the spin blockade. The second measurement is subtracted from the first to remove the noise in the measurements described in the text. Note that in b) and c) I have not used the noise-reducing second measurement. **b)** An inter-dot transition with a schematic of the pulse, with points in gate voltage space labelled as in a). No pulse is applied during this measurement. **c)** The same transition as in b) but with the pulse from a) applied along the detuning axis via gates V_{10} and V_{13}. The spin blockade triangle is highlighted by the black dotted line as a visual aid.

6.4.2 Difference measurements

A useful realisation to increase the signal-to-noise ratio of the PSB pulse scheme (or for any pulse scheme) is that the background noise level is directly measurable by measuring the device state after no pulse has been applied (Fig. 6.4.1 **a)**). A modified pulse scheme combines the pulse-and-measure sequence with a no-pulse-and-measure, with equal pulse (or wait, for the no-pulse case) and measurement times. I configure the Alazar (data acquisition) card to measure twice as many points as there are in the output of the measurement. The measured data points consist of a measurement of the system after pulsing, followed by a measurement of the system after no pulse has been applied. I process the resulting data by slicing the array into pairs $(V_{\text{pulse}}, V_{\text{no pulse}})$. The returned data is $V_{\text{rf2}} = V_{\text{pulse}} - V_{\text{no pulse}}$.

The result is that the PSB triangle is significantly brighter than the surrounding region (Fig. 6.4.2 **b)**, **c)**). As a human experimenter, this makes the search for PSB

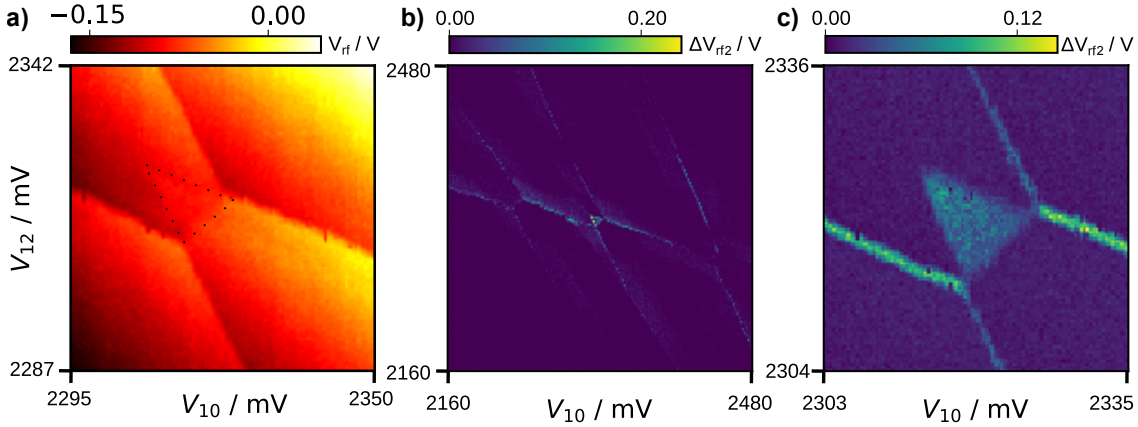


Figure 6.4.2: **a)** Charge stability diagram measured whilst pulsing as in Fig. 6.4.1, with the blockade region marked by the black dotted line. These measurements were taken with an out-of-plane magnetic field of 2 mT. **b)** Charge stability diagram measured as described in Section 6.4.2, using an acquisition of the system's noise to improve the signal-to-noise ratio. **c)** A single inter-dot transition measured using the double-acquisition method.

easier. The increased signal-to-noise ratio also improves the viability of using a computer vision algorithm for detecting this signature.

6.5 Conclusions

In this chapter, I presented several methods that are useful both for the human experimenter and for a fully-automated tuning pipeline. By introducing some novel measurement approaches, I wish to show that the standard approach to measuring quantum devices via charge stability diagrams is somewhat limited. By probing the system in creative ways, such as the circular measurements or by measuring the system's noise level directly after a signal measurement, we may be able to tune these devices more quickly.

The first set of algorithms presented in this chapter help to identify the location of an inter-dot transition using the novel circular measurement approach. This technique has shown promise in identifying that feature, but requires some further work to

improve its noise tolerance and handling. While I believe the approach is useful in its own right, it also opens the doors to further creative measurement approaches that move away from the charge stability diagram. Measurements based on ray traces through device gate voltage space have featured in the literature [127, 131], and are useful for our work in all-rf double dot tuning (Chapter 4).

The second set of algorithms are designed to maintain the position of this inter-dot transition such that the voltage configuration of the device can be adjusted to search for a regime in which a qubit can be encoded. By creating orthogonal virtual gates from the device gates, the job of the human experimenter or the automated tuning algorithm is made easier. There are related studies in the literature [83, 89]; my contribution improves the speed efficiency of generating these virtual gate sets. Since we know they are not globally valid in device gate voltage space, the calibration time is an important consideration. Virtual gate sets need to be recalibrated for each transition under study to ensure they are as accurate as possible.

The final method presented improves the signal-to-noise ratio of the Pauli spin blockade signature, making it more straightforward to identify this feature. As with the circular measurements, while this is already a useful measurement, the approach can be extended to any other measurement made in which we are measuring transient behaviour. We induce the phenomenon and measure how it affects the system, and then leave the system to rest and measure it once more. The difference in these measurements contains purely the signal due to the effect of interest.

By combining the algorithms presented in this chapter, and using a tool to navigate through the voltage space of the device, we can go from a double dot to Pauli spin blockade using automated methods using purely rf reflectometry measurements.

Chapter 7

Qubit readout optimisation

The methods proposed in Chapter 6 aid in taking a double quantum device, tuned using the approaches in Chapters 4 and 5, to the qubit regime. Repeatedly measuring the state of a qubit in single-shot mode can give us information about various metrics, such as the qubit's T_1 time. In this chapter, I derive a parametric probability distribution for the outcomes of repeated spin measurements, and demonstrate how we can use maximum likelihood estimation to infer the parameters of this distribution for a set of qubit readout data. With these parameters, we are able to tune the experiment to move us towards, for instance, a higher readout fidelity.

7.1 Introduction

In Chapter 4, we saw a *coarse tuning* procedure, where a device is tuned into a double dot regime. A double dot provides the right physical framework for hosting a spin qubit, but requires further *fine tuning* (Chapter 6). Once we have a well-defined two-level system (a qubit), there are further refinements we can make to the experiment to improve the operation of the qubit.

The qubit metrics that we are interested in optimising include its dephasing times, the readout fidelity, and the Rabi frequency, amongst others. In this chapter, I present an algorithm that fits data from repeated single-shot qubit measurements to a parametric model, derived in Section 7.2. From this fitting, we can extract parameters of interest such as the T_1 time and readout fidelity, which we can then use as a utility score to optimise. In principle, we can optimise these scores over any of the experimental parameters, such as device gate voltages, magnetic field strength and orientation [132], measurement integration time, etc.

The wide parameter space of gate-defined spin qubit devices has proven a significant challenge in the scale-up process. In the literature, parameters that lead to optimal qubit performance are known as *sweet spots* [44, 132, 133]. For instance, the coherence of a qubit can be improved by isolating it from noise sources [134], and is a tunable property [132]. Equally, the readout performance can be improved over experimental parameters. In this chapter, I explore how we can tune the system to optimise for improved qubit readout.

Readout fidelity

When we measure a qubit, we expect to see one of two values, representing the $|0\rangle$ state and the $|1\rangle$ state, with each existing at one of the poles of the Bloch sphere. Measurement noise and qubit decoherence means the histogram of repeated single-shot measurements is bimodal [135] with peaks at $V_{|0\rangle}$ and $V_{|1\rangle}$, representing the mean $|0\rangle$ and $|1\rangle$ state readout voltages[1] respectively.

For a single qubit measurement, we use the measured voltage to classify the state of the qubit by defining a threshold voltage V_T which separates these two states. The readout fidelity is a representation of how confident we can be in our classification.

[1] In principle this could be another unit, but it represents the output of the readout system, which is typically volts.

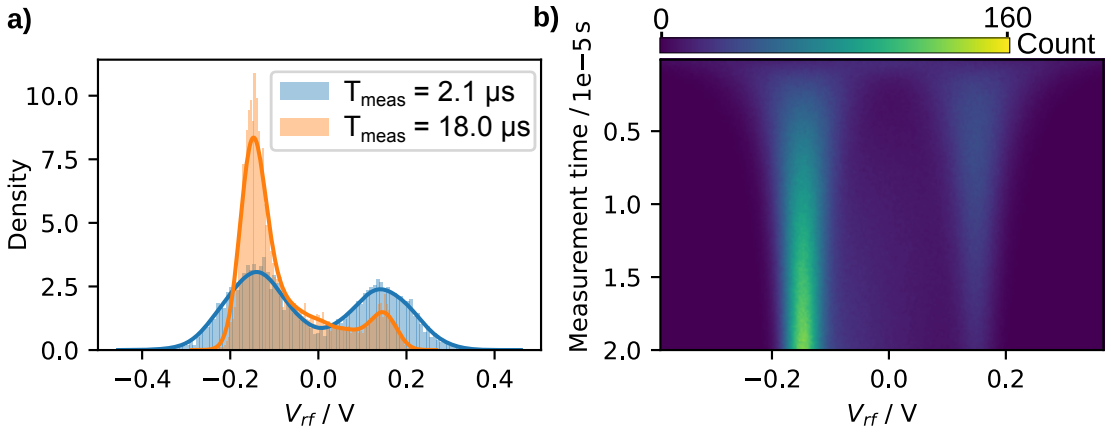


Figure 7.1.1: The effect of integration time on the readout value distribution. **a)** Longer integration times reduce the noise, but the longer duration means high-energy states are more likely to decay into low-energy states. **b)** As integration time gets longer (towards the bottom of the plot), we are more likely to measure a low energy state (lower V_{rf} value). Also, more measurements return a value between the two peaks as can be seen by the slight increase in density to the right side of the left peak at long measurement times. These measurements were generated with the simulator described in Section 7.4.

The individual state readout fidelities are given by

$$F_{|0\rangle} = 1 - \int_{V_T}^{\infty} n_{|0\rangle}(V)\, dV \tag{7.1.1}$$

$$F_{|1\rangle} = 1 - \int_{-\infty}^{V_T} n_{|1\rangle}(V)\, dV, \tag{7.1.2}$$

where $n_{|s\rangle}$ is the probability density of the measurements of state s [49, 135], with $s \in \{|0\rangle, |1\rangle\}$.

The total *visibility*[2] is given by

$$\text{Vis} = F_{|0\rangle} + F_{|1\rangle} - 1, \tag{7.1.3}$$

[2]In the literature, visibility is usually referred to as V. For clarity in this chapter, with so many references to voltages, I have chosen to use Vis.

which is equal to

$$1 - \int_{-\infty}^{\infty} \min \left[n_{|0\rangle}(V), n_{|1\rangle}(V) \right] dV. \tag{7.1.4}$$

With full knowledge of the distributions $n_{|0\rangle}$ and $n_{|1\rangle}$, we can accurately obtain the visibility, which is a key metric for the qubit's readout fidelity.

Integration time

The integration time of a single-shot measurement represents a trade-off between improving the signal-to-noise ratio and losing information due to qubit decoherence. Increasing the readout integration time past an optimal value τ_{opt} starts to harm rather than help, because the higher energy state has an increased chance of decaying within the measurement time, resulting in poor fidelities (Fig. 7.1.1). The value of τ_{opt} depends on the state decay time constant T_1 and the measurement noise inherent in the system [135, 136].

7.2 Derivation of signal probability distribution function

The qubit's state after measurement projection is one of its two basis states, $|0\rangle$ and $|1\rangle$. For a noiseless system in which the states do not decay or excite into one another, we expect the probability density function representing the expected outcome of n_{meas} single-shot spin measurements to look like two delta functions at the readout values $V_{|0\rangle}$ and $V_{|1\rangle}$ (Fig. 7.2.1 **a)**) (in the limit $n_{\text{meas}} \to \infty$).

As it happens, the noiseless, decay-less approximation is not a very accurate one. Firstly, the high energy $|1\rangle$ state may lose energy to its surroundings, thereby decaying

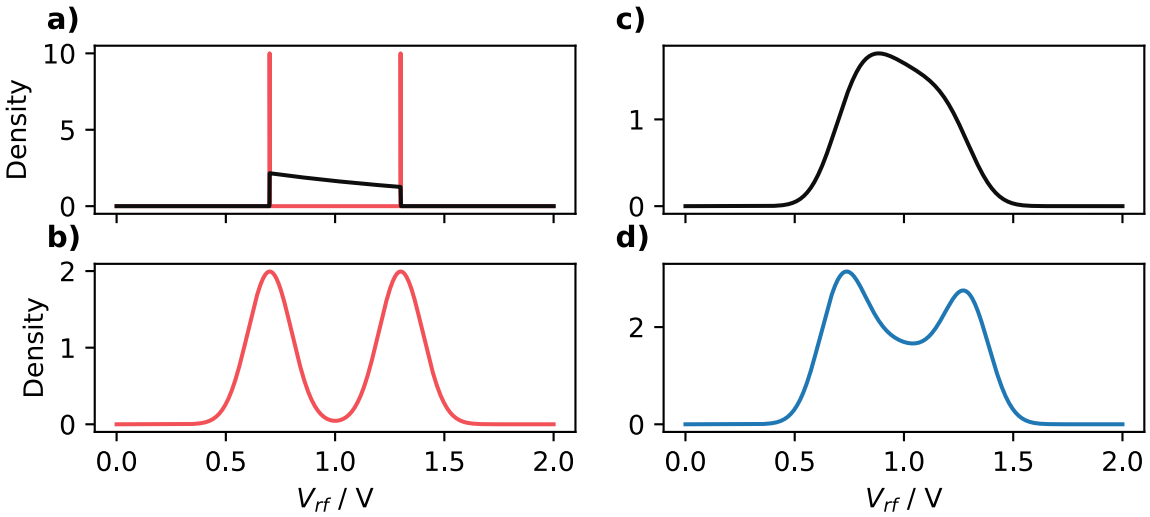


Figure 7.2.1: **a)** The noiseless case, with the delta functions at $V_{|0\rangle}$ and $V_{|1\rangle}$ in red and the exponential decay in black. **b)** A convolution of the delta functions with a Gaussian representing various noise sources results in a bimodal Gaussian mixture distribution function. **c)** The element missing from b) is the distribution of decay events from $V_{|1\rangle}$ to $V_{|0\rangle}$, meaning they have some contribution from both of the Gaussians in b). **d)** The convolution of the sum of the components in a) with the noise Gaussian increases the density between the two modes relative to the case in which we ignore the decayed samples (as in b)).

to the $|0\rangle$ state, with a time constant $T_{\downarrow}$. It is also possible for the low energy $|0\rangle$ state to absorb energy from its surroundings and become excited into the $|1\rangle$ state, which occurs at a rate $T_{\uparrow}$. Since $T_{\downarrow} \ll T_{\uparrow}$, the overall state decoherence occurs in the direction of $|1\rangle$ to $|0\rangle$, with an approximate decay rate of $T_{\downarrow}$; the timescale of the overall decoherence is the T_1 time. With this consideration included, our signal distribution S consists of two delta functions at $V_{|0\rangle}$ and $V_{|1\rangle}$, with an exponential decay of rate T_1 between the delta functions (Fig. 7.2.1 **a)**).

