

Adaptive Bayesian Calibration of Two Rabi Frequencies from Collective Two-Qubit Measurements

Nicole Widmer
ETH Zürich

Supervised by:

Vlad Negnevitsky, Brennan de Neeve and Matteo Marinelli

Supervising professor:

Prof. Dr. Jonathan Home

ETH Zürich, Trapped Ion Quantum Information Group

August 2018

Abstract

With the development of large quantum computing systems it becomes necessary to find protocols which allow for the calibration of individual parts of this system to be automated. In this work, a Bayesian approach to perform such calibrations is explored. First, a one-dimensional inference scheme to estimate one Rabi frequency in a two qubit system is derived for both distinguishable and indistinguishable qubits. A similar approach is then used to derive a two-dimensional scheme which estimates both Rabi frequencies simultaneously. Simulations show that the performance of the one-dimensional protocol is highly dependent on an initial guess for one of the Rabi frequencies given as input. When comparing the actual Rabi frequency Ω to the estimated value $\hat{\Omega}$ and accepting an estimated result as correct if $|\Omega - \hat{\Omega}| \leq \frac{0.2}{100}\Omega$, a maximal accuracy of 97% can be achieved, but this is only the case for specific inputs and additionally depends on the difference between the two Rabi frequencies. For other inputs and differences, the accuracy can drop as low as 0%. However, an iterative protocol, using the result of the previous estimation as input to a new inference, lifts this dependency. For an iterative protocol, accuracies as high as 98% can be achieved and do not drop below 46%. The two-dimensional protocol shows some dependency on the difference between the two Rabi frequencies which should be estimated. Setting the threshold of acceptance of a result to $\frac{2}{100}\Omega_i$, a maximal accuracy of 93% can be achieved when $\Omega_1 = \Omega_2$ and an accuracy of 51% when $|\Omega_1 - \Omega_2| = 0.1$ MHz.

Acknowledgements

During my work on this thesis, I was supported by my supervisors Vlad Negnevitsky, Brennan de Neeve and Matteo Marinelli. I would like to thank each of them for their useful feedback and advice during the whole time, and especially Brennan for helping me with derivations and debugging code. I would also like to thank Professor Jonathan Home for letting me work in his laboratory, and Professor Andrey Lebedev for explaining the mathematics of the work large parts of my thesis are based on in more detail, so I could derive the expressions I needed.

I am also grateful for all the useful discussions, talks and contributions from all other members of the trapped ion quantum information group at ETH. I got interesting insights in different aspects of quantum computing.

I would also like to thank the NCCR QSIT for awarding me the QSIT Master Internship Award. I was grateful for the financial support this gave me during my work as well as for the interesting talks offered.

Finally, I would like to thank Nikola Djokic for the useful mathematical discussions which helped me refine my algorithms.

Contents

Abstract	4
Acknowledgements	4
1 Introduction	5
2 Ion traps	7
2.1 Qubits	7
2.2 Quantum gates	9
2.3 The DiVincenzo criteria in ion traps	10
2.4 Beryllium	11
2.4.1 State preparation	12
2.4.2 Qubit rotations	12
2.4.3 Readout	12
2.5 Laser setup	13
2.6 Theoretical consideration of single qubit manipulation	15
3 Estimation of a Rabi frequency	18
3.1 Bayesian inference	18
3.2 The task	19
3.3 Phase estimation	20
3.3.1 Distinguishable qubits	21
3.3.2 Indistinguishable qubits	24
3.3.3 The protocol	26
3.4 Implementation	27
3.5 Complexity	30
3.6 Estimation of two frequencies	31
3.6.1 Procedure to estimate two frequencies	33
3.6.2 Implementation details	33

4	Results	35
4.1	Estimation of one Rabi frequency	35
4.1.1	Number of measurements	36
4.1.2	Effect of input parameter	38
4.1.3	Iterative protocol	44
4.2	Estimation of two frequencies	47
5	Outlook	51
	Appendix	53

Chapter 1

Introduction

In the beginning of the 20th century, the theory of quantum mechanics first began to take form in the works of Planck, Einstein, Bohr and many more [15, 5, 3]. A few years later, in the 1980s, the field of quantum computing emerged, initially proposed by Yuri Manin [12] and soon considered by others [6]. What at first was a theoretical consideration soon became reality as the first machines that are able to perform quantum computation were built. Nowadays, there are small-scale quantum computers with varying architectures set up in laboratories all over the world.

A quantum computer uses quantum mechanical systems to perform computation. To be a universal quantum computer, the so called DiVincenzo Criteria must be fulfilled [4]. Namely, a quantum computer has to:

1. be a scalable physical system with a well characterized qubit
2. be able to initialize the qubit to some defined state
3. have long coherence times
4. implement a universal set of quantum gates
5. be able to measure the qubits
6. have the possibility to convert flying to stationary qubits and vice versa
7. reliably transmit flying qubits between locations

One of the architectures which implements all of these criteria is the ion trap. In such a setup, single ions trapped in an electric potential are used as qubits - the fundamental unit of information in quantum computing. This is the architecture used in our laboratory. We currently have the ability to

perform operations on up to three qubits simultaneously. Although there are quantum computers which contain far more qubits ^[9], most of these do not yet fulfill all the DiVincenzo criteria. It is, therefore, still useful to consider smaller systems. The most important topics of current research include quantum error correction. High fidelities are needed to meaningfully carry out fault-tolerant algorithms, which can be improved in small-scale systems and will be required for larger systems to work in the desired way.

Since physical systems demonstrating quantum computation can now be built, it is possible to think about the technical aspects that limit the computation and how one could improve the accuracy of computation. This will ensure that once larger systems fulfilling all DiVincenzo criteria are realized, the solutions to the most basic issues are already known.

In trapped ions, computation is performed by manipulating the states of ions using lasers. However, this coupling is only useful once the laser and ions are aligned properly. The alignment can fluctuate due to external factors, making a daily and sometimes hourly calibration necessary. In our laboratory, some of these calibration steps were so far done by hand, requiring a lot of time that could be used to run higher-level experiments. In this thesis, I will explain how parts of this calibration can be performed in an automated way. In chapter 2 I will give an introduction to both quantum computing and ion traps. I will then describe the calibration task I automated in chapter 3, giving detailed explanations of the algorithm used. Finally, I will present some results of the performance of this calibration and analyze them in chapter 4 before giving a conclusion to this thesis in chapter 5.

Chapter 2

Ion traps

As stated in the introduction, this thesis is written with an ion trap architecture in mind. Thus, it is necessary to understand the basics of trapped ion physics. I will first give a short overview on quantum computing. Then, I will explain how each of the DiVincenzo criteria is implemented in this architecture before presenting our setup, as well as providing a more mathematical discussion on qubit manipulation.

2.1 Qubits

In classical computing, the concept of the bit is well known. It is the basic unit of information that is used to perform computation and can have a value of either 0 or 1, often also referred to as *false* and *true*. Since classical computing is based on classical physics, these are the only states available to the computer. If a classical register contains N bits, then since each of the bits can be in two states, there are 2^N possible states in which the register can be. However, only one of these states is occupied at any given time. During computation, the state realized by the system changes depending on the algorithm that is run.

There is an equivalent to the bit in quantum computing, the qubit, short for quantum bit ^[13]. In this case, we consider the physical unit to be a quantum mechanical two-level system, meaning while unobserved it will be in a state

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ are the so called basis states, and the state of the qubit $|\Psi\rangle$ is a superposition of these states. Only once the state of the qubit is measured does its wave function collapse

to either $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively. This superposition enables a quantum computer to potentially solve certain problems with faster run times than classical computers - for example it is possible to factorize a number in polynomial time ^[17], while the best known algorithm on classical computers is exponential in time.

In the field of quantum information, it is often helpful to think of the state of a qubit represented on the Bloch sphere. For this consideration, one rewrites the previous equation as

$$|\Psi\rangle = e^{i\gamma} \left(\cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \right)$$

The prefactor $e^{i\gamma}$ is a global phase which does not show any experimentally observable effects and can, therefore, be ignored. The angles θ and ϕ describe angles on a unit sphere, the Bloch sphere, as shown in figure 2.1.

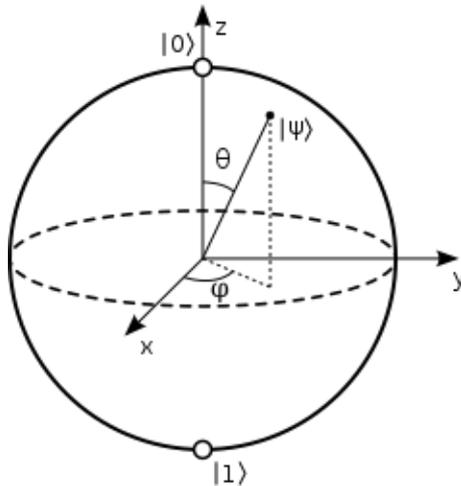


Figure 2.1: The Bloch sphere: the x, y and z axis define three different measurement basis. The angles θ and ϕ uniquely determine the position of the state Ψ on the unit sphere. [Smite-Meister, *Bloch sphere, a geometrical representation of a two-level quantum system*, 30.01.2009, https://commons.wikimedia.org/wiki/File:Bloch_sphere.svg]

This visualization is often useful to consider when talking about single qubit gates or operations. However, it cannot be universally generalized to multiple qubits.