However, our measured value is not just made up of the signal, but also has a noise contribution. We can model our measurement distribution as $M = S + N$, where M is our measured distribution, S is the signal distribution (the two delta functions with exponential decay), and N is our noise distribution, which we can assume is Gaussian under the central limit theorem. To obtain the overall probability distribution from

the sum of independent variables, we convolve the individual probability distributions. Therefore our overall probability distribution is the convolution of a Gaussian with the sum of two delta functions and an exponential decay function from the higher energy state to the lower energy state (Fig. 7.2.1 **d)**).

In summary, the combined probability distribution from which a sample $\bar{x}$ in a single-shot qubit measurement is drawn is

$$\bar{x} \sim w_1 \mathcal{N}(V_{|0\rangle}, \sigma) + w_2 \mathcal{N}(V_{|1\rangle}, \sigma) + w_3 \int_{V_{|0\rangle}}^{V_{|1\rangle}} f(V; \lambda_{T_1}) * \mathcal{N}(V, \sigma) dV, \tag{7.2.1}$$

where $\lambda_{T_1} = 1/T_1$ is the exponential decay rate parameter, $f(V; \lambda_{T_1})$ is an exponential decay function over V with a decay rate λ_{T_1}, $\mathcal{N}$ is the normal distribution, and $\vec{w}$ is a weight vector containing the weights for each of the three probability densities w_1, w_2, and w_3. The convolution has an analytical solution based around the error function

$$\int_{V_{|0\rangle}}^{V_{|1\rangle}} f(V; \lambda_{T_1}) * \mathcal{N}(V, \sigma) dV = \frac{\tau P_{|1\rangle}}{\sqrt{2\pi}\Delta V} \cdot \exp\left(\frac{\tau}{\Delta V}(V_{|0\rangle} - V + \frac{\tau\sigma^2}{2\Delta V})\right) \cdot \left(\operatorname{erf}\left[\frac{\sigma\tau}{\Delta V\sqrt{2}} + \frac{V_{|1\rangle} - V}{\sigma\sqrt{2}}\right] - \operatorname{erf}\left[\frac{\sigma\tau}{\Delta V\sqrt{2}} + \frac{V_{|0\rangle} - V}{\sigma\sqrt{2}}\right]\right) \tag{7.2.2}$$

where τ is the integration time divided by the T_1 time, $P_{|1\rangle}$ is the probability of a high-energy $|1\rangle$ measurement, and $\Delta v = V_{|1\rangle} - V_{|0\rangle}$.

Dividing Eq. (7.2.2) by its integral normalises the area, such that the function is a probability distribution function. Plugging the normalised Eq. (7.2.2) into Eq. (7.2.1) gives the total analytical probability distribution function for repeated single-shot readout measurements.

7.2.1 Parametric model

Based on the derivation in Section 7.2, I present a parametric model with the following parameters:

1. $V_{|0\rangle}$: the noise-less $|0\rangle$ state readout voltage;
2. $V_{|1\rangle}$: the noise-less $|1\rangle$ state readout voltage;
3. σ: the Gaussian experimental noise;
4. w_1: the weight parameter for the low-energy state Gaussian;
5. w_2: the weight parameter for the high-energy state Gaussian;
6. w_3: the weight parameter for the decayed high-energy state (decaying exponential) distribution;
7. λ_{T_1}: the inverse of the state decay time T_1, or the exponential decay rate parameter.

The parametric model has the form

$$p(V) = w_1 \mathcal{N}(V; V_{|0\rangle}, \sigma) + w_2 \mathcal{N}(V; V_{|1\rangle}, \sigma) + w_3 f(V; V_{|0\rangle}, V_{|1\rangle}, \sigma, \lambda_{T_1}), \tag{7.2.3}$$

where $f(V; \cdot)$ is the probability distribution function based on the normalised Eq. (7.2.2). I also note here that the inference of $P_{|1\rangle}$ is unnecessary because implicitly $P_{|1\rangle} = (w_2 + w_3)/(w_1 + w_2 + w_3) = w_2 + w_3$. The total low-energy probability distribution is given by

$$n_{|0\rangle} = \mathcal{N}(V; V_{|0\rangle}, \sigma), \tag{7.2.4}$$

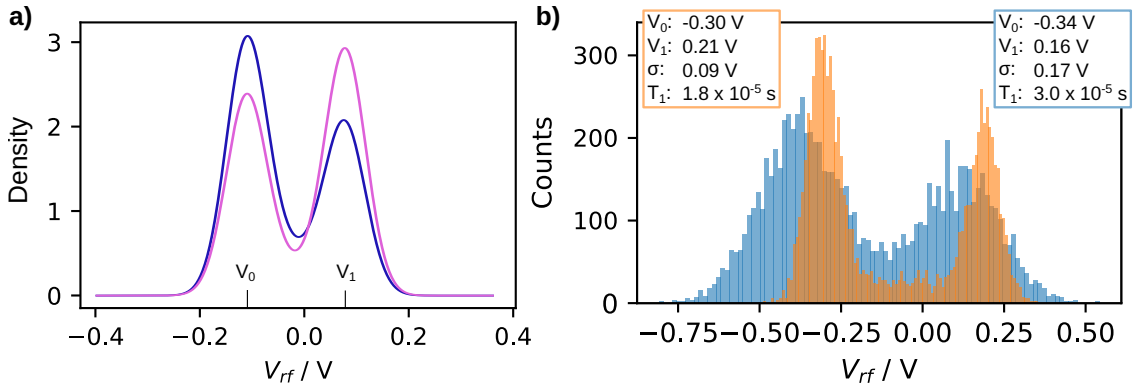


Figure 7.2.2: **a)** Two probability density functions generated by the parametric model derived in Section 7.2.1, with equal values for V_0 and V_1. The blue plot is simulated with a shorter T_1 time than the purple, resulting in a higher probability density near V_0. **b)** Two sets of simulated data generated by the simulator described in Section 7.4, with parameters labelled. The simulated integration time is 1 µs. The simulator accurately models the results of repeated single-shot qubit measurements.

while the total high-energy probability distribution is given by

$$n_{|1\rangle} = \frac{1}{w_2 + w_3} \left(w_2 \mathcal{N}(V; V_{|1\rangle}, \sigma) + w_3 f(V; V_{|0\rangle}, V_{|1\rangle}, \sigma, \lambda_{T_1}) \right) \tag{7.2.5}$$

In the following subsections, I show how I removed the need to optimise two of the parameters in the parametric model: the rate parameter λ_{T1}, and the noise parameter σ.

Rate parameter

We can estimate the rate parameter of the exponential decay that governs the decay of high-energy states into low-energy states, λ_{T1}, based on the properties of the exponential distribution. By analysing the cumulative distribution function (CDF) of the exponential distribution, we can show that the T_1 value is linked to the weights vector $\vec{w}$.

The exponential distribution's CDF is

$$P(X \leq x) = 1 - \exp^{-\lambda x}, \tag{7.2.6}$$

where λ is the rate parameter, X is the random variable and x is the value at which the CDF is evaluated. In our case, the rate parameter is λ_{T1}, the random variable is the decay time T_{decay}, and the value in question is the measurement integration time, τ_{meas}. Since w_2 and w_3 give us the weights of the non-decayed and decayed high-energy state samples respectively, the probability of a high-energy state sample decaying before the end of the measurement is

$$P(T_{\text{decay}} \leq \tau_{\text{meas}}) = \frac{w_3}{w_2 + w_3}. \tag{7.2.7}$$

Therefore

$$1 - \exp^{-\lambda_{T1}\tau_{\text{meas}}} = \frac{w_3}{w_2 + w_3}. \tag{7.2.8}$$

It is then trivial to show that

$$\lambda_{T1} = \frac{1}{\tau_{\text{meas}}} \ln\left(1 + \frac{w_3}{w_2}\right). \tag{7.2.9}$$

The rate parameter λ_{T_1} is therefore a function of two other model parameters, w_2 and w_3; as such it does not need to be included in the trainable parameters of the model.

Noise parameter

In this subsection, I detail how we can directly measure the noise using principal component analysis (PCA), meaning it does not have to be included in the training

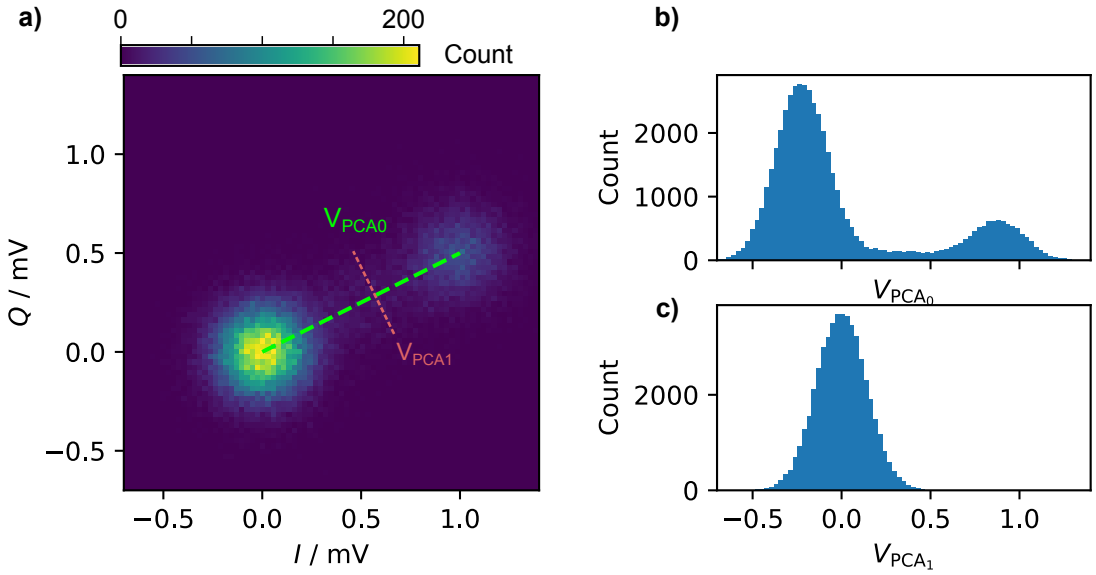


Figure 7.2.3: **a)** Simulated repeated single-shot data in the IQ plane. Performing PCA on this data returns two principal axes: the first, **b)**, contains data distributed as in the derivation from Section 7.2, while the second, **c)**, contains a Gaussian. The width of this Gaussian gives us a measure of the experimental noise which we can use as a fixed parameter to speed up the fitting of the parametric model.

of the parametric model using maximum likelihood estimation (Section 7.3). This process only works if we are measuring the qubit's state using reflectometry and obtain both the I and Q quadratures of the demodulated signal as in the previous chapters.

Experimentally, the demodulated qubit state data is returned to the lab PC in both the in-phase (I) and out-of-phase (Q) components (Fig 7.2.3 **a)**). The model I have derived uses just a one-dimensional representation of the readout data, because fitting one-dimensional data is faster and without loss of accuracy. This one-dimensional representation corresponds to the first principal component of the IQ data (Fig. 7.2.3 **b)**). In previous chapters, we have not used the projection onto the second principal component. However, we can use it here to estimate the measurement noise.

Due to the fact that a two-dimensional Gaussian is a separable function, we do not lose any information by considering the two orthogonal axes separately. Our two-

dimensional IQ data has two principal components: the first contains data distributed as derived in Section 7.2 (V_{PCA_0}, Fig. 7.2.3 **b)**), and the second is orthogonal to this axis (V_{PCA_1}, Fig. 7.2.3 **c)**). The only variance in V_{PCA_1} is due to the width of the noise Gaussian. The variance of this projection is equal to the measurement noise in our system, which is the parameter σ.

It is possible that this hypothesis may not always represent a realistic case. The noise Gaussian may have some covariance, or the data may be non-linearly skewed in the IQ plane. My hypothesis is that the data projection onto V_{PCA_1} contains only Gaussian measurement noise, which we can check using the Anderson-Darling test for normality. If V_{PCA_1} does not contain normally-distributed data, or if the demodulation setup does not return both I and Q signals, I revert to fitting the noise parameter along with the other parameters.

7.3 Maximum likelihood estimation

Maximum likelihood estimation (MLE) is a method for obtaining the distribution that most likely generated a data sample from a family of distribution functions. The distributions in this family are related because they are parametric. With MLE, we can obtain the parameters that create the distribution that is most likely to have generated the data.

We start with a parametric joint probability distribution function, governed by parameters $\theta = [\theta_1, \ldots, \theta_k]^T$. Over this parameter space $\boldsymbol{\Theta}$, there exists a set of possible distribution functions $\{f(\cdot;\theta) \mid \theta \in \boldsymbol{\Theta}\}$. With an accurate parametric model, we can find parameters that generate the function $\hat{f}$ that best represents our data. To obtain that distribution, we evaluate the joint density of the sample $\boldsymbol{x}$ as a function of the parameters θ:

$$\mathcal{L}(\theta) = \mathcal{L}(\theta; \boldsymbol{x}) = f(\boldsymbol{x}; \theta), \tag{7.3.1}$$

where $\mathcal{L}(\cdot)$ is known as the *likelihood function*.

Rather than looking at a probability distribution function generated by a fixed set of parameters θ_i and calculating the probability of obtaining a sample x, we reverse our perspective and look at the probability of obtaining a fixed sample x_i as a function of the parameters θ. Maximum likelihood estimation is the process of finding parameters θ that maximise the likelihood for a sample x_i,

$$\hat{\theta} = \arg\max_{\theta \in \boldsymbol{\Theta}} \mathcal{L}(\theta; \boldsymbol{x_i}). \tag{7.3.2}$$

As with any fitting procedure, there is guaranteed to be a $\hat{\theta}$, so long as the likelihood function $\mathcal{L}$ has a dependence on the parameters θ. Of course, this does not mean that the fitting is actually accurate. To obtain an accurate fit, we have to be sure that the distribution function set $\{f(\cdot; \theta) \,|\theta \in \boldsymbol{\Theta}\}$ models the system in a satisfactory manner [137].

Negative log likelihood

Rather than directly optimising the likelihood, it is conventional to minimise the negative log likelihood. There are two reasons for working with the logarithm of the likelihood function. The first is that it allows joint probabilities to be calculated via summation rather than multiplication, which is less computationally expensive. The second is that it makes it possible to deal with very small probabilities without their values falling below the floating point limit, which could cause an underflow error.

By convention, optimisers minimise cost functions, rather than maximise score functions. This is as trivial as it sounds. Thus rather than maximising the likelihood

function $\mathcal{L}$, we minimise the negative log likelihood, which I denote as NLL.

7.4 Simulator

To test the model and MLE-based fitting procedure, I wrote a simulator that generates a sample $\vec{V}$ of n_{meas} repeated single-shot qubit measurements. The concept is similar to the derivation of the parametric model from Section 7.2, so I will explain it briefly.

For a single output sample $\vec{V} \in \mathcal{R}^{n_{\text{meas}}}$, the simulator generates three vectors, each of size n_{meas}:

$$\vec{V_0} \in \mathcal{R}^{n_{\text{meas}}} \sim \mathcal{N}(\bar{V_{|0\rangle}}, \sigma^2) \tag{7.4.1}$$

$$\vec{V_1} \in \mathcal{R}^{n_{\text{meas}}} \sim \mathcal{N}(\bar{V_{|0\rangle}}, \sigma^2) \tag{7.4.2}$$

$$\vec{V_2} \in \mathcal{R}^{n_{\text{meas}}} \sim \mathcal{N}(\bar{V_{|1\rangle}}, \sigma^2), \tag{7.4.3}$$

where $\bar{V_{|0\rangle}}$ and $\bar{V_{|1\rangle}}$ are simulator parameters representing the low-energy and high-energy state readout voltage means, and σ^2 is the simulated measurement noise scale. These vectors represent the readout voltages, V_{rf}, of the three different qubit measurement cases from which we are sampling, namely a low-energy state ($\vec{V_0}$), a high-energy state ($\vec{V_1}$), and a high-energy state that has decayed into a low-energy state ($\vec{V_2}$).

Next, I generate a sample $\vec{T}_{\text{decay}}$ of size n_{meas}, which contains simulated decay times for each element V_i in $\vec{V}$. I generate these from an exponential distribution with a decay rate $\lambda_{T_1} = 1/T_1$:

$$\vec{T}_{\text{decay}} \in \mathcal{R}^{n_{\text{meas}}} \sim \text{Exp}\left(\lambda_{T_1}\right), \tag{7.4.4}$$

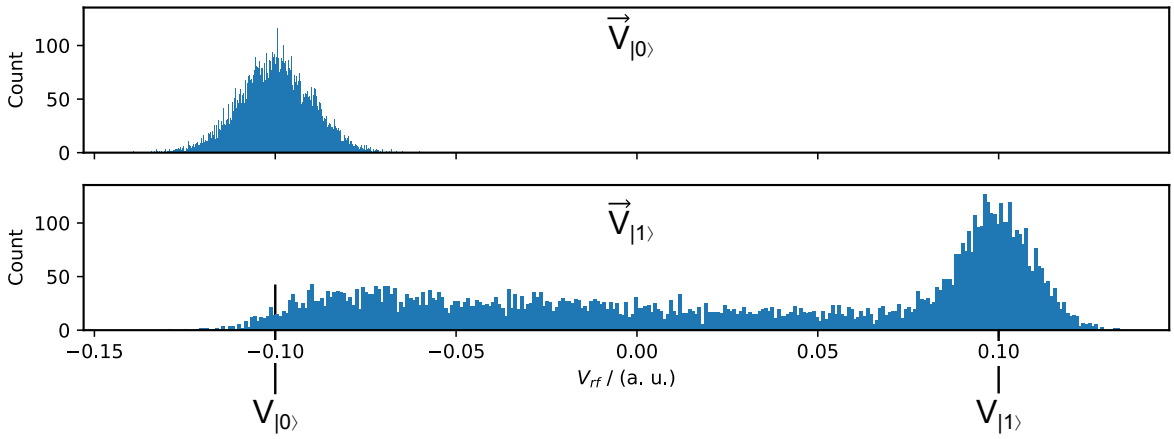


Figure 7.4.1: Distributions of $\vec{V}_{|0\rangle}$ (top) and $\vec{V}_{|1\rangle}$ (bottom) as generated by the simulator. The low-energy $\vec{V}_{|0\rangle}$ is simply a Gaussian centred on $\bar{V}_0$, while the high-energy $\vec{V}_{|1\rangle}$ state distribution has a most weight around the $\bar{V}_1$, but with a long tail stretching to $\bar{V}_0$ representing decayed states.

where $\mathrm{Exp}(\lambda_{T_1})$ is a probability distribution defined by the function

$$f(x;\lambda_{T_1}) = \begin{cases} \lambda_{T_1} e^{-\lambda_{T_1} x} & x \geq 0, \\ 0 & x < 0. \end{cases} \tag{7.4.5}$$

At this stage, the low-energy state measurements, V_1, do not need further processing. For the high-energy state measurements, we need to look at the simulated decay times and calculate how this decay affects the simulated measurement voltage.

Over the course of the measurement integration time τ_{meas}, if a sample decays from the $|1\rangle$ state to the $|0\rangle$ state in time T_{decay}, the integrated measurement outcome will be a voltage V_{decay} between $V_{|0\rangle}$ and $V_{|1\rangle}$, depending on the ratio $T_{\mathrm{decay}}/\tau_{\mathrm{meas}}$. I find these ratios by $\vec{r} = \vec{T}_{\mathrm{decay}}/\tau_{\mathrm{meas}}$, with the limit clipped at 1 representing a $|1\rangle$ state that did not decay during the measurement.

To get the final readout values of the states that started as $|1\rangle$, I perform the calculation

$$\vec{V}_{|1\rangle} = \vec{r}\vec{V}_2 + (\vec{1} - \vec{r})\vec{V}_1, \tag{7.4.6}$$

giving the final vector representing measurements of states that started the measurement as the high-energy $|1\rangle$ state. For the low-energy state, since there is no decay, the final vector representing measurements of that state is $\vec{V}_{|0\rangle} = \vec{V}_0$ (Fig. 7.4.1).

The result is two samples of size n_{meas}, representing measurements of a qubit in the low-energy state ($\vec{V}_{|0\rangle}$), and of a qubit that started the measurement in a high-energy state and decays with some probability based on its characteristic decay rate ($\vec{V}_{|1\rangle}$).

The T_1 cycle in a spin qubit experiment involves preparing a $|0\rangle$ state, evolving it for some time, and measuring. I generated such data by selecting data points from the two samples $\vec{V}_{|0\rangle}$ and $\vec{V}_{|1\rangle}$ by generating a third sample, $\vec{B} \in \mathcal{R}^{n_{\text{meas}}}$, which was drawn from the binomial distribution with a probability $P_{|1\rangle}$. This random sample dictates which of the $V_{|.\rangle}$ vectors to draw each element of the final sample from. Two samples generated in this way are shown in Fig. 7.2.2.

I mention for interest that by using different distributions to generate $\vec{B}$, it is possible to generate time-dependent measurements such as Rabi oscillations. For instance, rather than sampling from a binomial distribution, I introduced a time-dependent state probability

$$\vec{P}_{|1\rangle} = \frac{\sin(\vec{t}) + 1}{2}, \tag{7.4.7}$$

where t models the Rabi time axis. I generate a second random sample $\vec{T} \sim \text{Uniform}(0, 1)$ to represent a threshold. The points I take from the $\vec{V}_{|1\rangle}$ sample occur at the elements where $\vec{P}_{|1\rangle} > \vec{T}$; otherwise, I take from the $\vec{V}_{|0\rangle}$ sample.

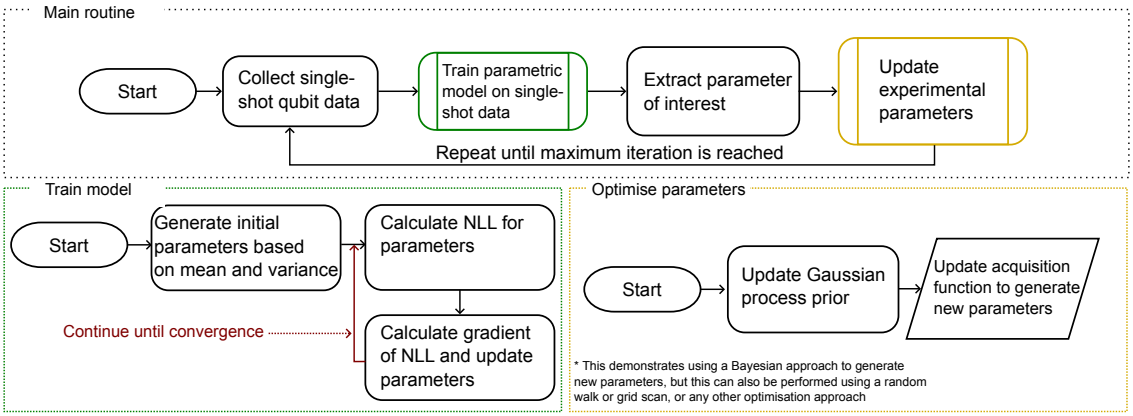


Figure 7.5.1: A flowchart representing the main steps of the algorithm. Note that the *optimise parameters* box depicts the use of Bayesian optimisation with a Gaussian process to update the experimental parameters, but any other approach could be used in its place. In the main text, I use a grid scan approach to demonstrate the utility of the algorithm.

7.5 Algorithm

In this section, I describe how I use the methods from the previous sections to create an algorithm for improving the readout fidelity of qubit experiments. The idea is to take repeated single-shot measurements, train a parametric model on those measurements, and use the parameters of the data to understand more about the experiment. We can use these parameters as score functions to optimise the experiment over any experimental variables. A flowchart of the algorithm, including a Bayesian approach to optimising the parameters, is given in Fig. 7.5.1.

7.5.1 Eliminating measurements without spin

Given a set of single-shot data, we first need to assess whether or not the data contain spin physics. If we do not see spin physics, we are not in a suitable regime for a qubit. However, a limiting factor of performing parametric fits is that there is always a way to fit data, no matter how poor the accuracy of the fit. Thus we must first model other possible cases and see if it is likely that the data originates from

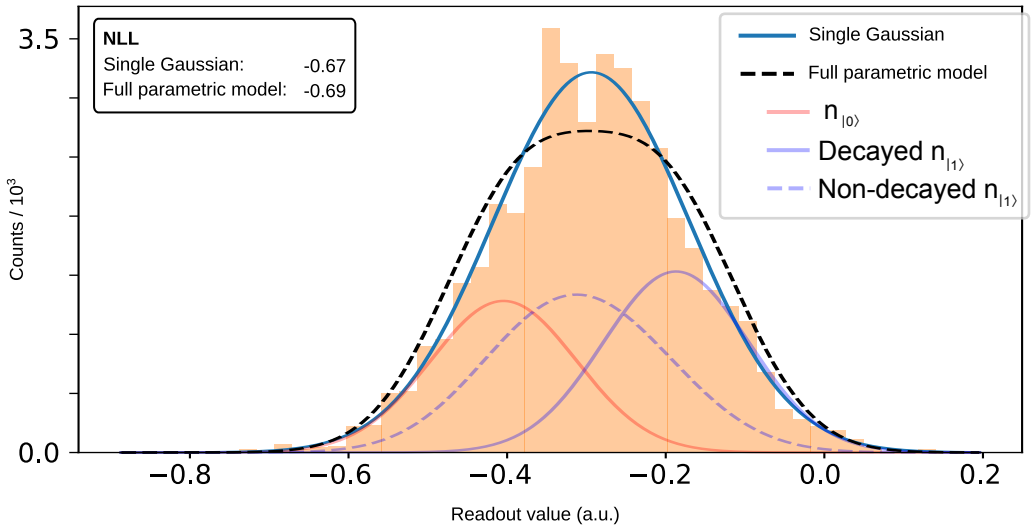


Figure 7.5.2: Data generated from a Gaussian distribution. I have trained both a single-Gaussian model and the full parametric model from Section 7.2.1. The fit of the parametric model has a lower negative log likelihood than the single Gaussian, even though the latter generated the data.

these cases before fitting to the parametric spin model.

In a regime where there is no spin physics, or if we measure just noise for any other reason, a histogram of the data generated by repeated single-shot measurements will be Gaussian. Fitting the full parametric model to this histogram does not result in a computational error, and in fact returns a lower negative log likelihood than the fit of a single Gaussian, suggesting it fits the data better (Fig.7.5.2). To avoid spurious results, we have to first ensure the data we are fitting are not normally distributed, which would imply that we did not measure spin physics. We cannot use the NLL metric for this, and instead test the data for normality using an Anderson-Darling statistical test.

Model selection

To detect whether the samples were generated by a single Gaussian or the parametric mixture model, I calculate the mean μ_{meas} and standard deviation σ_{meas} of the data and use this to generate a second sample set from a Gaussian with these parameters

$\mathcal{N}(\mu_{\text{meas}}, \sigma_{\text{meas}})$. I then compare the samples generated by the experiment (or simulation) to the samples generated from the single Gaussian using an Anderson-Darling test. This test is a modification of the Kolmogorov-Smirnov test that gives more weight to the tails of the distribution. This is beneficial because in regimes where we only have a small probability of loading a $|1\rangle$ state, we might see a distribution that is very similar to a Gaussian in the centre, but diverges from a Gaussian at the tails. This test reliably distinguishes between samples drawn from a single Gaussian (as in the case where we have no spin physics) and samples drawn from the parametric model.

If the samples are drawn from a single Gaussian, there is no need to run the MLE algorithm since we know that the samples are not generated by the parametric distribution. This saves considerable computation time because spin physics occurs only in a small window in device gate space. We save an optimisation loop every time we do not register spin physics by performing the Anderson-Darling test first.

7.5.2 MLE parameter inference

If we measure spin physics, the measured data are not Gaussian distributed, and so I fit the model derived in Section 7.2. I have several implementations of the model, including one in `JAX`, allowing for automatic differentiation and therefore gradient descent. I minimise the NLL until it converges, at which point I terminate the optimisation process and take the result.

With this trained model, I extract the parameters of interest from the data (Section 7.6.1). I can also rebuild the individual probability distributions, and from there calculate the optimal threshold voltage (V_T) and state visibility (Vis).