When considering multiple unentangled qubits, the total state of the system is described by the tensor product of the individual states. While this

is not true for entangled qubits, in this thesis I do not consider entangled states since the states produced during the calibrations are separable. For an unentangled two qubit system:

$$|\Psi\rangle = (\alpha_1 |0\rangle_1 + \beta_1 |1\rangle_1) \otimes (\alpha_2 |0\rangle_2 + \beta_2 |1\rangle_2) = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

where the coefficients are of course again such that the wave function is normalized.

2.2 Quantum gates

In classical computing, there are logic gates allowing the user to manipulate the bits by transforming their states. It can be shown that the only gate required to implement all other logic gates is the NAND gate. This gate takes two inputs and gives an output of 0 if and only if both input bits are equal to 1. In all other cases, 1 is returned. The fact that all other logic gates can be converted to NAND gates means that the NAND gate is a universal gate.

In quantum computing, there are gates that also convert the state of a qubit to another state. These gates can be described as unitary transformations. For example, the NOT gate (often called X gate), can be written as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The action of this gate on a qubit state can be understood if we write the basis states as vectors

$$|0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Applying the X gate to our state $|\Psi\rangle$ will result in

$$X|\Psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \left(\alpha \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \alpha |1\rangle + \beta |0\rangle$$

It can be seen that the state $|0\rangle$ gets transformed to $|1\rangle$ and vice versa, which is identical to the action of a NOT gate in classical computation. To facilitate gates on N qubits, one needs to consider $2^N \times 2^N$ matrices. There are several possibilities to construct a universal set of gates for two qubits, one of which is ^[13]

$$\begin{aligned}
H &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\
CNOT &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\
T &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}
\end{aligned}$$

2.3 The DiVincenzo criteria in ion traps

As mentioned before, the DiVincenzo criteria need to be fulfilled by a universal quantum computer.

The system used as a qubit in ion traps is of course a trapped ion. Due to the Earnshaw theorem, it is not possible to trap ions in a static electric field. In our laboratory, we use an electric field that varies with time - a so-called Paul trap - to trap the ions ^[1]. It is possible to trap such ions in a string to obtain more than one qubit, although several other geometries are being researched ^[14]. Using optical pumping - driving a transition to a certain level using a laser - a state can be initialized, which corresponds to the second criterion. To fulfill the third DiVincenzo criterion, it is necessary for the system to exhibit long coherence times with respect to the time it takes to apply gates. The typical time scale for an operation on an ion trap quantum computer is in the order of microseconds, while the decoherence times are usually in the range of milliseconds^[7].

To implement quantum gates, two methods are used. On the one hand, it is possible to use a laser pulse to drive Rabi oscillations between two chosen basis states. This is used for single qubit gates. On the other hand, to implement multi-qubit gates, the long range interaction between the ions due to the Coulomb force is used in combination with laser pulses.

To perform read-out, it is possible to drive a transition between either the ground or excited state and an auxiliary level which exhibits fluorescent decay. It is thus possible to measure the state of a qubit with a photon multiplier tube.

The last two DiVincenzo criteria are fulfilled since ions can be stored in a cavity. This means it is possible to map the ion state to a photonic state, and a photon can be transmitted using fibers [7].

2.4 Beryllium

The experiments in this thesis were performed in a trap which is able to trap ${}^9\text{Be}^+$ and ${}^{40}\text{Ca}^+$ ions. This thesis considers the calibration of experiments in which two ${}^9\text{Be}^+$ ions and one ${}^{40}\text{Ca}^+$ ion are trapped, with the ${}^{40}\text{Ca}^+$ ion being in between the two ${}^9\text{Be}^+$ ions. The nuclear spin angular momentum of ${}^9\text{Be}^+$ is $\mathbf{I} = 3/2$. This results in a splitting of the energy levels of the ion, the most important energy levels of which are shown in figure 2.2.

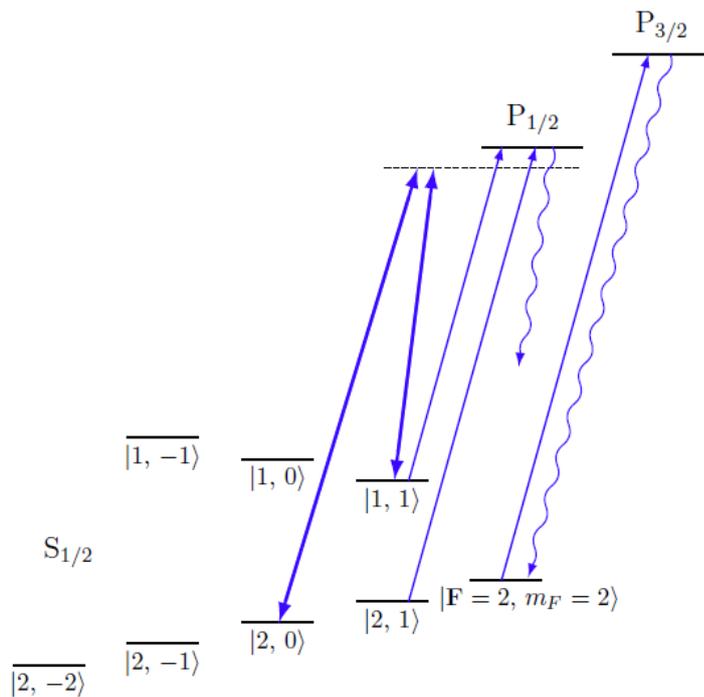


Figure 2.2: The level structure of Beryllium [11]

Here, $\mathbf{F} = \mathbf{I} + \mathbf{L} + \mathbf{S}$ is the total angular momentum quantum number and m_F is its projection along the z-axis. Denote the total orbital quantum

number as \mathbf{L} and the total spin angular momentum quantum number as \mathbf{S} . The excited states P are coupled to the ground states S by a dipole transition with a lifetime of 8.2 ns.

2.4.1 State preparation

Optical pumping is commonly used to initialize qubits in ion traps. In this method, one drives a transition where the ion ends up in a state which will decay rapidly to the desired ground state. Thus, after driving this transition for some time, the ion is in the ground state with high probability [7].

In our trap, the $P_{1/2}$ state decays very quickly to $S_{1/2} |\mathbf{F} = 2, m_F = 1\rangle$, our chosen ground state. By driving the transition between $S_{1/2} |\mathbf{F} = 2, m_F = 1\rangle$ and $P_{1/2}$ as well as the transition between $S_{1/2} |\mathbf{F} = 1, m_F = 1\rangle$ and $P_{1/2}$, we can pump the ion to $S_{1/2} |\mathbf{F} = 2, m_F = 1\rangle$. This transition is driven by a near-resonance 313 nm σ^+ polarized pulse.

2.4.2 Qubit rotations

Qubit rotations are performed by coupling a pair of $S_{1/2}$ states via the $P_{1/2}$ state. We can use the fact that the transition between $S_{1/2} |\mathbf{F} = 2, m_F = 0\rangle$ and $S_{1/2} |\mathbf{F} = 1, m_F = 1\rangle$ has a frequency which is independent of the magnetic field to first order in a magnetic field of $B \approx 119.45\text{G}$. The transition frequency is in this case 1018 MHz. In contrast, the transition between $S_{1/2} |\mathbf{F} = 2, m_F = 2\rangle$ and $S_{1/2} |\mathbf{F} = 1, m_F = 1\rangle$ can be used to describe a frequency-dependent qubit and has a transition frequency of 1207 MHz.

2.4.3 Readout

Finally, we need to be able to perform readout of the qubit state. In general in ion traps, we use the fact that some transitions fluoresce under irradiating light while others do not is used.

Since $S_{1/2} |\mathbf{F} = 2, m_F = 2\rangle$ and $S_{1/2} |\mathbf{F} = 2, m_F = 0\rangle$ are close in frequency, it can happen that light gets scattered from the second state. This can lead to incorrect measurement results. Therefore, we transfer the population of $S_{1/2} |\mathbf{F} = 2, m_F = 0\rangle$ to $S_{1/2} |\mathbf{F} = 1, m_F = -1\rangle$ and the population of $S_{1/2} |\mathbf{F} = 1, m_F = 1\rangle$ to $S_{1/2} |\mathbf{F} = 2, m_F = 2\rangle$. Then, readout is performed as explained above by first irradiating the ion with light and then observing whether or not light is scattered using a photon multiplier tube. This lets us

distinguish between $|0\rangle$ and $|1\rangle$.