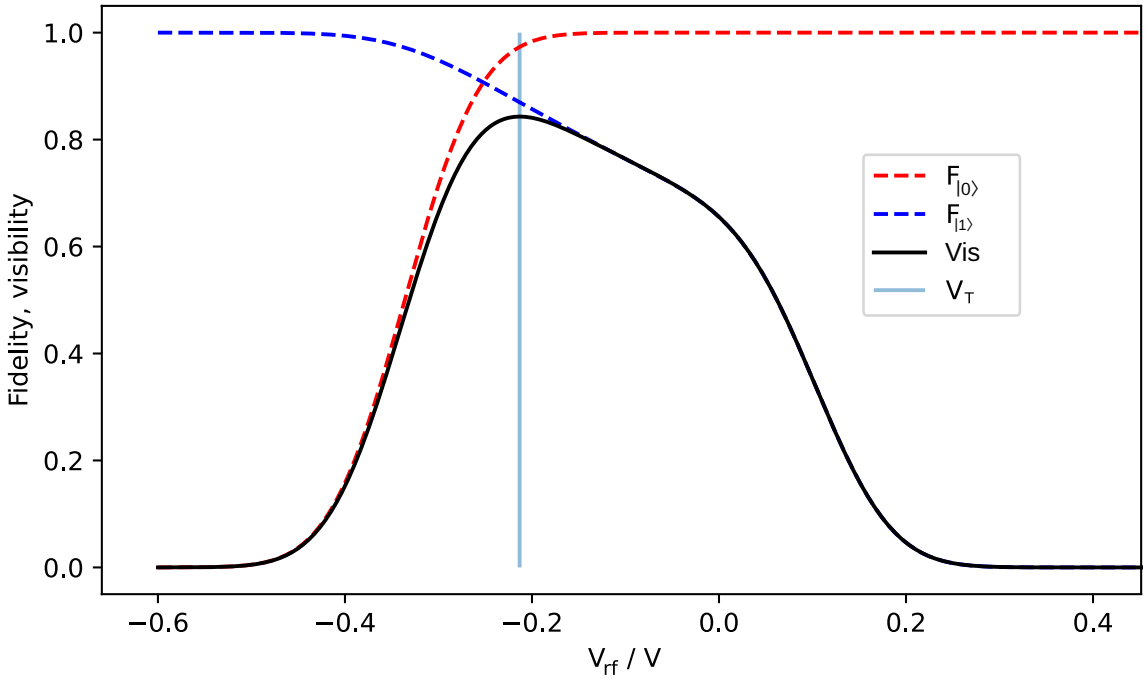


Figure 7.5.3: The visibility and individual fidelities as a function of V_T for a set of simulated data. Increasing the V_T towards the high-energy state readout voltage V_1 improves the fidelity of the low-energy $V_{|0\rangle}$ state, but dramatically worsens the fidelity of the $V_{|1\rangle}$ state. The optimal V_T represents a trade-off between these competing constraints.

7.5.3 Optimal threshold voltage

The optimal threshold voltage V_T is the one corresponding to the maximum visibility (Eq. (7.1.3)). In Appendix C, I show that this optimal V_T corresponds to the voltage $V_=$ at which the high-energy $n_{|1\rangle}$ probability distribution overlaps with the low-energy $n_{|0\rangle}$ distribution. Choosing a threshold voltage higher or lower than $V_=$ results in a worse visibility. The optimal threshold voltage is therefore $V_=$.

After training the model on data, we have access to both the $n_{|0\rangle}$ distribution and the $n_{|1\rangle}$ distribution, meaning we can extract the optimal V_T.

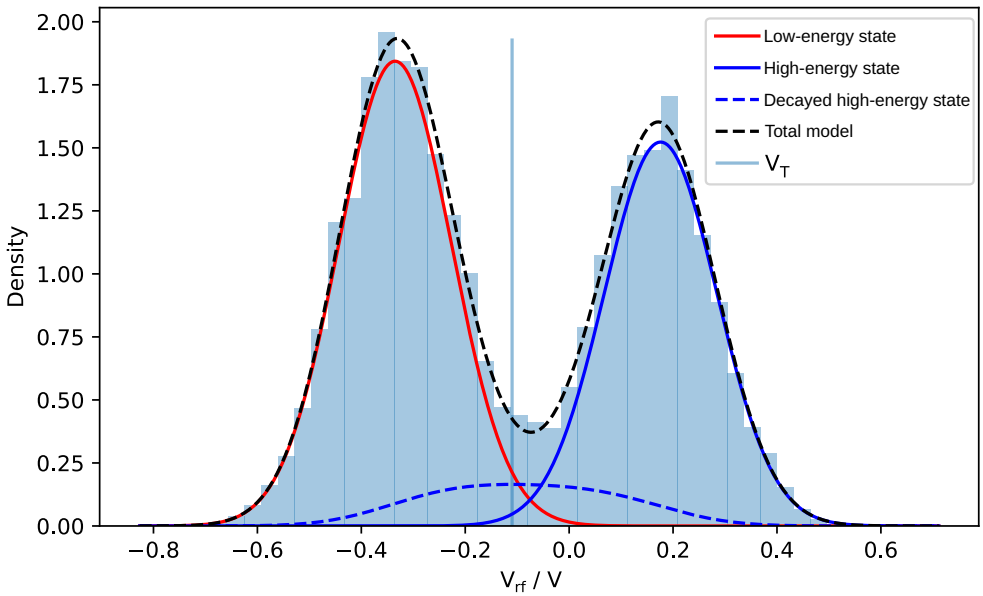


Figure 7.6.1: Example of a fit to a simulation of 8000 single-shot measurements. In this case, the integration time is simulated to be 1.0 µs and the T_1 time is 5.3 µs. The probability of loading a high-energy state is 0.5. The result is that the contribution from the decayed high-energy states is fairly small, and the difference between the low- and high-energy Gaussian weights ($w_1 = 0.5$ and $w_2 = 0.41$) is also fairly small.

7.6 Results

Unfortunately, by the time I wrote this algorithm, our precious spin qubit in Oxford had been blown up after a series of unfortunate events starting with an explosion at the Chemistry Department. I did not get a chance to try it out in Oxford, but I did get some experiment time during a brief collaboration with Ferdinand Kuemmeth's group at the Niels Bohr Institute, University of Copenhagen. I also used the simulator to test the fitting routine.

7.6.1 Simulator

I tested the accuracy of the algorithm on the simulator described in Section 7.4. Of course, there is nothing to optimise in this simulator, so the results are purely for

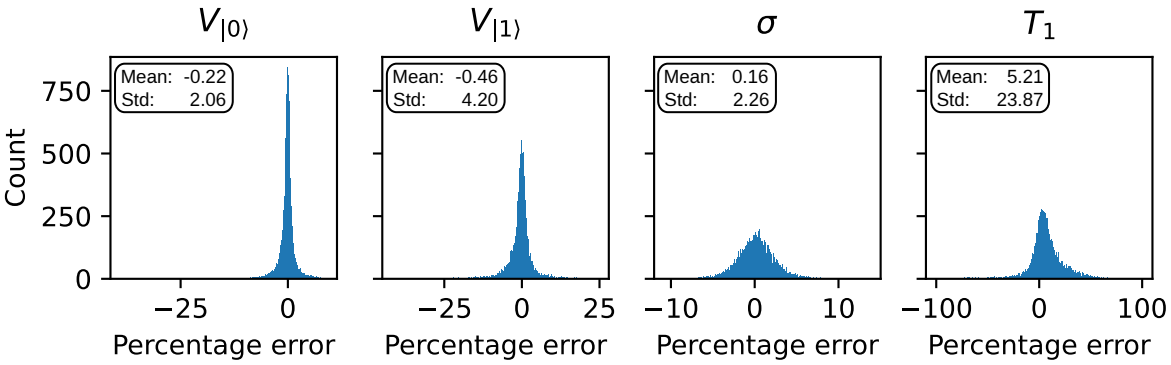


Figure 7.6.2: Histograms of percentage errors from 10,000 fits of simulated samples of size $n = 5000$ for the four parameters as titled.

validating the fitting procedure.

For these simulated results, I generated the data as if they had been processed by PCA, i.e. optimally projected onto a single dimension. The noise parameter is therefore inferred via maximum likelihood estimation rather than estimated via PCA using the noise parameter reduction from Section 7.2.1. For this analysis, I generated $10,000$ samples of $n = 5000$ data points. I then fitted the model to these data points to infer $V_{|0\rangle}$, $V_{|1\rangle}$, σ, and T_1.

Overall, the algorithm extracts the four parameters with reasonable accuracy (Fig. 7.6.2). The simulated results show that inferring $V_{|0\rangle}$, $V_{|1\rangle}$ and σ is an accurate procedure, with mean percentage errors (standard deviation) over the $10,000$ runs being $-0.22\,(2.06)\%$, $-0.46\,(4.20)\%$, and $0.16\,(2.26)\%$ respectively. However, T_1 is much harder to infer accurately. The mean percentage error is reasonably low, at 5.21%, but the standard deviation is 23.9%, which reduces confidence in the inferred value. The reason for this is that the effect of the T_1 time on the overall probability distribution is minor.

7.6.2 Spin qubit

In this section, I present measurements taken in collaboration with Torbjørn Rasmussen from Ferdinand Kuemmeth's group at the University of Copenhagen.

Device

The device measured in this subsection was a two-by-two singlet-triplet qubit array in gallium arsenide, as used in Ref. [12]. The state of each qubit was measured using one of four proximal charge sensors, each of which was itself measured using frequency-multiplexed reflectometry. For these measurements, we measured one of the four $S - T_0$ qubits using the charge sensor closest to it.

Measurements

The measurements consist of a pulse similar to that shown in Fig. 6.4.1, with the acquisition taken when the device was pulsed into the blockade region. The data is collected via sliced demodulation, which is a specific feature of the instrument used to take these measurements. The demodulated signal is sliced into chunks; by taking a cumulative average, we can investigate the effect of increasing the effective integration time on the characteristics of the readout measurements. Each chunk of 0.76 ns consists of 4,600 samples. Histograms of each of the 131 0.76 ns cumulatively-averaged sub-intervals are shown in Fig. 7.6.3 **a)**.

Optimal readout time

I trained the parametric readout model on each of the histogram slices from Fig. 7.6.3 **a)**, with the aim of finding the integration time that led to the maximum visibility. Each consecutive slice represents an increased integration time of 0.76 ns.

To begin with, the visibility increases as a function of integration time (Fig. 7.6.3 **d)**).

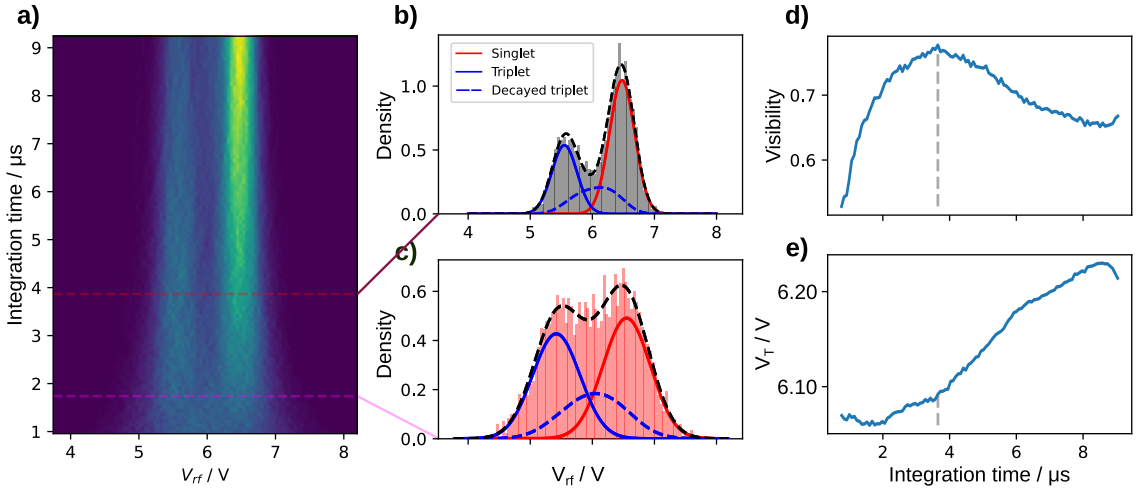


Figure 7.6.3: **a)** Histograms of readout samples from the Copenhagen singlet-triplet qubit for increasing integration time. **b)** The results of training the readout model on the red slice in a), which represents the integration time with maximum visibility. **c)** The results of training the model on the pink slice in a). **d)** Visibility as a function of integration time. The vertical dashed corresponds to the red trace in a) and the results in b). **d)** The optimal readout voltage V_T as a function of integration time, with the dashed line again corresponding to the integration time that results in optimal visibility.

This occurs because we are averaging out noise from the signal while maintaining state coherence. Beyond the optimal integration time $T_{\text{opt}} = 3.65\,\mu\text{s}$, the visibility starts to decrease (Fig. 7.6.3 **d)** as it becomes more likely that states in the higher-energy state decay into a low-energy state within the integration time window. Unfortunately, we did not collect data on the T_1 time of the qubit for comparison.

Fig. 7.6.3 **e)** shows the calculated optimal threshold voltage V_T as a function of integration time. As the integration time increases, the optimal threshold voltage drifts towards higher readout values, meaning closer to the low-energy singlet state. Intuitively, this makes sense: as the integration time gets longer, high-energy triplet states are more likely to relax, and therefore read out closer to the singlet readout voltage.

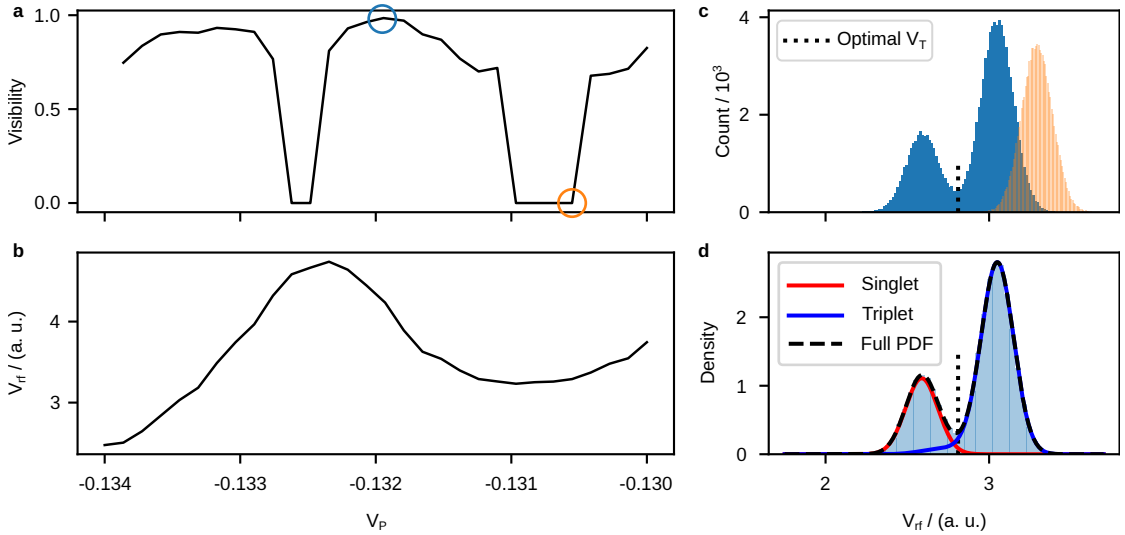


Figure 7.6.4: **a**: State visibility as a function of sensor plunger gate voltage V_P. The sharp drops to zero occur when the readout data for a given plunger voltage are Gaussian distributed, in which case the sample is allocated a visibility score of zero. **b**: Charge sensor signal for each voltage. **c**: Histograms of the single shot data at the blue and orange points as labelled in **a**. **d**: Fitting of blue data point using the maximum likelihood estimation fit as described in the main text. The optimal threshold as calculated by the model is indicated in **d** and the blue data in **c**.