In a setup of multiple qubits, it is not possible to distinguish all possible states, namely it is not possible to distinguish between $|01\rangle$ and $|10\rangle$ since the light from all ions is collected in one photon multiplier tube. This means that there are only three possible measurement results: $|00\rangle$, $|01\rangle$ or $|10\rangle$ and $|11\rangle$. This experimental limitation will be of importance later on.

2.5 Laser setup

One set of laser beams is used for state preparation, one for readout and one for qubit rotations. To control these lasers, we use acousto-optic modulators, AOMs. These devices are used to change the frequency of the lasers by using sound waves. In our setup, they are used to send laser pulses as well as to control the amplitude, frequency and phase of the light that reaches the ions. Direct-digital synthesizers, DDSs, generate radio frequency signals that are the input to the AOMs. A DDS allows the user to generate arbitrary sequences of single-frequency rf tones. The light used to control ${}^9\text{Be}^+$ has a wavelength of 313 nm. Our setup uses multiple AOMs to accurately control the parameters of the lasers. A schematic of the setup can be seen in figure 2.3. Simulated Raman transitions are performed by addressing the ions with two lasers simultaneously. The frequency difference of the two lasers should be equal to the qubit transition.

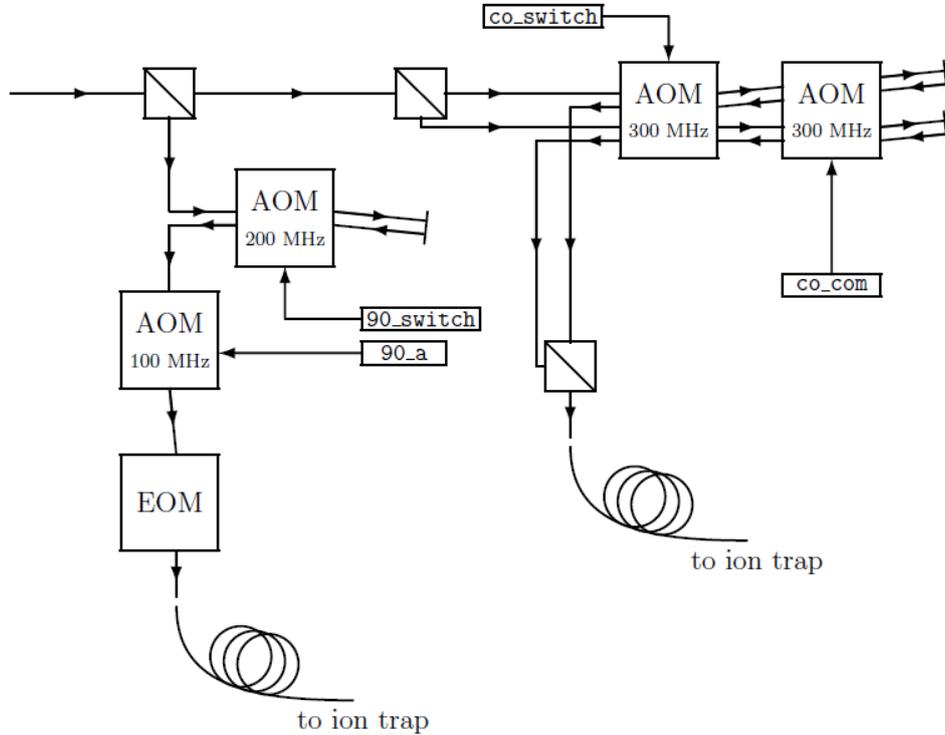


Figure 2.3: The schematic setup of the AOMs and DDSs [11]

For each of the lasers, we can define wave vectors

$$\vec{k}_1 = \frac{2\pi}{\lambda_1} \hat{n}_1$$

$$\vec{k}_2 = \frac{2\pi}{\lambda_2} \hat{n}_2$$

where λ_1 and λ_2 are the wavelengths of the two beams, while \hat{n}_1 and \hat{n}_2 are normal vectors pointing along the direction of the phase velocities of the beam. In order to drive carrier transitions - a type of transition discussed in section 2.6 - we need to overlap the two lasers such that the difference between the wavevectors is minimized. If we, however, would like to drive sideband transitions, we need the difference between wavevectors to have a large component in the direction of the motional axis of the ion.

As can be seen in the figure, the laser light first passes through a polarizing beam splitter, a PBS. Each of the beams resulting from this splitting will be used as one of the beams to address the ion. The transmitted beam passes

through another PBS. Both the transmitted and reflected beam after this pass are sent through two AOMs, both of which are double-passed. These AOMs are addressed by DDS pulses which we call `co_com` and `co_switch`. The prefix `co` emphasizes the fact that the two beams will be co-propagating once they reach the ion. In each AOM, one of the two beams is unaffected, while for the other we choose a reflected beam. After passing the AOMs, the beams are joined using another PBS.

Let us now consider the laser beam which was reflected in the first PBS. This beam first passes through a double-passed AOM addressed by the DDS `90_switch` and then goes through a single-passed AOM addressed by the DDS `90_a`. Since these beams will be propagating at a 90° angle to the the beam coming from the other AOMs, the associated DDS channels have the prefix `90`.

The `90_a` is kept at constant amplitude and 0 frequency. Once a pulse should be sent, the frequency is switched to some non-zero value. To perform carrier transitions, pulses to the `co_com` and `co_switch` are sent, while to perform sideband transitions pulses are sent to the `co_com` and `90_switch` ^[11].

2.6 Theoretical consideration of single qubit manipulation

The derivations in this section are based on the derivations of [7] and [11].

In order to describe the state of a qubit after applying a laser pulse, one needs to solve the Schrödinger equation

$$H |\Psi(\vec{x}, t)\rangle = i\hbar \frac{\partial}{\partial t} |\Psi(\vec{x}, t)\rangle.$$

Here, $|\Psi(\vec{x}, t)\rangle$ is the wave function of the qubit and H is the Hamiltonian.

The internal state of the ion with a qubit transition frequency ω_t , as well as its axial motional state in a motional mode with frequency ω_m are described by the Hamiltonian

$$H = \frac{\hbar\omega_t}{2}\sigma_z + \hbar\omega_m a^\dagger a.$$

Here, σ_z is the Pauli z operator and a and a^\dagger are the creation and annihilation operators.

The interaction Hamiltonian between a qubit transition and near resonance laser light with frequency ω and phase ϕ is given by

$$H_{int} = \frac{\Omega}{2}(\sigma_+ e^{-i\phi} e^{i(\eta(a+a^\dagger)-\omega t)} + \sigma_- e^{i\phi} e^{-i(\eta(a+a^\dagger)-\omega t)}). \quad (2.1)$$

σ_+ and σ_- are the atomic raising and lowering operators acting on the qubit. The Rabi frequency Ω is a measure of the strength of the laser field and will be of importance later on. η is the Lamb-Dicke parameter

$$\eta = \sqrt{\frac{\hbar k_l^2}{2m\omega_m}} \cos(\theta).$$

In this equation, k_l is the wavevector of the laser, m is the mass of the ion and θ is the angle between the motional axis of the ion and the wavevector of the laser. For the experiments presented in this thesis, the Lamb-Dicke parameter is so small that the exponential term in the Hamiltonian can be approximated via a first order Taylor expansion. One can, thus, rewrite 2.1 as

$$H_{int} = \frac{\Omega}{2}(\sigma_+ e^{-i\phi} e^{-i\omega t} + \sigma_- e^{i\phi} e^{i\omega t}) + \frac{i\eta\Omega}{2}(\sigma_+ a e^{-i\phi} e^{-i\omega t} + \sigma_- a^\dagger e^{i\phi} e^{i\omega t}) + \frac{i\eta\Omega}{2}(\sigma_+ a^\dagger e^{-i\phi} e^{-i\omega t} + \sigma_- a e^{i\phi} e^{i\omega t}).$$

Or, when changing the frame of reference:

$$H = \frac{\Omega}{2}(\sigma_+ e^{-i\phi} e^{-i(\omega-\omega_0)t} + \sigma_- e^{i\phi} e^{i(\omega-\omega_0)t}) + \frac{i\eta\Omega}{2}(\sigma_+ a e^{-i\phi} e^{-i(\omega-\omega_0+\omega_m)t} + \sigma_- a^\dagger e^{i\phi} e^{i(\omega-\omega_0+\omega_m)t}) + \frac{i\eta\Omega}{2}(\sigma_+ a^\dagger e^{-i\phi} e^{-i(\omega-\omega_0+\omega_m)t} + \sigma_- a e^{i\phi} e^{i(\omega-\omega_0+\omega_m)t}). \quad (2.2)$$

This Hamiltonian describes how one drives different transitions using the coupling between a laser and the qubit, given that the linewidth of the laser is much narrower than the motional frequency of the qubit. Coupling to the first term leads to an evolution of the qubit's internal state. This is called the *carrier transition* where only the internal ion states are changed, while no motional states are excited. Coupling to the second and third term results in evolution of both the motional state and the internal qubit state, giving the *red* and *blue sideband transitions*.