Plunger gate tuning

We collected single shot measurement data as a function of the sensor dot's plunger gate voltage. Finding the optimal plunger gate voltage is not a particularly hard problem, but it serves to show the utility of the algorithm. In Coulomb blockade, the sensor has zero sensitivity, so the qubit state fidelity is near zero. In this regime, the sensor cannot distinguish between the two readout states and so we measure purely Gaussian noise. The procedure from Section 7.5.1 identifies this and the sample is allocated Vis = 0.

By comparing the inferred visibility with the charge sensor signal, we see that the best state visibility is measured at the regimes of high transconductance in the sensor (Fig. 7.6.4 **a**, **b**). As the transconductance of the sensor reduces, so does the visibility; when in blockade, as expected, the algorithm detects that the single-shot data are Gaussian distributed and returns a visibility score of zero (Fig. 7.6.4 **a)**, **c)**, orange

colour).

7.7 Conclusions

In this chapter, I presented an approach to improving qubit readout quality metrics using maximum likelihood estimation to fit data from repeated single-shot qubit measurements. I showed that the trained model extracts the parameters of the system with good accuracy, although the estimated T_1 times are not totally reliable. This is because changes in the T_1 time are hard to detect. Changes in other parameters look like changes in the T_1 time to the model, so it struggles to accurately distinguish these changes. One way to deal with this problem is to measure the T_1 time directly and remove this parameter from the inference. This is a simple measurement and I believe it would lead to a more reliable algorithm. Removing the rate parameter from the inference step is a sensible idea, because that parameter is directly related to other parameters in the problem and attempting to fit them all leads to more complicated optimisation space. However, when the decay rate is long relative to the integration time, the weight parameter w_3 is small, and extracting information from minute changes in small values is not necessarily a particularly accurate process.

I show that the model can be used to optimise the readout fidelity by finding the optimal integration time for the experiment, and also that it can be used to tune experimental parameters to optimise the readout fidelity, such as in the simple case of tuning the charge sensor's plunger gate voltage. I also demonstrated a simple simulator that can be used to develop further, more accurate models based on a similar premise. Given more experimental time, it would be interesting to explore the dependence of qubit metrics over the device parameter space, perhaps to automate sweet-spot identification.

There are other approaches to this type of algorithm, some of which I explored with

my colleagues during my doctorate. These include model fitting of, for instance, Rabi chevron data to extract parameters from the experiment. In a separate approach, we tried using the Fourier transform to extract information from Rabi oscillations, including the decay rate and frequency. All model fitting-based approaches, including the one in this chapter, have the problem that any data can be fit to any model. To ensure that the fitting process is valid, we need to first perform checks to make sure the data is of the type we expect such that we do not lose significant information about the data by referring to the trained model instead.

Chapter 8

Conclusions

We are headed for extinction. Can quantum computers help? They might be able to delay the inevitable, but we are yet to find out. It turns out that manipulating the universe at the nano-scale to do our physics homework for us is a little harder than anticipated.

Manipulating single spins in zero-dimensional quantum dots is an impressive feat of engineering, but using this technology as the basis for a quantum computer is not yet a reality. Of the many challenges that stand in our way, the ones I sought to tackle in this thesis were (are) the tuning and readout problems. Tuning a single or double quantum dot, with modern devices and high-speed reflectometry measurements, is fairly feasible for a well-trained ape (such as the author). Asking the ape to manually tune an array of several quantum dots might get you a groan, but the ape may succeed in this task also. However, tuning several hundred, or thousand, quantum dots, as is required for a full-scale quantum processor, will result in the death of the ape.

One of the inherent physical properties standing in the way of a simple tuning procedure is crosstalk. These devices are tiny, meaning the electric fields generated

by the device gates spread over and interfere with dots in a region larger than desired. Presently, there are attempts to solve this issue at the device level using screening gates. In Chapter 5, I demonstrated that there are fast ways to combine pairs of gates to reduce the effect of crosstalk, in this case to prevent the charge sensing quantum dot from losing sensitivity as we make large changes to the device gate voltages. Similar techniques are explored in Chapter 6 to rapidly create virtual gates.

Electric field strength decays with the square of the distance from the source. As we start to reach larger arrays, crosstalk between dots and gates at larger distances from each other will reduce to barely-measurable levels. In light of this, we should aim to create flexible, modular algorithms that can aid in tuning sections of these arrays. In Chapter 4, I discussed work on this theme.

As the reproducibility of devices improves, the complexity of the algorithms needed to tune them will reduce. Improvements in control electronics, especially those specifically designed for qubit experiments, make the future of this work promising.

8.1 Summary of results

In Chapter 4, I showed how we tuned a double quantum dot using only radio-frequency measurements. These measurements are required for qubit measurement, meaning they are already set up for this part of device operation. Using them for tuning is desirable for this reason, and the fact that the speed of measurement is increased significantly over transport measurements. This work represents the fastest automated tuning of a double dot device in the literature at the time of release. It also highlights the importance of a score function that accurately captures our desire; optimising a score function that does not reflect what we want to improve takes us nowhere. We are fortunate that the signals we seek are periodic, meaning we can take advantage of the deep understanding and optimisation of Fourier transforms. This

all comes down to reducing the number of measurements we need to take before we identify a double dot. In principle, we could take a two-dimensional plunger-plunger stability diagram measurement at every possible voltage location as dictated by the resolution and range of the DACs used to control the experiment. Of course, this is not a scalable approach. Instead, we aim to efficiently navigate device space, searching for signals that indicate the presence of a double quantum dot.

In Chapter 6, I presented methods that contribute towards automating the fine-tuning stage of a reflectometry-based tuning pipeline. The methodology builds on the paradigm introduced in the previous chapters: it is important to be fast if this process is to be scalable. I present a method for extracting virtual gates on a millisecond timescale using fast measurements, which represents a significant speed-up over the manual approach usually taken in laboratories. The constant sensitivity afforded by the results of Chapter 5 mean simple signal processing techniques can be used to stabilise the location of double dot features as we tune the device.

In Chapter 7, I demonstrated how we can extract metrics of interest from qubit measurements by training a parametric model on the data. Some of the parameters of this model are of direct interest, such as the readout visibility. Being able to quickly extract these parameters means we can tune the experiment to a regime in which we see optimal qubit performance. This can lead to the automatic discovery of sweet spots that are known to exist within qubit experiment parameter spaces, which are otherwise tuned to by human experimenters. I demonstrated the algorithm on a realistic simulator, as well as on a real spin qubit.

In combination, these algorithms contribute work towards the unification of tuning and measurement of quantum dot devices. Previous work on quantum dot tuning has been bottle-necked by slow measurement techniques which are not of use in the actual operation of these devices as spin qubits [4], or on neural networks that are trained

on manually-labelled data [128, 138]. Approaches based on neural networks also may need to be fine-tuned on the devices they are tuning, which makes the solution rather circular. By combining our approach with the compensation algorithm in Chapter 5, we could tune a quantum dot array entirely using measurements obtained via a charge sensor, which would allow for direct tuning to a low-charge regime without additional complexity in the rf cavity engineering stage. Further, this would allow for the tuning of large arrays of devices with multiple charge sensors, which are likely to be the future of the field [121].

8.2 Limitations and further work

The algorithm of Chapter 4 represent a strong step forwards, but is not without its limitations. Firstly, we must carefully engineer the rf cavity such that the matching condition is met when the device is in the regime we are searching for. This requires prior knowledge and is also dependent upon tiny, barely-controllable parameters such as the parasitic capacitance due to the bonds between sample and holder. Future work could integrate further improvements from the machine learning toolbox, such as using a Bayesian optimiser to intelligently guide the choice of the next measurement location. We could further speed up the algorithm by using intelligent measurements to capture the same amount of information in fewer pixels. For instance, the circular measurements in Chapter 6 are useful for identifying specific features of the charge stability diagram. This type of approach may be helpful to further reduce the number of measurements needed to tune large arrays. Naturally, this would require a new score function, but would lead to further speed and scalability improvements.

The approach in Chapter 5 is based on open-loop control. The experiments are noisy, with factors such as spurious dots and charge switches to consider. Only closed-loop control can deal with such events. Integrating closed-loop control of the charge sensor

plunger could result in a more stable algorithm, but it requires high-specification hardware to measure and control the experiment at high frequencies. This also requires a modified experimental setup, because low-frequency feedback control needs to be sent via the low-frequency lines, and equivalently for high-frequency feedback control.

The methods in Chapter 6 show promise, but some require refinement. Future work could combine my approaches and explore the double dot regime using the virtual gates to search for inter-dot transitions. A classifier for the Pauli spin blockade measurement in rf would result in a powerful tuning pipeline.

My verification of results for Chapter 7 shows that it is does not accurately obtain the T_1 value, but is accurate for other device parameters. The T_1 is fairly straightforward to measure directly, so a future algorithm could take this as an input. Future work would involve using this algorithm to tune more complex parameters than the charge sensor plunger gate and the optimal readout time.

As discussed in Chapter 4, one of the most useful contributions yet to be made to the field of automatic device tuning is a standardised tuning benchmark routine. I would argue that the variance in real devices means this would have to be a simulated dataset, or perhaps a digital twin, such that tuning pipelines can be tested in a controlled environment. I have contributed to this effort with my colleagues in Ref. [40], in which we created a software for rapidly simulating charge stability diagrams of quantum dot arrays. This could provide the basis for a benchmarking routine in which tuning algorithms are tested for their efficiency and accuracy. Using simulated data also removes the need for a human to score the double dot quality, since we know exactly what regime the data is simulating. Zwolak *et al.* have created a database of simulated measurements for training and testing machine learning models [139], but this database only consists of 1599 idealised, simulated measurements. The

simulator approach from Ref. [40] allows for custom data generation with various noise models, and further developments are in process, such as the addition of spin physics and latching. Capturing the signatures of these complex physical processes is critical, otherwise these models will only be useful in the initial coarse tuning phase. In Ref. [140], the authors train a neural network on a huge amount of data from a single device, effectively the network to replicate the main features from the entire device. They then use the network to find the voltage configuration required for a given charge state. This is slightly circular, since they have trained the network by measuring the entire device, but this type of approach may be useful in combination with classical simulators to include features that are hard to capture with simple models such as the constant capacitance model.

8.2.1 Automated tuning versus device reproducibility

Automated tuning is required to address the variability in devices, such that human experimenters do not have to spend hours or days in the laboratory fiddling with device parameters. As devices get larger, automating the process will become ever more important. However, the coarse tuning requirements of perfect, identical devices would be much lower, because uniform devices should have identical operating regimes to one another. For identical chips, one device could be tuned carefully, and the voltage configuration could then be used for each of the others. This underscores the importance of improving device homogeneity, and work from Intel is spearheading this research direction [141]. However, atomic uniformity across the thousands or millions of qubits required for an operational quantum computer is highly unlikely, so fine-tuning optimisation approaches such as the one I presented in Chapter 6 are likely to remain important.

Bibliography

[1] Stephan G J Philips, Mateusz T Mądzik, Sergey V Amitonov, Sander L de Snoo, Maximilian Russ, Nima Kalhor, Christian Volk, William I L Lawrie, Delphine Brousse, Larysa Tryputen, Brian Paquelet Wuetz, Amir Sammak, Menno Veldhorst, Giordano Scappucci, and Lieven M K Vandersypen. Universal control of a six-qubit quantum processor in silicon. *Nature*, 609(7929):919–924, 2022.

[2] W. I.L. Lawrie, N. W. Hendrickx, F. Van Riggelen, M. Russ, L. Petit, A. Sammak, G. Scappucci, and M. Veldhorst. Spin Relaxation Benchmarks and Individual Qubit Addressability for Holes in Quantum Dots. *Nano Letters*, 20(10):7237–7242, 10 2020.

[3] Simon Geyer, Bence Hetényi, Stefano Bosco, Leon C. Camenzind, Rafael S. Eggli, Andreas Fuhrer, Daniel Loss, Richard J. Warburton, Dominik M. Zumbühl, and Andreas V. Kuhlmann. Anisotropic exchange interaction of two hole-spin qubits. *Nature Physics*, 2024.

[4] H Moon, D T Lennon, J Kirkpatrick, N M Van Esbroeck, L C Camenzind, and Liuqi Yu. Machine learning enables completely automatic tuning of a quantum device faster than human experts. pages 1–19, 2020.

[5] Luigi Federico Menabrea and Ada Lovelace. Sketch of the analytical engine invented by Charles Babbage, 1842.

[6] James R. Powell. The Quantum Limit to Moore's Law. *Proceedings of the IEEE*, 96(8):1247, 2008.

[7] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11, 2019.

[8] Bjoern Lekitsch, Sebastian Weidt, Austin G Fowler, Klaus Mølmer, Simon J Devitt, Christof Wunderlich, and Winfried K Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3(2):e1601540, 2 2017.

[9] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):21314, 2019.

[10] Shuntaro Takeda and Akira Furusawa. Toward large-scale fault-tolerant universalphotonic quantum computing. *APL Photonics*, 4:060902, 2019.

[11] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M.K. Vandersypen. Spins in few-electron quantum dots. *Reviews of Modern Physics*, 79(4):1217–1265, 2007.

[12] Federico Fedele, Anasua Chatterjee, Saeed Fallahi, Geoffrey C. Gardner, Michael J. Manfra, and Ferdinand Kuemmeth. Simultaneous Operations in a Two-Dimensional Array of Singlet-Triplet Qubits. *PRX Quantum*, 2(4):1, 2021.

[13] David P DiVincenzo. The Physical Implementation of Quantum Computation.

Fortschritte der Physik, 48(9-11):771–783, 9 2000.

[14] E Kawakami, P Scarlino, D R Ward, F R Braakman, D E Savage, M G Lagally, Mark Friesen, S N Coppersmith, M A Eriksson, and L M K Vandersypen. Electrical control of a long-lived spin qubit in a Si/SiGe quantum dot. *Nature Nanotechnology*, 9(9):666–670, 2014.

[15] M Veldhorst, J C C Hwang, C H Yang, A W Leenstra, B de Ronde, J P Dehollain, J T Muhonen, F E Hudson, K M Itoh, A Morello, and A S Dzurak. An addressable quantum dot qubit with fault-tolerant control-fidelity. *Nature Nanotechnology*, 9(12):981–985, 2014.

[16] M Veldhorst, C H Yang, J C C Hwang, W Huang, J P Dehollain, J T Muhonen, S Simmons, A Laucht, F E Hudson, K M Itoh, A Morello, and A S Dzurak. A two-qubit logic gate in silicon. *Nature*, 526(7573):410–414, 2015.

[17] D M Zajac, A J Sigillito, M Russ, F Borjans, J M Taylor, G Burkard, and J R Petta. Resonantly driven CNOT gate for electron spins. *Science*, 359(6374):439 LP – 442, 1 2018.

[18] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G S L Brandao, David A Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P Harrigan, Michael J Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S Humble, Sergei V Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R McClean, Matthew

McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C Platt, Chris Quintana, Eleanor G Rieffel, Pedram Roushan, Nicholas C Rubin, Daniel Sank, Kevin J Satzinger, Vadim Smelyanskiy, Kevin J Sung, Matthew D Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[19] T F Watson, S G J Philips, E Kawakami, D R Ward, P Scarlino, M Veldhorst, D E Savage, M G Lagally, Mark Friesen, S N Coppersmith, M A Eriksson, and L M K Vandersypen. A programmable two-qubit quantum processor in silicon. *Nature*, 555(7698):633–637, 2018.

[20] David K Tuckett, Stephen D Bartlett, Steven T Flammia, and Benjamin J Brown. Fault-Tolerant Thresholds for the Surface Code in Excess of $5%$ Under Biased Noise. *Physical Review Letters*, 124(13):130501, 3 2020.

[21] David S Wang, Austin G Fowler, and Lloyd C L Hollenberg. Surface code quantum computing with error rates over 1%. *Physical Review A*, 83(2):20302, 2 2011.

[22] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):32324, 9 2012.

[23] Rui Chao and Ben W Reichardt. Fault-tolerant quantum computation with few qubits. *npj Quantum Information*, 4(1):42, 2018.

[24] Smite-Meister. File:Bloch_sphere.svg [Bloch sphere image]. https://en.wikipedia.org/wiki/File:Bloch_sphere.svg, 2008.

[25] Guido Burkard, Thaddeus D Ladd, John M Nichol, Andrew Pan, and Jason R Petta. Semiconductor spin qubits. *arXiv preprint arXiv:2112.08863*, 2021.

[26] Yinan Fang, Pericles Philippopoulos, Dimitrie Culcer, W. A. Coish, and Stefano Chesi. Recent advances in hole-spin qubits. 10 2022.

[27] Peter Stano and Daniel Loss. Review of performance metrics of spin qubits in gated semiconducting nanostructures. *Nature Reviews Physics*, 4(10):672–688, 10 2022.

[28] D. M. Zajac, T. M. Hazard, X. Mi, E. Nielsen, and J. R. Petta. Scalable Gate Architecture for a One-Dimensional Array of Semiconductor Spin Qubits. *Physical Review Applied*, 6(5):1–8, 2016.

[29] Jun Yoneda, Kenta Takeda, Tomohiro Otsuka, Takashi Nakajima, Matthieu R. Delbecq, Giles Allison, Takumu Honda, Tetsuo Kodera, Shunri Oda, Yusuke Hoshi, Noritaka Usami, Kohei M. Itoh, and Seigo Tarucha. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%. *Nature Nanotechnology*, 13(2):102–106, 2018.

[30] Daniel Loss and David P. DiVincenzo. Quantum computation with quantum dots. *Physical Review A - Atomic, Molecular, and Optical Physics*, 1998.

[31] L M K Vandersypen, H Bluhm, J S Clarke, A S Dzurak, R Ishihara, A Morello, D J Reilly, L R Schreiber, and M Veldhorst. Interfacing spin qubits in quantum dots and donors—hot, dense, and coherent. *npj Quantum Information*, 3(1):34, 2017.

[32] Ruoyu Li, Luca Petit, David P Franke, Juan Pablo Dehollain, Jonas Helsen, Mark Steudtner, Nicole K Thomas, Zachary R Yoscovits, Kanwal J Singh, Stephanie Wehner, Lieven M K Vandersypen, James S Clarke, and Menno

Veldhorst. A crossbar network for silicon quantum dot qubits. *Science Advances*, 4(7):eaar3960, 7 2018.

[33] David S. Weiss and Mark Saffman. Quantum computing with neutral atoms. *Physics Today*, 70(7):44–50, 7 2017.

[34] Lieven M K Vandersypen and Mark A Eriksson. Quantum computing with semiconductor spins. *Physics Today*, 72:8–38, 2019.

[35] Ying Li and Simon C Benjamin. Hierarchical surface code for network quantum computing with modules of arbitrary size. *Physical Review A*, 94(4):42303, 10 2016.

[36] L P Kouwenhoven, D G Austing, and S Tarucha. Few-electron quantum dots. *Reports on Progress in Physics*, 64(6):701–736, 2001.

[37] Andreas Fuhrer Dipl Phys ETH. Phase Coherence, Orbital and Spin States in Quantum Rings. Technical report, 1973.

[38] Haruki Kiyama, Alexander Korsch, Naomi Nagai, Yasushi Kanai, Kazuhiko Matsumoto, Kazuhiko Hirakawa, and Akira Oiwa. Single-electron charge sensing in self-assembled quantum dots. *Scientific Reports*, 8(1), 12 2018.

[39] W. G. Van der Wiel, S De Franceschi, J M Elzerman, T Fujisawa, S Tarucha, and L P Kouwenhoven. Electron transport through double quantum dots, 2003.

[40] Barnaby van Straaten, Joseph Hickie, Lucas Schorling, Jonas Schuff, Federico Fedele, and Natalia Ares. QArray: a GPU-accelerated constant capacitance model simulator for large quantum dot arrays. 4 2024.

[41] Hanno Flentje, Benoit Bertrand, Pierre-André Mortemousque, Vivien Thiney, Arne Ludwig, Andreas D Wieck, Christopher Bäuerle, and Tristan Meunier. A

linear triple quantum dot system in isolated configuration. *Applied Physics Letters*, 110(23):233101, 6 2017.

[42] Will J. Hardy, C. Thomas Harris, Yi Hsin Su, Yen Chuang, Jonathan Moussa, Leon N. Maurer, Jiun Yun Li, Tzu Ming Lu, and Dwight R. Luhman. Single and double hole quantum dots in strained Ge/SiGe quantum wells. *Nanotechnology*, 30(21):0–16, 2019.

[43] Daniel Jirovec, Andrea Hofmann, Andrea Ballabio, Philipp M. Mutter, Giulio Tavani, Marc Botifoll, Alessandro Crippa, Josip Kukucka, Oliver Sagi, Frederico Martins, Jaime Saez-Mollejo, Ivan Prieto, Maksim Borovkov, Jordi Arbiol, Daniel Chrastina, Giovanni Isella, and Georgios Katsaros. A singlet triplet hole spin qubit in planar Ge. *arXiv*, 2020.

[44] Zhanning Wang, Elizabeth Marcellina, Alex R. Hamilton, James H. Cullen, Sven Rogge, Joe Salfi, and Dimitrie Culcer. Optimal operation points for ultrafast, highly coherent Ge hole spin-orbit qubits. *npj Quantum Information*, 7(1), 2021.

[45] M. Lodari, A. Tosato, D. Sabbagh, M. A. Schubert, G. Capellini, A. Sammak, M. Veldhorst, and G. Scappucci. Light effective hole mass in undoped Ge/SiGe quantum wells. *Physical Review B*, 100(4), 7 2019.

[46] Giordano Scappucci, Christoph Kloeffel, Floris A Zwanenburg, Daniel Loss, Maksym Myronov, Jian-Jun Zhang, Silvano De Franceschi, Georgios Katsaros, and Menno Veldhorst. The germanium quantum information route. *Nature Reviews Materials*, 6(10):926–943, 2021.

[47] Daniel Jirovec, Andrea Hofmann, Andrea Ballabio, Philipp M. Mutter, Giulio Tavani, Marc Botifoll, Alessandro Crippa, Josip Kukucka, Oliver Sagi, Frederico Martins, Jaime Saez-Mollejo, Ivan Prieto, Maksim Borovkov, Jordi Arbiol,

Daniel Chrastina, Giovanni Isella, and Georgios Katsaros. A singlet-triplet hole spin qubit in planar Ge. *Nature Materials*, 20(8):1106–1112, 2021.

[48] Christoph Kloeffel, Mircea Trif, and Daniel Loss. Strong spin-orbit interaction and helical hole states in Ge/Si nanowires. *Physical Review B - Condensed Matter and Materials Physics*, 84(19):1–8, 2011.

[49] J. M. Elzerman, R. Hanson, L. H. Willems Van Beveren, L. M.K. Vandersypen, and L. P. Kouwenhoven. Excited-state spectroscopy on a nearly closed quantum dot via charge detection. *Applied Physics Letters*, 84(23):4617–4619, 2004.

[50] Andrea Morello, Jarryd J. Pla, Floris A. Zwanenburg, Kok W. Chan, Kuan Y. Tan, Hans Huebl, Mikko Möttönen, Christopher D. Nugroho, Changyi Yang, Jessica A. Van Donkelaar, Andrew D.C. Alves, David N. Jamieson, Christopher C. Escott, Lloyd C.L. Hollenberg, Robert G. Clark, and Andrew S. Dzurak. Single-shot readout of an electron spin in silicon. *Nature*, 467(7316):687–691, 2010.

[51] B E Kane. Silicon-based Quantum Computation. 48, 2000.

[52] M. Pioro-Ladrière, T. Obata, Y. Tokura, Y. S. Shin, T. Kubo, K. Yoshida, T. Taniyama, and S. Tarucha. Electrically driven single-electron spin resonance in a slanting Zeeman field. *Nature Physics*, 4(10):776–779, 2008.

[53] N. W. Hendrickx, W. I.L. Lawrie, L. Petit, A. Sammak, G. Scappucci, and M. Veldhorst. A single-hole spin qubit. *Nature Communications*, 11(1), 12 2020.

[54] J R Petta, A C Johnson, J M Taylor, E A Laird, A Yacoby, M D Lukin, C M Marcus, M P Hanson, and A C Gossard. Coherent manipulation of coupled electron spins in semiconductor quantum dots. 309(September):2180–2185, 2005.

[55] K. Ono, D. G. Austing, Y. Tokura, and S. Tarucha. Current rectification by Pauli exclusion in a weakly coupled double quantum dot system. *Science*, 297(5585):1313–1317, 2002.

[56] N. S. Lai, W. H. Lim, C. H. Yang, F. A. Zwanenburg, W. A. Coish, F. Qassemi, A. Morello, and A. S. Dzurak. Pauli spin blockade in a highly tunable silicon double quantum dot. *Scientific Reports*, 1, 2011.

[57] L. Petit, H. G.J. Eenink, M. Russ, W. I.L. Lawrie, N. W. Hendrickx, S. G.J. Philips, J. S. Clarke, L. M.K. Vandersypen, and M. Veldhorst. Universal quantum logic in hot silicon qubits. *Nature*, 580(7803):355–359, 4 2020.

[58] A. C. Johnson, J. R. Petta, J. M. Taylor, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Triplet-singlet spin relaxation via nuclei in a double quantum dot. *Nature*, 435(7044):925–928, 2005.

[59] Santiago Serrano, MengKe Feng, Wee Han Lim, Amanda E Seedhouse, Tuomo Tanttu, Will Gilbert, Christopher C Escott, Nikolay V Abrosimov, Hans-Joachim Pohl, Michael L W Thewalt, Fay E Hudson, Andre Saraiva, Andrew S Dzurak, and Arne Laucht. Improved Single-Shot Qubit Readout Using Twin rf-SET Charge Correlations. *PRX Quantum*, 5(1):10301, 1 2024.

[60] Federico Fedele. Spin interactions within a two-dimensional array of GaAs double dots. *Thesis*, 2020.

[61] Nico W. Hendrickx, William I.L. Lawrie, Maximilian Russ, Floor van Riggelen, Sander L. de Snoo, Raymond N. Schouten, Amir Sammak, Giordano Scappucci, and Menno Veldhorst. A four-qubit germanium quantum processor. *Nature*, 591(7851):580–585, 2021.

[62] Ruoyu Li, Luca Petit, David P. Franke, Juan Pablo Dehollain, Jonas Helsen, Mark Steudtner, Nicole K. Thomas, Zachary R. Yoscovits, Kanwal J. Singh,

Stephanie Wehner, Lieven M.K. Vandersypen, James S. Clarke, and Menno Veldhorst. A crossbar network for silicon quantum dot qubits. *Science Advances*, 4(7):1–10, 2018.

[63] Cody Jones, Michael A Fogarty, Andrea Morello, Mark F Gyure, Andrew S Dzurak, and Thaddeus D Ladd. Logical Qubit in a Linear Array of Semiconductor Quantum Dots. *Physical Review X*, 8(2):21058, 6 2018.

[64] O. Crawford, J. R. Cruise, N. Mertig, and M. F. Gonzalez-Zalba. Compilation and scaling strategies for a silicon quantum processor with sparse two-dimensional connectivity. *npj Quantum Information*, 9(1), 2023.

[65] D. L. Craig, H. Moon, F. Fedele, D. T. Lennon, B. Van Straaten, F. Vigneau, L. C. Camenzind, D. M. Zumbuhl, G. A.D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares. Bridging the Reality Gap in Quantum Devices with Physics-Aware Machine Learning. *Physical Review X*, 14(1), 1 2024.

[66] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M.K. Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21), 2016.

[67] Tim Botzem, Michael D. Shulman, Sandra Foletti, Shannon P. Harvey, Oliver E. Dial, Patrick Bethke, Pascal Cerfontaine, Robert P.G. McNeil, Diana Mahalu, Vladimir Umansky, Arne Ludwig, Andreas Wieck, Dieter Schuh, Dominique Bougeard, Amir Yacoby, and Hendrik Bluhm. Tuning Methods for Semiconductor Spin Qubits. *Physical Review Applied*, 10(5):1, 2018.