Since for this thesis only the carrier transition will be of importance, one

can from now on only consider the first part of the Hamiltonian. Define the detuning $\delta = \omega - \omega_0$, assuming $\Omega \ll \omega_m$ and $\delta \ll \omega_m$. This approximation is often called a rotating wave approximation. Inserting this approximation in 2.2, the resulting Hamiltonian takes the form

$$H_{carrier,tot} = \frac{\hbar(\omega_0 + \delta)}{2}\sigma_z + \hbar\omega_m a^\dagger a,$$

leading to a time-independent Hamiltonian

$$H_{carrier} = \frac{\hbar\delta}{2}\sigma_z + \frac{\Omega}{2}(\sigma_+ e^{-i\phi} + \sigma_- e^{i\phi}).$$

From this, one obtains a time evolution of the qubit's internal state given by

$$U(t) = \cos\left(\frac{\Omega_{eff}t}{2}\right)\mathbb{1} - \frac{i}{\Omega_{eff}}\sin\left(\frac{\Omega_{eff}t}{2}\right)[\delta\sigma_z + \Omega_{eff}(\cos(\phi)\sigma_x + \sin(\phi)\sigma_y)].$$

The effective Rabi frequency given by $\Omega_{eff} = \sqrt{\Omega^2 + \delta^2}$ was introduced here.

However, for an arbitrary single-qubit gate, one can tune the laser frequency such that $\delta \approx 0$, which simplifies this equation to

$$U_{pulse}(\Omega t, \phi) = \cos\left(\frac{\Omega t}{2}\right)\mathbb{1} - i\sin\left(\frac{\Omega t}{2}\right)(\cos(\phi)\sigma_x + \sin(\phi)\sigma_y). \quad (2.3)$$

On the Bloch sphere, this describes the rotation around the axis $\cos(\phi)\hat{x} + \sin(\phi)\hat{y}$ with frequency Ω , the Rabi frequency. It can be seen that if the Rabi frequency as well as phase of the laser are known, one can prepare the ion in different states using appropriate times t . While the phase of the laser is easy to set, the Rabi frequency depends on the laser intensity at the ion, which in turn depends on the alignment between laser and ion, as well as the transition properties. The remainder of this thesis will discuss a possible way to perform this alignment.

Chapter 3

Estimation of a Rabi frequency

3.1 Bayesian inference

The task of statistical inference is to estimate the distribution of some parameters given data which depends on these parameters. Bayes' theorem is well known from probability theory and can be used to perform inference:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)} \quad (3.1)$$

Here, θ is the parameter (or set of parameters) that should be inferred and X is the data measured. The left side of equation 3.1 corresponds to the posterior, the distribution of the parameter after taking X into account. The likelihood $p(X|\theta)$ is the probability that one measures data X given some parameters θ and $p(\theta)$ is the prior distribution of the parameters, not taking into account the data. This is normalized by the marginal distribution of the data, i.e. the probability of obtaining data X independent of the parameters θ .

If we define some initial distribution for our parameters, which can come from an educated guess, we can update this distribution to support data that we measure. This process can be repeated multiple times to give a good estimate of the true distribution of the parameters. However, there are some limitations. Namely, it is necessary for the right hand side to be a closed expression. It is possible for this not to be the case in many ways. First, the likelihood might not have an analytic form or might be very expensive to evaluate. However, even if the likelihood has an analytical formula, it is still possible for Bayes' theorem 3.1 to not be a closed expression depending on the prior chosen and depending on whether the fraction can be simplified

to again represent a probability distribution. In the general case, it is not possible to derive a closed expression for equation 3.1 and different inference schemes need to be investigated, especially for more complex models.

As I will explain in sections 3.2 and 3.3, in our case we choose a Fourier series as our prior:

$$p(\theta) = \sum_{n=-\infty}^{\infty} c_n e^{in\theta}$$

To update such a distribution according to Bayes' theorem, it is necessary to derive an overall relation between the Fourier coefficients of the prior and of the posterior. The likelihood which emerges from our experimental setup provides such a relation and lets us infer our parameter of interest with a very simple scheme.

3.2 The task

We are considering a setup of two Beryllium ions and one Calcium ion. They are arranged in a string with the Calcium ion between the two Beryllium ions. Assume that the evolution given by a laser pulse acting on both Beryllium ions simultaneously can be described by the tensor product of two evolutions 2.3:

$$U_{total}(\Omega_1 t, \Omega_2 t, \phi) = U_{pulse}(\Omega_1 t, \phi) \otimes U_{pulse}(\Omega_2 t, \phi)$$

The laser pulse will be applied to both ions for the same amount of time. We would like to find a pulse that lets us rotate the state of both ions by the same angle, for example by π . This rotation not only depends on the time, but also on the Rabi frequencies of both ions, which depend on the laser intensity at the ion and thus on the alignment between the laser and ions. The phase ϕ of the laser can be set to an arbitrary value and can, therefore, be ignored. We would like to estimate the Rabi frequencies of the two ions for some given beam alignment. If this is possible, we can then align the laser such that both Rabi frequencies are equal and a pulse applied for a time t_π will indeed cause a rotation of both ion states by π .

The protocol in section 3.3 is derived from a protocol already implemented in our lab for one ion. It is based on work done by Andrey Lebedev ^[10]. It uses adaptive robust phase estimation (ARPE) to estimate a phase related to the Rabi frequency. ARPE is similar to robust phase estimation (RPE), which implements the estimation of single-qubit gate parameters. While RPE applies a fixed number of rotations for every measurement shot of the

experiment, ARPE replaces this with an adaptive procedure. Our laboratory has implemented RPE as well as ARPE to estimate the Rabi frequency of a single ion. However, these protocols differ from the ones described in this chapter since two ions are used here. While the overall idea remains the same, the mathematics needs to be adjusted for the two qubit case.

3.3 Phase estimation

This section presents a protocol to estimate a phase θ_0 . Later on, an experimental setup which links this phase to the Rabi frequency of one of the two qubits in use will be described. The other Rabi frequency has to be estimated or *known* by the user and is given as an input to the algorithm. This known Rabi frequency can again be linked to a known phase. Of course, the more accurate the estimate given is, the better the protocol will perform the estimation of the second Rabi frequency. In this section, the known phase will be called θ_{known} and the current estimate of the phase θ .

In subsection 3.3.1, I will discuss how the estimation of this phase can be performed if the qubits are distinguishable. Then, in subsection 3.3.2 the same will be done in case the qubits are not distinguishable - in other words, the states $|01\rangle$ and $|10\rangle$ can not be distinguished.

In both of these cases, the knowledge of θ can be represented at some step $j - 1$ of the protocol by a Fourier series:

$$P_{j-1}(\theta) = \sum_{n=-\infty}^{\infty} c_n^{(j-1)} e^{in\theta} \quad (3.2)$$

Taking this equation as the prior, one can calculate the posterior according to Bayes' theorem as

$$P_j(\theta|\alpha, \epsilon) = \frac{p(\alpha, \epsilon|\theta)P_{j-1}(\theta)}{\int_0^{2\pi} \frac{1}{2\pi} p(\alpha, \epsilon|\theta)P_{j-1}(\theta)d\theta}, \quad (3.3)$$

where the angle at which the measurement is performed is denoted as α and the outcome of a measurement as ϵ . I will now investigate the exact form of this theorem for distinguishable as well as indistinguishable qubits.

3.3.1 Distinguishable qubits

Assume that we can distinguish all measurement results: $|00\rangle$, $|10\rangle$, $|01\rangle$ and $|11\rangle$. Although not immediately useful for our current setup, the calculations presented will be similar to the ones used for the case where some measurement results are indistinguishable. Moreover, an experimental goal of our laboratory is to make the measurement outcomes all distinguishable, allowing us to use the results in this section.

In this situation, one can rewrite equation 3.3 as

$$P_j(\theta|\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)}) = \frac{p(\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)}|\theta)P_{j-1}(\theta)}{\int_0^{2\pi} \frac{1}{2\pi} p(\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)}|\theta)P_{j-1}(\theta)d\theta}, \quad (3.4)$$

where ξ_1^j and ξ_2^j represent the spin of the measurement outcome at the j th step for qubit 1 and 2 respectively (i.e. a measurement of $|0\rangle$ corresponds to $\xi = 1$ and a measurement of $|1\rangle$ corresponds to $\xi = -1$).

Assume that the experiment is set up such that the probability of measurement outcomes is equal to

$$p(\alpha, \xi_1, \xi_2, \theta, \theta_{known}) = \frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2}.$$

One can now calculate the denominator of equation 3.4 - the marginal distribution, here denoted as $\Gamma(\alpha, \xi_1, \xi_2)$ - as

$$\Gamma(\alpha, \xi_1^{(j)}, \xi_2^{(j)}) = \int_0^{2\pi} \frac{1}{2\pi} \frac{1 + \xi_1^{(j)} \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2^{(j)} \cos(\alpha - \theta)}{2} P_{j-1}(\theta) d\theta = \frac{1 + \xi_1^{(j)} \cos(\alpha - \theta_{known})}{2} \left(\frac{1}{2} + \frac{\xi_2^{(j)}}{4} \left(c_{-1}^{(j-1)} e^{-i\alpha} + c_1^{(j-1)} e^{i\alpha} \right) \right).$$

One can, therefore, write equation 3.4 as

$$P_j(\theta|\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)}) = \frac{p(\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)}|\theta)P_{j-1}(\theta)}{\Gamma(\alpha, \xi_1^{(j-1)}, \xi_2^{(j-1)})}. \quad (3.5)$$

Inserting equation 3.2 for the prior as well as using

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2}$$

an update rule for the Fourier coefficients can now be formulated:

$$\begin{aligned}\tilde{c}_n^{(j)} &= c_n^{(j-1)} + \frac{\xi_2^{(j)}}{2} (e^{i\alpha} c_{n+1}^{(j-1)} + e^{-i\alpha} c_{n-1}^{(j-1)}) \\ c_n^{(j)} &= \frac{\tilde{c}_n^{(j)}}{\tilde{c}_0^{(j)}}.\end{aligned}\tag{3.6}$$

For this updating to work optimally, one can choose the measurement angle α such that the expected entropy gain is maximized, in accordance with the principle of maximum entropy [8]. Here, define entropy as the Shannon entropy

$$S[P(\theta)] = - \int_0^{2\pi} \frac{1}{2\pi} P(\theta) \ln \left[\frac{P(\theta)}{2\pi} \right] d\theta.$$

The expected entropy gain for the j th step is then defined by

$$\Delta_j S(\alpha) = \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \Gamma(\alpha, \xi_1, \xi_2) (S[P_{j-1}(\theta)] - S[P_j(\theta|\alpha, \xi_1, \xi_2)]).$$

Inserting equation 3.5, one obtains

$$\begin{aligned}\Delta_j S(\alpha) &= \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \Gamma(\alpha, \xi_1, \xi_2) \int_0^{2\pi} d\theta \frac{1}{2\pi} \left(P_{j-1}(\theta) \ln \left(\frac{P_{j-1}(\theta)}{2\pi} \right) - \right. \\ &\quad \left. \frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \frac{1}{\Gamma(\alpha, \xi_1, \xi_2)} P_{j-1}(\alpha) \right. \\ &\quad \left. \left[\ln \left(\frac{P_{j-1}(\theta)}{2\pi} \right) + \ln \left(\frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \frac{1}{\Gamma(\alpha, \xi_1, \xi_2)} \right) \right] \right).\end{aligned}\tag{3.7}$$

Due to the summation over all possible ξ , the first and second logarithm term will sum to 0, resulting in

$$\begin{aligned}
\Delta_j S(\alpha) &= \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \int_0^{2\pi} d\theta \frac{1}{2\pi} \frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} P_{j-1}(\theta) \\
&\quad \ln \left(\frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \frac{1}{\Gamma(\alpha, \xi_1, \xi_2)} \right) \\
&= \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \int_0^{2\pi} d\theta \frac{1}{2\pi} P_{j-1}(\theta) \frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \\
&\quad \ln \left[\frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \right] - \\
&\quad \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \int_0^{2\pi} d\theta \frac{1}{2\pi} P_{j-1}(\theta) \ln[\Gamma(\alpha, \xi_1, \xi_2)] \\
&= \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \int_0^{2\pi} d\theta \frac{1}{2\pi} P_{j-1}(\theta) \frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \\
&\quad \left(\ln \left[\frac{1 + \xi_1 \cos(\alpha - \theta_{known})}{2} \right] + \ln \left[\frac{1 + \xi_2 \cos(\alpha - \theta)}{2} \right] \right) - \\
&\quad \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \int_0^{2\pi} d\theta \frac{1}{2\pi} P_{j-1}(\theta) \ln[\Gamma(\alpha, \xi_1, \xi_2)].
\end{aligned}$$

Observing that the cosine term in front of the integral will add up to 1, inserting the Taylor expansion of the logarithm for the second logarithm in the first term, observing that the second term is equal to $\Gamma(\alpha, \xi_1, \xi_2)$ and inserting equation 3.2, one gets

$$\Delta_j S(\alpha) = 1 - 2\ln(2) + \sum_{n=1}^{\infty} \frac{\text{Re}\{c_{2n}^{(j-1)} e^{2in\alpha}\}}{n(4n^2 - 1)} - \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \Gamma(\alpha, \xi_1, \xi_2) \ln(\Gamma(\alpha, \xi_1, \xi_2)). \quad (3.8)$$

Thus, the optimal measurement angle to use at the j th step α_i is equal to

$$\alpha_j = \alpha : \max \Delta_j S(\alpha).$$

To maximize the entropy gain, any algorithm that finds maxima or minima can be used. In my implementation, the golden-section search is used.

By repeatedly measuring the ions, updating the Fourier coefficients according to equations 3.6, and finding the measurement angle which maximizes entropy gain according to equation 3.8, we can generate a posterior

which accurately models the distribution of the phase. This protocol will be described in more detail, however, it is useful to first consider the case of indistinguishable qubits.

3.3.2 Indistinguishable qubits

The calculation presented in subsection 3.3.1 requires the possibility to distinguish the two measurements $|01\rangle$ and $|10\rangle$. However, this is not currently possible in our laboratory setup. Instead, readout of the qubits only provides the information that one of the measurement results $\{|00\rangle, |11\rangle, \{|01\rangle \text{ or } |10\rangle\}$ was observed. Due to the indistinguishability of $|01\rangle$ and $|10\rangle$, I present a way to perform a weighted updating in this section.

Again Bayes' theorem 3.3 is considered. However, the likelihood $p(\alpha, \epsilon|\theta)$ will differ from the distinguishable case since only three different measurement results can be obtained.

For indistinguishable qubits, the probabilities of each outcome for the measurement will be

$$\begin{aligned}
p_{|00\rangle}(\alpha|\theta) &= \frac{1 + \cos(\alpha - \theta_{known})}{2} \frac{1 + \cos(\alpha - \theta)}{2} \\
p_{|01\rangle \text{ or } |10\rangle}(\alpha|\theta) &= \frac{1 + \cos(\alpha - \theta_{known})}{2} \frac{1 - \cos(\alpha - \theta)}{2} + \\
&\quad \frac{1 - \cos(\alpha - \theta_{known})}{2} \frac{1 + \cos(\alpha - \theta)}{2} \\
p_{|11\rangle}(\alpha|\theta) &= \frac{1 - \cos(\alpha - \theta_{known})}{2} \frac{1 - \cos(\alpha - \theta)}{2}.
\end{aligned} \tag{3.9}$$

Introduce some notation:

$$\begin{aligned}
\zeta^+ &= \frac{c_0^{(j-1)} + \text{Re}(c_1^{(j-1)} e^{i\alpha})}{2} = \frac{1 + \text{Re}(c_1^{(j-1)} e^{i\alpha})}{2} \\
\zeta^- &= \frac{c_0^{(j-1)} - \text{Re}(c_1^{(j-1)} e^{i\alpha})}{2} = \frac{1 - \text{Re}(c_1^{(j-1)} e^{i\alpha})}{2}.
\end{aligned}$$

The denominator of equation 3.3 will now be equal to

$$\begin{aligned}
\Gamma(\alpha) &= \frac{1 + \cos(\alpha - \theta_{known})}{2} \zeta^+ \\
\Gamma(\alpha) &= \frac{1 + \cos(\alpha - \theta_{known})}{2} \zeta^- + \frac{1 - \cos(\alpha - \theta_{known})}{2} \zeta^+ \\
\Gamma(\alpha) &= \frac{1 - \cos(\alpha - \theta_{known})}{2} \zeta^-.
\end{aligned}$$

Observe that for the measurements of $|00\rangle$ and $|11\rangle$, the rule to update the Fourier coefficients of the distribution will be the same as in subsection 3.3.1. However, for a measurement of $|01\rangle$ or $|10\rangle$, one obtains

$$\begin{aligned}\tilde{c}_n^{(j)} &= c_n^{(j-1)} - \frac{\cos(\alpha - \theta_{known})}{2} (e^{i\alpha} c_{n+1}^{(j-1)} + e^{-i\alpha} c_{n-1}^{(j-1)}) \\ c_n^{(j)} &= \frac{\tilde{c}_n^{(j)}}{\tilde{c}_0^{(j)}}.\end{aligned}\tag{3.10}$$