[68] C. J. Van Diepen, P. T. Eendebak, B. T. Buijtendorp, U. Mukhopadhyay, T. Fujita, C. Reichl, W. Wegscheider, and L. M.K. Vandersypen. Automated tuning of inter-dot tunnel coupling in double quantum dots. *Applied Physics Letters*, 113(3):1–9, 2018.

[69] A. R. Mills, M. M. Feldman, C. Monical, P. J. Lewis, K. W. Larson, A. M. Mounce, and J. R. Petta. Computer-automated tuning procedures for semiconductor quantum dot arrays. *Applied Physics Letters*, 115(11), 2019.

[70] Sandesh S. Kalantre, Justyna P. Zwolak, Stephen Ragole, Xingyao Wu, Neil M. Zimmerman, M. D. Stewart, and Jacob M. Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):1–10, 2019.

[71] Renato Durrer, Benedikt Kratochwil, Jonne V. Koski, Andreas J. Landig, Christian Reichl, Werner Wegscheider, Thomas Ihn, and Eliska Greplova. Automated Tuning of Double Quantum Dots into Specific Charge States Using Neural Networks. *Physical Review Applied*, 13(5):54019, 5 2020.

[72] Julian D. Teske, Simon Sebastian Humpohl, René Otten, Patrick Bethke, Pascal Cerfontaine, Jonas Dedden, Arne Ludwig, Andreas D. Wieck, and Hendrik Bluhm. A machine learning approach for automated fine-tuning of semiconductor spin qubits. *Applied Physics Letters*, 114(13), 2019.

[73] J Darulová, S J Pauka, N Wiebe, K W Chan, M C Cassidy, and M Troyer. Autonomous tuning and charge state detection of gate defined quantum dots. *arXiv preprint arXiv:1911.10709*, 2019.

[74] D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Fast single-charge sensing with a rf quantum point contact. *Applied Physics Letters*, 91(16):2005–2008, 2007.

[75] J. M. Hornibrook, J. I. Colless, A. C. Mahoney, X. G. Croot, S. Blanvillain, H. Lu, A. C. Gossard, and D. J. Reilly. Frequency multiplexing for readout of spin qubits. *Applied Physics Letters*, 104(10):1–5, 2014.

[76] Takashi Nakajima, Yohei Kojima, Yoshihiro Uehara, Akito Noiri, Kenta Takeda,

Takashi Kobayashi, and Seigo Tarucha. Real-Time Feedback Control of Charge Sensing for Quantum Dot Qubits. *Physical Review Applied*, 15(3):1, 2021.

[77] C Barthel, M Kjærgaard, J Medford, M Stopa, C M Marcus, M P Hanson, and A C Gossard. Fast sensing of double-dot charge arrangement and spin state with a radio-frequency sensor quantum dot. *Physical Review B*, 81(16):161308, 4 2010.

[78] X. G. Croot, S. J. Pauka, M. C. Jarratt, H. Lu, A. C. Gossard, J. D. Watson, G. C. Gardner, S. Fallahi, M. J. Manfra, and D. J. Reilly. Gate-Sensing Charge Pockets in the Semiconductor-Qubit Environment. *Physical Review Applied*, 11(6):1–6, 2019.

[79] M. F. Gonzalez-Zalba, S. Barraud, A. J. Ferguson, and A. C. Betz. Probing the limits of gate-based charge sensing. *Nature Communications*, 6, 2015.

[80] J. I. Colless, A. C. Mahoney, J. M. Hornibrook, A. C. Doherty, H. Lu, A. C. Gossard, and D. J. Reilly. Dispersive readout of a few-electron double quantum dot with fast rf gate sensors. *Physical Review Letters*, 110(4):1–5, 2013.

[81] A. Crippa, R. Ezzouch, A. Aprá, A. Amisse, R. Laviéville, L. Hutin, B. Bertrand, M. Vinet, M. Urdampilleta, T. Meunier, M. Sanquer, X. Jehl, R. Maurand, and S. De Franceschi. Gate-reflectometry dispersive readout and coherent control of a spin qubit in silicon. *Nature Communications*, 10(1):1–6, 2019.

[82] Florian Vigneau, Federico Fedele, Anasua Chatterjee, David Reilly, Ferdinand Kuemmeth, Fernando Gonzalez-Zalba, Edward Laird, and Natalia Ares. Probing quantum devices with radio-frequency reflectometry. 2022.

[83] A. R. Mills, D. M. Zajac, M. J. Gullans, F. J. Schupp, T. M. Hazard, and J. R. Petta. Shuttling a single charge across a one-dimensional array of silicon quantum dots. *Nature Communications*, 10(1), 2019.

[84] C. Volk, A. M.J. Zwerver, U. Mukhopadhyay, P. T. Eendebak, C. J. van Diepen, J. P. Dehollain, T. Hensgens, T. Fujita, C. Reichl, W. Wegscheider, and L. M.K. Vandersypen. Loading a quantum-dot based "Qubyte" register. *npj Quantum Information*, 5(1):1–12, 2019.

[85] D. T. Lennon, H. Moon, L. C. Camenzind, Liuqi Yu, D. M. Zumbühl, G. A .D. Briggs, M. A. Osborne, E. A. Laird, and N. Ares. Efficiently measuring a quantum device using machine learning. *npj Quantum Information*, 5(1):1–8, 2019.

[86] J Darulová, S J Pauka, N Wiebe, K W Chan, G C Gardener, M J Manfra, M C Cassidy, and M Troyer. Autonomous Tuning and Charge-State Detection of Gate-Defined Quantum Dots. *Physical Review Applied*, 13(5):54005, 5 2020.

[87] Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, J P Dodson, E R MacQuarrie, D E Savage, M G Lagally, S N Coppersmith, Mark A Eriksson, and Jacob M Taylor. Autotuning of Double-Dot Devices In Situ with Machine Learning. *Physical Review Applied*, 13(3):34075, 3 2020.

[88] W I L Lawrie, H G J Eenink, N W Hendrickx, J M Boter, L Petit, S V Amitonov, M Lodari, B Paquelet Wuetz, C Volk, and S G J Philips. Quantum dot arrays in silicon and germanium. *Applied Physics Letters*, 116(8):80501, 2020.

[89] T-K Hsiao, C J van Diepen, Uditendu Mukhopadhyay, Christian Reichl, W Wegscheider, and L M K Vandersypen. Efficient orthogonal control of tunnel couplings in a quantum dot array. *arXiv preprint arXiv:2001.07671*, 2020.

[90] Florian Vigneau, Federico Fedele, Anasua Chatterjee, David Reilly, Ferdi-

nand Kuemmeth, Fernando Gonzalez-Zalba, Edward Laird, and Natalia Ares. Probing quantum devices with radio-frequency reflectometry. 2 2022.

[91] Barnaby van Straaten, Federico Fedele, Florian Vigneau, Joseph Hickie, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. All rf-based tuning algorithm for quantum devices using machine learning. 11 2022.

[92] J. R. Prance, Zhan Shi, C. B. Simmons, D. E. Savage, M. G. Lagally, L. R. Schreiber, L. M.K. Vandersypen, Mark Friesen, Robert Joynt, S. N. Coppersmith, and M. A. Eriksson. Single-shot measurement of triplet-singlet relaxation in a Si/SiGe double quantum dot. *Physical Review Letters*, 108(4):1–4, 2012.

[93] Will Gilbert, Tuomo Tanttu, Wee Han Lim, Meng Ke Feng, Jonathan Y. Huang, Jesus D. Cifuentes, Santiago Serrano, Philip Y. Mai, Ross C.C. Leon, Christopher C. Escott, Kohei M. Itoh, Nikolay V. Abrosimov, Hans Joachim Pohl, Michael L.W. Thewalt, Fay E. Hudson, Andrea Morello, Arne Laucht, Chih Hwan Yang, Andre Saraiva, and Andrew S. Dzurak. On-demand electrical control of spin qubits. *Nature Nanotechnology*, 18(2):131–136, 2023.

[94] Guoji Zheng, Nodar Samkharadze, Marc L. Noordam, Nima Kalhor, Delphine Brousse, Amir Sammak, Giordano Scappucci, and Lieven M.K. Vandersypen. Rapid gate-based spin read-out in silicon using an on-chip resonator. *Nature Nanotechnology*, 14(8):742–746, 2019.

[95] M Field, C G Smith, Michael Pepper, D A Ritchie, J E F Frost, G A C Jones, and D G Hasko. Measurements of Coulomb blockade with a noninvasive voltage probe. *Physical review letters*, 70(9):1311, 1993.

[96] D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Fast single-charge

sensing with a rf quantum point contact. *Applied Physics Letters*, 91(16):1–4, 2007.

[97] C Barthel. Control and Fast Measurement of Spin Qubits. *PhD Thesis, Harvard University*, pages 1–184, 2010.

[98] C. H. Yang, W. H. Lim, F. A. Zwanenburg, and A. S. Dzurak. Dynamically controlled charge sensing of a few-electron silicon quantum dot. *AIP Advances*, 1(4), 2011.

[99] Bruno Küng, Simon Gustavsson, Theodore Choi, Ivan Shorubalko, Oliver Pfäffli, Fabian Hassler, Gianni Blatter, Matthias Reinwald, Werner Wegscheider, Silke Schön, Thomas Ihn, and Klaus Ensslin. Measurement back-action in quantum point-contact charge sensing. *Entropy*, 12(7):1721–1732, 2010.

[100] R. J. Schoelkopf, P. Wahlgren, A. A. Kozhevnikov, P. Delsing, and D. E. Prober. The radio-frequency single-electron transistor (RF-SET): A fast and ultrasensitive electrometer. *Science*, 280(5367):1238–1242, 1998.

[101] N. Ares, F. J. Schupp, A. Mavalankar, G. Rogers, J. Griffiths, G. A.C. Jones, I. Farrer, D. A. Ritchie, C. G. Smith, A. Cottet, G. A.D. Briggs, and E. A. Laird. Sensitive Radio-Frequency Measurements of a Quantum Dot by Tuning to Perfect Impedance Matching. *Physical Review Applied*, 5(3):1–6, 2016.

[102] T Müller, B Küng, S Hellmüller, P Studerus, K Ensslin, T Ihn, M Reinwald, and W Wegscheider. An in situ tunable radio-frequency quantum point contact. *Applied Physics Letters*, 97(20):202104, 11 2010.

[103] Akito Noiri, Kenta Takeda, Jun Yoneda, Takashi Nakajima, Tetsuo Kodera, and Seigo Tarucha. Radio-Frequency-Detected Fast Charge Sensing in Undoped Silicon Quantum Dots. *Nano Letters*, 20(2):947–952, 2 2020.

[104] Anderson West, Bas Hensen, Alexis Jouan, Tuomo Tanttu, Chih Hwan Yang, Alessandro Rossi, M. Fernando Gonzalez-Zalba, Fay Hudson, Andrea Morello, David J. Reilly, and Andrew S. Dzurak. Gate-based single-shot readout of spins in silicon. *Nature Nanotechnology*, 14(5):437–441, 2019.

[105] M R Hogg, P Pakkiam, S K Gorman, A V Timofeev, Y Chung, G K Gulati, M G House, and M Y Simmons. Single-Shot Readout of Multiple Donor Electron Spins with a Gate-Based Sensor. *PRX Quantum*, 4(1):10319, 2 2023.

[106] J Stehlik, Y.-Y. Liu, C M Quintana, C Eichler, T R Hartke, and J R Petta. Fast Charge Sensing of a Cavity-Coupled Double Quantum Dot Using a Josephson Parametric Amplifier. *Phys. Rev. Applied*, 4(1):14018, 7 2015.

[107] Carl Edward Rasmussen. Gaussian Processes in Machine Learning. Technical report.

[108] Thomas Beckers. An Introduction to Gaussian Process Models. 2 2021.

[109] Austin Tripp and José Miguel Hernández-Lobato. Diagnosing and fixing common problems in Bayesian optimization for molecule design. 6 2024.

[110] Mateusz T. Mądzik, Serwan Asaad, Akram Youssry, Benjamin Joecker, Kenneth M. Rudinger, Erik Nielsen, Kevin C. Young, Timothy J. Proctor, Andrew D. Baczewski, Arne Laucht, Vivien Schmitt, Fay E. Hudson, Kohei M. Itoh, Alexander M. Jakob, Brett C. Johnson, David N. Jamieson, Andrew S. Dzurak, Christopher Ferrie, Robin Blume-Kohout, and Andrea Morello. Precision tomography of a three-qubit donor quantum processor in silicon. *Nature*, 601(7893):348–353, 2022.

[111] Elliot J Connors, J J Nelson, and John M Nichol. Rapid High-Fidelity Spin-State Readout in Si/Si-Ge Quantum Dots via rf Reflectometry. *Physical Review Applied*, 13(2):24019, 2 2020.

[112] E. A. Laird, J. M. Taylor, D. P. Divincenzo, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Coherent spin manipulation in an exchange-only qubit. *Physical Review B - Condensed Matter and Materials Physics*, 82(7):1–6, 2010.

[113] Zhenyu Cai, Michael A. Fogarty, Simon Schaal, Sofia Patomäki, Simon C. Benjamin, and John J.L. Morton. A silicon surface code architecture resilient against leakage errors. *Quantum*, 3:1–27, 2019.

[114] Frank J Massey. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[115] Sandesh S Kalantre, Justyna P Zwolak, Stephen Ragole, Xingyao Wu, Neil M Zimmerman, Michael D Stewart, and Jacob M Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):1–10, 2019.

[116] Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P. Zwolak. Automated extraction of capacitive coupling for quantum dot systems. pages 1–8, 2023.

[117] M. Lapointe-Major, O. Germain, J. Camirand Lemyre, D. Lachance-Quirion, S. Rochette, F. Camirand Lemyre, and M. Pioro-Ladrière. Algorithm for automated tuning of a quantum dot into the single-electron regime. *Physical Review B*, 102(8):85301, 2020.

[118] Joseph Hickie, Barnaby van Straaten, Federico Fedele, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. Automated long-range compensation of an rf quantum dot sensor. pages 1–9, 2023.

[119] T. Hensgens, T. Fujita, L. Janssen, Xiao Li, C. J. Van Diepen, C. Reichl, W. Wegscheider, S. Das Sarma, and L. M.K. Vandersypen. Quantum simulation

of a Fermi-Hubbard model using a semiconductor quantum dot array. *Nature*, 548(7665):70–73, 2017.

[120] F. J. Schupp, F. Vigneau, Y. Wen, A. Mavalankar, J. Griffiths, G. A.C. Jones, I. Farrer, D. A. Ritchie, C. G. Smith, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A.D. Briggs, N. Ares, and E. A. Laird. Sensitive radiofrequency readout of quantum dots using an ultra-low-noise SQUID amplifier. *Journal of Applied Physics*, 127(24), 2020.