Due to the changes regarding the likelihood, it is also necessary to recalculate the entropy gain. Calling the three possible measurement outcomes χ , one can write down an expression equivalent to equation 3.7:

$$\begin{aligned}\Delta_j S(\alpha) &= \int_0^{2\pi} \frac{d\theta}{2\pi} P_{j-1}(\theta) \ln\left(\frac{P_{j-1}(\theta)}{2\pi}\right) \sum_{\chi} \Gamma(\chi, \alpha) \left(\frac{p(\chi, \alpha, \theta_{known}, \theta)}{\Gamma(\chi, \alpha)} - 1\right) + \\ &\sum_{\chi} \Gamma(\chi, \alpha) \left(\int_0^{2\pi} \frac{d\theta}{2\pi} \frac{p(\chi, \alpha, \theta_{known}, \theta) P_{j-1}(\theta)}{\Gamma(\chi, \alpha)} [\ln(p(\chi, \alpha, \theta_{known}, \theta)) - \ln(\Gamma(\chi, \alpha))]\right).\end{aligned}$$

Again, the first summation over χ will equal 0. This leads to an expression similar to the entropy gain 3.8:

$$\Delta_j S(\alpha) = \sum_{\chi} \int_0^{2\pi} \frac{d\theta}{2\pi} p(\chi, \alpha, \theta_{known}, \theta) P_{j-1}(\theta) \ln(p(\chi, \alpha, \theta_{known}, \theta)) - \sum_{\chi} \Gamma(\chi, \alpha) \ln(\Gamma(\chi, \alpha)).$$

It can be shown that this integral, similar to its counterpart in subsection 3.3.1, can be solved analytically. The obtained result is

$$\begin{aligned}
\Delta_j S(\alpha) = & -\frac{1 + \cos(\alpha - \theta_{known})}{2} \zeta^+ \ln(\zeta^+) - \\
& \frac{1 - \cos(\alpha - \theta_{known})}{2} \zeta^- \ln(\zeta^-) - \\
& \left(\frac{1 + \cos(\alpha - \theta_{known})}{2} \zeta^- + \frac{1 - \cos(\alpha - \theta_{known})}{2} \zeta^+ \right) \\
& \ln \left(\frac{1 + \cos(\alpha - \theta_{known})}{2} \zeta^- + \frac{1 - \cos(\alpha - \theta_{known})}{2} \zeta^+ \right) + \\
& \frac{1}{2} \left(1 - 4 \ln(2) + \right. \\
& \left. \cos(\alpha - \theta_{known}) \left[\frac{\cos(\alpha - \theta_{known}) + (\zeta^+ - \zeta^-) \sin(\alpha - \theta_{known})}{1 + \sin(\alpha - \theta_{known})} \right] + \right. \\
& \left. (1 - \cos(\alpha - \theta_{known})) (\zeta^+ - \zeta^-) \ln(1 + \sin(\alpha - \theta_{known})) \right) + \\
& \sum_{m=2}^{\infty} \left(\frac{1}{2} (1 - \cos(\alpha - \theta_{known}) + (1 + \cos(\alpha - \theta_{known})) (-1)^m) + \right. \\
& \left. (1 + m \sin(\alpha - \theta_{known})) \left[\frac{\cos(\alpha - \theta_{known})}{1 + \sin(\alpha - \theta_{known})} \right]^m \right) \frac{\text{Re}\{c_m^{(j-1)} e^{im\alpha}\}}{m(m^2 - 1)}.
\end{aligned} \tag{3.11}$$

As for the distinguishable case, this expression should be maximized.

3.3.3 The protocol

The overall algorithm to estimate a phase is divided into steps. At each step of the protocol, the user defines the number of measurement shots that should be performed. For each *shot*, the optimal measurement angle is calculated by maximizing the expected entropy gain. Then, a measurement is performed and the Fourier coefficients are updated. After each *step*, the distribution is transformed to further increase accuracy^[11]. In the end, the protocol outputs the phase estimate. Due to the construction of our Fourier series in equation 3.2, this phase is given by

$$\theta = \arg \int_0^{2\pi} d\theta' \frac{1}{2\pi} P(\theta') e^{i\theta'} = \arg(c_{-1}). \tag{3.12}$$

A step of phase estimation can be written in pseudo code as follows in algorithm 1

Algorithm 1 Adaptive Fixed Rotation Phase Estimation

```
for  $shot \leftarrow 1$  to  $N$  do
  if  $shot == 1$  and prior is uniform then
     $\alpha \leftarrow 0$ 
  else
    find optimal  $\alpha$  by maximizing the entropy gain 3.8 or 3.11
  end if
  Measure
  Update coefficients according to 3.6 or 3.10
end for
return  $\theta$  given by 3.12
```

3.4 Implementation

Given the protocol from section 3.3, one can now implement a phase estimation to find the Rabi frequency of one ion given the Rabi frequency of the other. For this, one needs to specify an initial set of Fourier coefficients. In my case, I choose a Gaussian distribution for each Fourier coefficient:

$$c_n = e^{-2\sigma^2 n^2 - in\mu}$$

where $\sigma = \frac{\pi}{8}$ is the standard deviation and $\mu = \pi$ is the mean of the distribution.

Please note that since $c_{-n} = c_n^*$, there is no need to save the negative Fourier coefficients during computation. Additionally, laser pulses which fulfill equation 3.9 are needed. One such pulse sequence is

$$U_{total} \left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2} \right) U_{total} \left(\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2} - \alpha \right) U_{total} (k_j \Omega_1 t_{pulse}, k_j \Omega_2 t_{pulse}, -\alpha).$$

Here, the third unitary is a unitary applied for a time $k_j t_{pulse}$ with Rabi frequencies Ω_1 and Ω_2 , where one of the two Rabi frequencies should be estimated and the other one is known. If one ion is considered to be in the ground state, applying the unitary transformation will lead to the state

$$\begin{aligned}
& U_{pulse} \left(\frac{\pi}{2}, \frac{\pi}{2} \right) U_{pulse} \left(\frac{\pi}{2}, -\frac{\pi}{2} - \alpha \right) U_{pulse} (k_j \Omega_1 t_{pulse}, -\alpha) |0\rangle \\
&= U_{pulse} \left(\frac{\pi}{2}, \frac{\pi}{2} \right) U_{pulse} \left(\frac{\pi}{2}, -\frac{\pi}{2} - \alpha \right) \left(\cos \left(\frac{\alpha - k_j \Omega_1 t_{pulse}}{2} \right) |0\rangle \right. \\
&\quad \left. + i \sin \left(\frac{\alpha - k_j \Omega_1 t_{pulse}}{2} \right) |1\rangle \right) \\
&\quad = U_{pulse} \left(\frac{\pi}{2}, \frac{\pi}{2} \right) \frac{1}{\sqrt{2}} (|0\rangle - e^{-i(\alpha - k_j \Omega_1 t_{pulse})} |1\rangle) \\
&= \cos \left(\frac{\alpha - k_j \Omega_1 t_{pulse}}{2} \right) |0\rangle + i \sin \left(\frac{\alpha - k_j \Omega_1 t_{pulse}}{2} \right) |1\rangle,
\end{aligned}$$

which leads to an outcome probability of

$$p(\alpha, \xi) = \frac{1 + \xi \cos(\alpha - k_j \Omega_1 t_{pulse})}{2}.$$

Now assume both ions to be independent. Therefore, to obtain the outcome probabilities of the pulse sequence mentioned above, one can simply multiply the individual outcome probabilities given above:

$$p(\alpha, \xi_1, \xi_2) = \frac{1 + \xi_1 \cos(\alpha - k_j \Omega_1 t_{pulse})}{2} \frac{1 + \xi_2 \cos(\alpha - k_j \Omega_2 t_{pulse})}{2}.$$

This is equivalent to equation 3.9 with $\theta_{known} = k_j \Omega_1 t_{pulse}$ and $\theta = k_j \Omega_2 t_{pulse}$ (if the outcomes for measurements $|01\rangle$ and $|10\rangle$ are added together). The protocol will estimate $\theta = k_j A$, where $A = \Omega t_{pulse}$. This means that the Rabi frequency which should be estimated is related to the estimated phase as

$$\Omega = \frac{\theta}{k_j t_{pulse}}.$$

The protocol is run for K steps. For each step $j \in [1, K]$, one can choose $k_j = 2^{j-1}$. After every step of the protocol, one gets an estimate $\widehat{k_j A}$ of the phase $k_j A$ modulo 2π . One then computes an estimate $\hat{A}_j = \frac{\widehat{k_j A}}{k_j}$ restricted to the range $(\hat{A}_{j-1} - \frac{\pi}{2^{j-1}}, \hat{A}_{j-1} + \frac{\pi}{2^{j-1}}]$ [16]. One then also shift the distribution 3.2 by some angle $\beta = k_{j+1} \hat{A}_j - \arg(c_{-1})$ which transforms the Fourier coefficients as

$$c_n \rightarrow c_n e^{-in\beta}.$$

In addition, one increases the standard deviation of the overall distribution by a factor a , meaning the coefficients transform as

$$c_n \rightarrow |c_n|^{a^2-1} c_n.$$

The pseudo code in algorithm 2 shows the complete protocol.