[121] Francesco Borsoi, Nico W. Hendrickx, Valentin John, Sayr Motz, Floor van Riggelen, Amir Sammak, Sander L. de Snoo, Giordano Scappucci, and Menno Veldhorst. Shared control of a 16 semiconductor quantum dot crossbar array. 2022.

[122] Fabian Hader, Jan Vogelbruch, Simon Humpohl, Tobias Hangleiter, Chimezie Eguzo, Stefan Heinen, Stefanie Meyer, and Stefan Van Waasen. On Noise-Sensitive Automatic Tuning of Gate-Defined Sensor Dots. *IEEE Transactions on Quantum Engineering*, 4, 2023.

[123] Jonas Schuff, Miguel J. Carballido, Madeleine Kotzagiannidis, Juan Carlos Calvo, Marco Caselli, Jacob Rawling, David L. Craig, Barnaby van Straaten, Brandon Severin, Federico Fedele, Simon Svab, Pierre Chevalier Kwon, Rafael S. Eggli, Taras Patlatiuk, Nathan Korda, Dominik Zumbühl, and Natalia Ares. Fully autonomous tuning of a spin qubit. 2 2024.

[124] N. W. Hendrickx, D. P. Franke, A Sammak, G Scappucci, and M Veldhorst. Fast two-qubit logic with holes in germanium. *Nature*, 577(7791):487–491, 2020.

[125] R. C.C. Leon, C. H. Yang, J. C.C. Hwang, J. Camirand Lemyre, T. Tanttu, W. Huang, K. W. Chan, K. Y. Tan, F. E. Hudson, K. M. Itoh, A. Morello,

A. Laucht, M. Pioro-Ladrière, A. Saraiva, and A. S. Dzurak. Coherent spin control of s-, p-, d- and f-electrons in a silicon quantum dot. *Nature Communications*, 11(1), 12 2020.

[126] Pieter Eendebak. Quantum Technology Toolbox, 2018.

[127] Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P. Zwolak. Tuning Arrays with Rays: Physics-Informed Tuning of Quantum Dot Charge States. *Physical Review Applied*, 20(3):1, 2023.

[128] Hanwei Liu, Baochuan Wang, Ning Wang, Zhonghai Sun, Huili Yin, Haiou Li, Gang Cao, and Guoping Guo. An automated approach for consecutive tuning of quantum dot arrays. *Applied Physics Letters*, 121(8):084002, 8 2022.

[129] Jana Darulová. Automated Tuning of Gate-defined Quantum Dots.

[130] Giovanni A. Oakes, Jingyu Duan, John J.L. Morton, Alpha Lee, Charles G. Smith, and M. Fernando Gonzalez Zalba. Automatic virtual voltage extraction of a 2×N array of quantum dots with machine learning. *arXiv*, pages 25–29, 2020.

[131] Justyna P. Zwolak, Thomas McJunkin, Sandesh S. Kalantre, Samuel F. Neyens, E. R. MacQuarrie, Mark A. Eriksson, and Jacob M. Taylor. Ray-Based Framework for State Identification in Quantum Dot Devices. *PRX Quantum*, 2(2):1, 2021.

[132] N. Piot, B. Brun, V. Schmitt, S. Zihlmann, V. P. Michal, A. Apra, J. C. Abadillo-Uriel, X. Jehl, B. Bertrand, H. Niebojewski, L. Hutin, M. Vinet, M. Urdampilleta, T. Meunier, Y. M. Niquet, R. Maurand, and S. De Franceschi. A single hole spin with enhanced coherence in natural silicon. *Nature Nanotechnology*, 17(10):1072–1077, 2022.

[133] Daniel Jirovec, Philipp M. Mutter, Andrea Hofmann, Alessandro Crippa, Marek Rychetsky, David L. Craig, Josip Kukucka, Frederico Martins, Andrea Ballabio, Natalia Ares, Daniel Chrastina, Giovanni Isella, Guido Burkard, and Georgios Katsaros. Dynamics of Hole Singlet-Triplet Qubits with Large g -Factor Differences. *Physical Review Letters*, 128(12):126803, 2022.

[134] J Bergli, Y M Galperin, and B L Altshuler. Decoherence in qubits due to low-frequency noise. 4 2009.

[135] C. Barthel, D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Rapid single-shot measurement of a singlet-triplet qubit. *Physical Review Letters*, 103(16):1–4, 2009.

[136] C Barthel, J Medford, H Bluhm, A Yacoby, C M Marcus, M P Hanson, and A C Gossard. Relaxation and readout visibility of a singlet-triplet qubit in an Overhauser field gradient. *Physical Review B*, 85(3):35306, 1 2012.

[137] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, 2002.

[138] J. Darulová, S. J. Pauka, N. Wiebe, K. W. Chan, G. C. Gardener, M. J. Manfra, M. C. Cassidy, and M. Troyer. Autonomous tuning and charge state detection of gate defined quantum dots. 11 2019.

[139] Justyna P. Zwolak, Joshua Ziegler, Sandesh S. Kalantre, and Jacob M Taylor. QFlow 2.0: Quantum dot data for machine learning, 2022.

[140] Hans Christian Ruiz Euler, Marcus N. Boon, Jochem T. Wildeboer, Bram van de Ven, Tao Chen, Hajo Broersma, Peter A. Bobbert, and Wilfred G. van der Wiel. A deep-learning approach to realizing functionality in nanoelectronic devices. *Nature Nanotechnology*, 15(12):992–998, 12 2020.

[141] Samuel Neyens, Otto K. Zietz, Thomas F. Watson, Florian Luthi, Aditi Nethwewala, Hubert C. George, Eric Henry, Mohammad Islam, Andrew J. Wagner, Felix Borjans, Elliot J. Connors, J. Corrigan, Matthew J. Curry, Daniel Keith, Roza Kotlyar, Lester F. Lampert, Mateusz T. Mądzik, Kent Millard, Fahd A. Mohiyaddin, Stefano Pellerano, Ravi Pillarisetty, Mick Ramsey, Rostyslav Savytskyy, Simon Schaal, Guoji Zheng, Joshua Ziegler, Nathaniel C. Bishop, Stephanie Bojarski, Jeanette Roberts, and James S. Clarke. Probing single electrons across 300-mm spin qubit wafers. *Nature*, 629(8010):80–85, 5 2024.

Appendix A

Impedance matching circuit derivation

To simplify further, since $R_{\text{dev}} \gg 1$ and $C_s \gg C_d$, we can remove the term including C_d to return

$$Z_{\text{load}} = i\omega L + \frac{R_{\text{dev}}}{1 + i\omega C_s R_{\text{dev}}}, \tag{A.0.1}$$

At the resonant frequency, $\text{Im}(Z_{\text{load}}) = 0$, and so by rearrangement and with some further simplification we calculate the resonant frequency to be

$$\omega_{\text{res}} = \sqrt{\frac{1}{LC_s}}. \tag{A.0.2}$$

For further analysis, I rearrange Eq. A.0.1 and analyse the real and imaginary parts separately. Thus:

$$Z_{\text{load}}(\omega) = i\omega L + \frac{R_{\text{dev}}(1 - i\omega C R_{\text{dev}})}{1 + \omega^2 C^2 R_{\text{dev}}^2} \tag{A.0.3}$$

$$= \frac{R_{\text{dev}}}{1 + \omega^2 C^2 R_{\text{dev}}^2} + i\left[\omega L = \frac{\omega C R_{\text{dev}}^2}{1 + \omega^2 C^2 R_{\text{dev}}^2}\right]. \tag{A.0.4}$$

By using the relation

$$\omega = \omega_r + \Delta\omega, \tag{A.0.5}$$

where $\Delta\omega$ is the difference between the applied ac frequency ω and the circuit's resonant frequency ω_r, the real part of this equation simplifies to

$$\Re(Z_{\text{load}}) = \frac{R_{\text{dev}}}{1 + \omega^2 C_s^2 R_{\text{dev}}^2} \tag{A.0.6}$$

$$\approx \frac{R_{\text{dev}}}{\omega^2 C_s^2 R_{\text{dev}}^2} \tag{A.0.7}$$

$$= \frac{L}{R_{\text{dev}} C_s}(1 - 2\frac{\Delta\omega}{\omega_r}) \tag{A.0.8}$$

$$= R_{\text{eff}}(1 - 2\frac{\Delta\omega}{\omega_r}), \tag{A.0.9}$$

where

$$R_{\text{eff}} = \frac{L}{R_{\text{dev}} C_s} \tag{A.0.10}$$

is the effective resistance of the circuit. By similar analysis, the imaginary part of Eq. A.0.1 becomes

$$\Im(Z_{\text{load}}) = \omega L - \frac{\omega C_s R_{\text{dev}}^2}{1 + \omega^2 C_s^2 R_{\text{dev}}^2} \tag{A.0.11}$$

$$\approx \omega L - \frac{\omega C_s R_{\text{dev}}^2}{\omega^2 C_s^2 R_{\text{dev}}^2} \tag{A.0.12}$$

$$= \omega L - \frac{1}{\omega C_s} \tag{A.0.13}$$

$$= L(\omega_r + \Delta\omega) - \frac{1}{C_s(\omega_r + \Delta\omega)}. \tag{A.0.14}$$

With rearrangement and considering that $\Delta\omega/\omega_r \ll 1$, we can express

$$\Im(Z_{\text{load}}) = 2\sqrt{\frac{L}{C}}\frac{\Delta\omega}{\omega_r}. \tag{A.0.15}$$

Combining the analysis of the real and imaginary parts, we reach the equation

$$Z_{\text{load}} = R_{\text{eff}}\left(1 - 2\frac{\Delta\omega}{\omega_r}\right) + 2i\sqrt{\frac{L}{C}}\frac{\Delta\omega}{\omega_r} \tag{A.0.16}$$

Appendix B

Propagation of errors in score function

The variance of our entire two-dimensional measurement, which is the value of the metric, is given by

$$\sigma_M^2 = \frac{1}{n}\sum_i m_i^2 - \frac{1}{n^2}\left(\sum_i m_i\right)^2, \tag{B.0.1}$$

where n is the number of pixels in our measurement M and i indexes each pixel. This formulation makes the assumption that the probability of each value in M is equal.

The variance formula for the propagation of errors is

$$\sigma_f^2 = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \left(\frac{\partial f}{\partial z}\right)^2 \sigma_z^2 + ... \tag{B.0.2}$$

where σ_f^2 is the variance of the function f and σ_k^2 is the variance of variable k. Applying Equation B.0.2 to Equation B.0.1 gives us the variance of the metric. We

first need the partial derivatives that make up Equation B.0.2:

$$\frac{\partial \sigma_M^2}{\partial m_j} = \frac{2m_i}{n}\delta_{ij} - \frac{2}{n^2}\delta_{ij}\left(\sum_i m_i\right) \tag{B.0.3}$$

$$= \frac{2m_j}{n} - \frac{2}{n^2}\left(\sum_j m_j\right), \tag{B.0.4}$$

where j indexes the pixels in the measurement M.

Therefore the variance of the metric, σ_T^2, is

$$\sigma_T^2 = \sum_j \left(\left(\frac{\partial \sigma_M^2}{\partial m_j}\right)\sigma_{m_j}^2\right)^2, \tag{B.0.5}$$

The variance of each pixel, $\sigma_{m_j}^2$, is directly from the noise in the measurement and is therefore the same as the variance of the noise N, σ_N^2. Therefore

$$\sigma_T^2 = \sum_j \sigma_N^2 \left(\frac{2}{n}m_j - \frac{2}{n^2}\left(\sum_j m_j\right)\right)^2 \tag{B.0.6}$$

$$= \sum_j \frac{4\sigma_N^2}{n^2}\left(m_j - \frac{1}{n}\sum_j m_j\right)^2 \tag{B.0.7}$$

$$= \frac{4\sigma_N^2}{n^2}\sum_j (m_j - \bar{m})^2 \tag{B.0.8}$$

$$= \frac{4\sigma_N^2}{n}\sum_j \frac{(m_j - \bar{m})^2}{n} \tag{B.0.9}$$

$$= \frac{4\sigma_N^2}{n}\sigma_M^2 \tag{B.0.10}$$

Appendix C

Derivation of optimal threshold voltage

The assumption that the $|1\rangle$ state reads out at a higher voltage than the $|0\rangle$ state implies that

$$n_{|1\rangle}(V) - n_{|0\rangle}(V) = \begin{cases} < 0 & \text{if } V < V_{=} \\ 0 & \text{if } V = V_{=} \\ > 0 & \text{if } V > V_{=} \end{cases} \tag{C.0.1}$$

By combination of Eqs. (7.1.1), (7.1.2) and (7.1.3), we get

$$V = 1 - \int_{V_T}^{\infty} n_{|0\rangle}(V)\, dV - \int_{-\infty}^{V_T} n_{|1\rangle}(V)\, dV. \tag{C.0.2}$$

Since $n_{|1\rangle}(V)$ is a probability distribution,

$$\int_{-\infty}^{\infty} n_{|1\rangle}(V)\, dV = 1, \tag{C.0.3}$$

therefore

$$\int_{-\infty}^{V_T} n_{|1\rangle}(V)\,dV + \int_{V_T}^{\infty} n_{|1\rangle}(V)\,dV = 1 \tag{C.0.4}$$

$$\int_{-\infty}^{V_T} n_{|1\rangle}(V)\,dV = 1 - \int_{V_T}^{\infty} n_{|1\rangle}(V)\,dV. \tag{C.0.5}$$

We can therefore write the visibility as

$$V = 1 - \int_{V_T}^{\infty} n_{|0\rangle}(V)\,dV - \left(1 - \int_{V_T}^{\infty} n_{|1\rangle}(V)\,dV\right) \tag{C.0.6}$$

$$= \int_{V_T}^{\infty} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV \tag{C.0.7}$$

If we select a threshold voltage $V_T < V_=$,

$$V = \int_{V_T}^{V_=} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV + \int_{V_=}^{\infty} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV. \tag{C.0.8}$$

By Eq. (C.0.1), $\int_{V_T}^{V_=} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV$ is negative, so

$$V = \int_{V_=}^{\infty} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV - C_1. \tag{C.0.9}$$

If we select a threshold voltage $V_T > V_=$,

$$V = \int_{V_=}^{\infty} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV - \int_{V_=}^{V_T} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V)\right)\,dV, \tag{C.0.10}$$

and in this case by Eq. (C.0.1) the second term is a positive number and therefore

$$V = \int_{V_=}^{\infty} \left(n_{|1\rangle}(V) - n_{|0\rangle}(V) \right) \, dV - C_2. \tag{C.0.11}$$