Algorithm 2 Adaptive Robust Phase Estimation

```

for  $j \leftarrow 1$  to  $K$  do
   $k_j \leftarrow 2^{j-1}$ 
  choose number of shots  $S_j$ 
  for  $s \leftarrow 1$  to  $S_j$  do
    perform adaptive fixed rotation phase estimation as described in
    algorithm 1
  end for
  if  $j=1$  then
     $\hat{A}_j \leftarrow \arg(c_{-1})$ 
  else
     $\hat{A}'_j \leftarrow \arg(c_{-1})/k_j$ 
     $q_1 \leftarrow \left\lfloor \frac{\hat{A}'_j}{2\pi/k_j} \right\rfloor$ 
     $q_2 \leftarrow q_1 + 1$ 
    if  $q_1 2\pi/k_j + \hat{A}'_j \in (\hat{A}_{j-1} - \pi/k_j, \hat{A}_{j-1} + \pi/k_j]$  then
       $\hat{A}_j \leftarrow q_1 2\pi/k_j + \hat{A}'_j$ 
    else if  $q_2 2\pi/k_j + \hat{A}'_j \in (\hat{A}_{j-1} - \pi/k_j, \hat{A}_{j-1} + \pi/k_j]$  then
       $\hat{A}_j \leftarrow q_2 2\pi/k_j + \hat{A}'_j$ 
    else
       $\hat{A}_j \leftarrow \hat{A}'_j + \pi/k_j$ 
    end if
  end if
   $\beta \leftarrow k_{j+1} \hat{A}_j - \arg(c_{-1})$ 
   $c_n \leftarrow c_n e^{-in\beta}$ 
   $c_n \leftarrow |c_n|^{a^2-1} c_n$ 
end for

```

A few additional things need to be considered when implementing the procedure as explained above. First, it is of course not possible to have an infinite number of Fourier coefficients, thus we let m be the number of Fourier coefficients. If m is fixed, then the updating rule 3.6 is not always defined, since c_{m+1} is not defined if $n == m$ (note that c_{n-1} is always defined since

$c_{-n} = c_n^*$). In my implementation, I decided to set $c_{m+1} = 0$.

Second, if one wishes to implement the protocol based on distinguishable qubits, but with an experimental realization of indistinguishable qubits, it is necessary to decide after each measurement where $|01\rangle$ or $|10\rangle$ is measured whether one performs the update of the Fourier coefficients as if $|01\rangle$ or $|10\rangle$ was measured. This can be done by assuming that the probability to being in the state $|01\rangle$ after having measured $|01\rangle$ or $|10\rangle$ is equal to $\frac{1+\cos(\alpha-\theta_{known})}{2}$. Therefore, with this probability we update the Fourier coefficients as though $|01\rangle$ had been measured. However, this introduces randomness into the estimation which may result in wrong estimations.

3.5 Complexity

Let us analyze the complexity of the algorithm described above to roughly estimate the runtime.

The user can choose the number of phase approximation steps as well as how many shots he wants to perform at each step. Call the number of steps K and the total number of shots Km . For each of these Km shots, the optimal measurement angle is calculated by first finding an optimal starting value for the gain and then finding the optimal angle using the golden-section search. The optimal starting value is found by calculating the Shannon entropy for 10 different angles.

To calculate the Shannon entropy, one needs to add up $\frac{n}{2}$ values with n being the number of Fourier coefficients. Then, one calculates all possible measurement outcome probabilities, the complexity of which is $\mathcal{O}(1)$. Finding the optimal starting value, therefore, has complexity $\mathcal{O}(n)$. The golden-section search has complexity $\mathcal{O}(\log(\frac{1}{\epsilon}))$ [2] with ϵ being the precision threshold set, in our case $\epsilon = 10^{-9}$. Thus, the complexity of finding the optimal angle is $\mathcal{O}(n)$.

Next, a measurement is performed. Assume the time this takes to be constant. Then, the Fourier coefficients are updated. Each coefficient is updated individually, giving a complexity of $\mathcal{O}(n)$. After every step, the current estimated phase is calculated. Updating the phase is $\mathcal{O}(1)$. However, the distribution is also shifted by some factor and its standard deviation increased by a factor at this point in the algorithm, which requires updating the Fourier coefficients in $\mathcal{O}(n)$.

Therefore, the algorithm has a complexity of $\mathcal{O}(Kmn)$ for both calculat-

ing the optimal angle and updating the Fourier coefficients, and additionally a complexity of $\mathcal{O}(Kn)$ to update the phase. This gives an overall complexity of $\mathcal{O}(Kmn)$.

Since all these computations are classical - the quantum part being only the application of laser pulses to ions and measuring them - this can be used to estimate the run time of the algorithm. For this, one needs to estimate the number of operations a compiled C++ code can perform per second. This is of course CPU dependent, however, a common benchmark is 10^6 operations per second. If one assumes values for each variables which were used in testing this protocol, $n = 100$, $K = 8$ and $m = 50$, this will result in $8 \cdot 50 \cdot 100 = 40000 = 4 \cdot 10^4$ operations, meaning the complete estimation should take on the order of a second or less. Note that these calculations do not take the time needed by the experiment into account. This is just the classical expected run time. The algorithm will take additional time to apply the pulse sequence and perform measurements.

3.6 Estimation of two frequencies

The protocol described in subsection 3.3.3 allows for an estimation of one of two Rabi frequencies given an estimate for the second Rabi frequency. However, it would be desirable to estimate both Rabi frequencies simultaneously. This will be explored in the following section.

Call the phases that should be estimated $\theta_{1,0}$ and $\theta_{2,0}$ and the current estimation of the respective phases θ_1 and θ_2 .

The knowledge of the phases at some step $j - 1$ can again be represented by a Fourier series. However, this time it is a 2-dimensional Fourier series:

$$P_{j-1}(\theta_1, \theta_2) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} c_{n,m}^{(j-1)} e^{in\theta_1} e^{im\theta_2}.$$

For the likelihood, assume distinguishable qubits. Since an implementation which allows us to distinguish both qubits will be implemented in our lab in the near future, this will be the case most useful to us. It is also difficult to calculate the expected entropy gain for indistinguishable qubits, making this protocol more attractive as a starting point. The probability of measurement outcomes will be

$$p(\alpha, \xi_1, \xi_2, \theta_1, \theta_2) = \frac{1 + \xi_1 \cos(\alpha - \theta_1)}{2} \frac{1 + \xi_2 \cos(\alpha - \theta_2)}{2}.$$

The denominator of Bayes' theorem 3.3 will then differ from the previous cases in that the integral is now 2-dimensional over both θ_1 and θ_2 . One

obtains

$$\begin{aligned}
\Gamma(\alpha, \xi_1, \xi_2) = & \frac{1}{2} \frac{\xi_1}{4} [c_{-1,0}^{(j-1)} e^{-i\alpha} + c_{1,0}^{(j-1)} e^{i\alpha}] + \\
& \frac{1}{2} \frac{\xi_2}{4} [c_{0,-1}^{(j-1)} e^{-i\alpha} + c_{0,1}^{(j-1)} e^{i\alpha}] + \\
& \frac{1}{4} c_{0,0}^{(j-1)} + \frac{1}{4} \frac{\xi_1 \xi_2}{4} [c_{-1,-1}^{(j-1)} e^{-2i\alpha} + c_{-1,1}^{(j-1)} + \\
& c_{1,-1}^{(j-1)} + c_{1,1}^{(j-1)} e^{2i\alpha}].
\end{aligned} \tag{3.13}$$

Using equation 3.13, an update rule for the Fourier coefficients can be derived in the same way it is done for one Rabi frequency

$$\begin{aligned}
\tilde{c}_{n,m}^{(j)} = & c_{n,m}^{(j-1)} + \frac{\xi_1}{2} [c_{n-1,m}^{(j-1)} e^{-i\alpha} + c_{n+1,m}^{(j-1)} e^{i\alpha}] + \\
& \frac{\xi_2}{2} [c_{n,m-1}^{(j-1)} e^{-i\alpha} + c_{n,m+1}^{(j-1)} e^{i\alpha}] + \\
& \frac{\xi_1 \xi_2}{4} [c_{n-1,m-1}^{(j-1)} e^{-2i\alpha} + c_{n-1,m+1}^{(j-1)} + c_{n+1,m-1}^{(j-1)} + c_{n+1,m+1}^{(j-1)} e^{2i\alpha}] \\
c_{n,m}^{(j)} = & \frac{\tilde{c}_{n,m}^{(j)}}{\tilde{c}_{0,0}^{(j)}}.
\end{aligned}$$

The equation for the entropy gain has to be adjusted in a similar way. The Shannon entropy now also contains two integrals over both θ_1 and θ_2 . Using the same approach as for the estimation of one Rabi frequency, one obtains

$$\begin{aligned}
\Delta_j S(\alpha) = & 2 - 4\ln(2) + \sum_{m=1}^{\infty} \frac{\operatorname{Re} \left(c_{2m,0}^{(j-1)} \right) e^{2im\alpha}}{m(4m^2 - 1)} + \\
& \sum_{m=1}^{\infty} \frac{\operatorname{Re} \left(c_{0,2m}^{(j-1)} \right) e^{2im\alpha}}{m(4m^2 - 1)} \\
& - \sum_{\xi_1=\pm 1} \sum_{\xi_2=\pm 1} \Gamma(\alpha, \xi_1, \xi_2) \ln(\Gamma(\alpha, \xi_1, \xi_2)).
\end{aligned}$$

The adaptive fixed rotation phase estimation will now be very similar to the algorithm described in subsection 3.3.3. The only difference will be the actual implementation of the updating. The estimated phases are now given by

$$\begin{aligned}\theta_1 &= \arg(c_{-1,0}) \\ \theta_2 &= \arg(c_{0,-1}).\end{aligned}$$

3.6.1 Procedure to estimate two frequencies

Again, the initial Fourier coefficients need to be set. For my implementation, I chose $c_{0,0} = 1$ with all remaining coefficients being set to 0. The same pulse sequence as for the estimation of one Rabi frequency is chosen.

As in the estimation of one Rabi frequency, a number of steps K is chosen to be performed and for each step $j \in [1, K]$ there is an associated $k_j = 2^{j-1}$. The overall protocol is described in algorithm 3.

3.6.2 Implementation details

As for the estimation of one Rabi frequency, if there is a number of k Fourier coefficients for each axis, there are issues while updating the Fourier coefficients since $c_{k+1,m}$, $c_{-k-1,m}$, $c_{n,k+1}$ and $c_{n,-k-1}$ are not defined. Again, I decided to set them to 0. Additionally, it is not possible to have an equivalent of $c_{-n} = c_n^*$. This implies that twice the amount of Fourier coefficients need to be saved compared to the protocol for one Rabi frequency.

In the current setup it is not possible to test this protocol since the qubits in our laboratory are not distinguishable. It might be possible to derive a protocol for indistinguishable qubits, however, we decided not to do this due to the fact that we are soon going to have the ability to distinguish the ions.

Now, consider the complexity of the algorithm estimating two Rabi frequencies simultaneously. The only difference between the two algorithms is that since a 2-dimensional Fourier series is considered, the number of coefficients will be n^2 . This leads to a complexity of $\mathcal{O}(Kmn^2)$.

Algorithm 3 Adaptive Robust Phase Estimation, two Rabi frequencies

```

for  $j \leftarrow 1$  to  $K$  do
   $k_j \leftarrow 2^{j-1}$ 
  choose number of shots  $S_j$ 
  for  $s \leftarrow 1$  to  $S_j$  do
    perform adaptive fixed rotation phase estimation as described in
    algorithm 1
  end for
  if  $j=1$  then
     $\hat{A}_{j,1} \leftarrow \arg(c_{-1,0})$  and  $\hat{A}_{j,2} \leftarrow \arg(c_{0,-1})$ 
  else
     $\hat{A}'_{j,1} \leftarrow \arg(c_{-1,0})/k_j$ 
     $q_1 \leftarrow \left\lfloor \frac{\hat{A}'_{j,1}}{2\pi/k_j} \right\rfloor$ 
     $q_2 \leftarrow q_1 + 1$ 
    if  $q_1 2\pi/k_j + \hat{A}'_{j,1} \in (\hat{A}_{j-1,1} - \pi/k_j, \hat{A}_{j-1,1} + \pi/k_j]$  then
       $\hat{A}_{j,1} \leftarrow q_1 2\pi/k_j + \hat{A}'_{j,1}$ 
    else if  $q_2 2\pi/k_j + \hat{A}'_{j,1} \in (\hat{A}_{j-1,1} - \pi/k_j, \hat{A}_{j-1,1} + \pi/k_j]$  then
       $\hat{A}_{j,1} \leftarrow q_2 2\pi/k_j + \hat{A}'_{j,1}$ 
    else
       $\hat{A}_{j,1} \leftarrow \hat{A}'_{j,1} + \pi/k_j$ 
    end if
     $\hat{A}'_{j,2} \leftarrow \arg(c_{0,-1})/k_j$ 
     $q_3 \leftarrow \left\lfloor \frac{\hat{A}'_{j,2}}{2\pi/k_j} \right\rfloor$ 
     $q_4 \leftarrow q_3 + 1$ 
    if  $q_3 2\pi/k_j + \hat{A}'_{j,2} \in (\hat{A}_{j-1,2} - \pi/k_j, \hat{A}_{j-1,2} + \pi/k_j]$  then
       $\hat{A}_{j,2} \leftarrow q_3 2\pi/k_j + \hat{A}'_{j,2}$ 
    else if  $q_4 2\pi/k_j + \hat{A}'_{j,2} \in (\hat{A}_{j-1,2} - \pi/k_j, \hat{A}_{j-1,2} + \pi/k_j]$  then
       $\hat{A}_{j,2} \leftarrow q_4 2\pi/k_j + \hat{A}'_{j,2}$ 
    else
       $\hat{A}_{j,2} \leftarrow \hat{A}'_{j,2} + \pi/k_j$ 
    end if
  end if
   $\beta_1 \leftarrow k_{j+1} \hat{A}_{j,1} - \arg(c_{-1,0})$ 
   $\beta_2 \leftarrow k_{j+1} \hat{A}_{j,2} - \arg(c_{0,-1})$ 
   $\beta = \beta_1 - \beta_2$ 
   $c_{n,m} \leftarrow c_{n,m} e^{-inm\beta}$ 
   $c_{n,m} \leftarrow |c_{n,m}|^{a^2-1} c_{n,m}$ 
end for

```

Chapter 4

Results

This chapter presents and analyzes the results obtained from simulating the protocols discussed in chapter 3. Unfortunately, it was not possible to test them in our setup due to issues with the loading of Beryllium.

4.1 Estimation of one Rabi frequency

In this section, the protocol from subsection 3.3.2 - a one-dimensional inference for indistinguishable qubits - is considered. This is the only protocol that can be currently used in our laboratory due to the fact that detection is done via a single PMT for all ions. It is, therefore, investigated in depth. The algorithm has multiple parameters which can be controlled. The two actual Rabi frequencies Ω_1 and Ω_2 can be changed to explore how the protocol behaves if the two Rabi frequencies are close together or further apart. Additionally, it is possible to change the input to the protocol Ω_{est} , the estimated value of one of the Rabi frequencies. Also, the number of steps performed as well as the number of measurements per step can be changed. All of these parameters might influence the accuracy of the estimation. Therefore, simulations were performed to explore the effects of all these parameters.

Another idea which will be explored here is the possibility of an iterative protocol. While there is already iteration in the one-dimensional protocol in that the Fourier series gets adjusted multiple times in steps, the iterative protocol provides a higher level iteration, repeating the algorithm for different inputs that are based on the previous estimation result. The algorithm provides us with an estimate $\hat{\Omega}_1$. This estimate can now be used as an input to a new protocol which will then estimate Ω_2 . This process can be repeated, which will increase the accuracy for cases where the accuracy was low previously, i.e. when either the initial input for Ω_i was an unfavorable estimate.

A number of additional parameters were kept constant for all simulations in this section. These are the number of Fourier coefficients $n = 100$, the factor which increases the standard deviation of the distribution after each step $a = 2$, and the time for which the laser pulse was applied $t_{pulse} = \frac{\Omega_1}{\pi}$. For each simulation, the protocol was run 100 times.

4.1.1 Number of measurements

We would like to find out how many steps are optimal for the protocol, as well as how many measurement shots should be performed per step. A small number of steps will lead to a less accurate protocol, and so will a lower number of measurement shots. However, a too large number of steps or measurement shots will unnecessarily increase the run time of the algorithm. Therefore, it is important to benchmark how the algorithm performs for various numbers of steps and measurement shots.

We choose $\Omega_1 = \Omega_2 = 0.5$ MHz and the input $\Omega_{est} = 0.5$ MHz. As we will see in subsection 4.1.2, the protocol performs best for these values, justifying our choice.

The first set of simulations was performed by keeping the number of shots fixed at 50 measurement shots per step. The number of steps were the integers in the range $[5, 11]$.

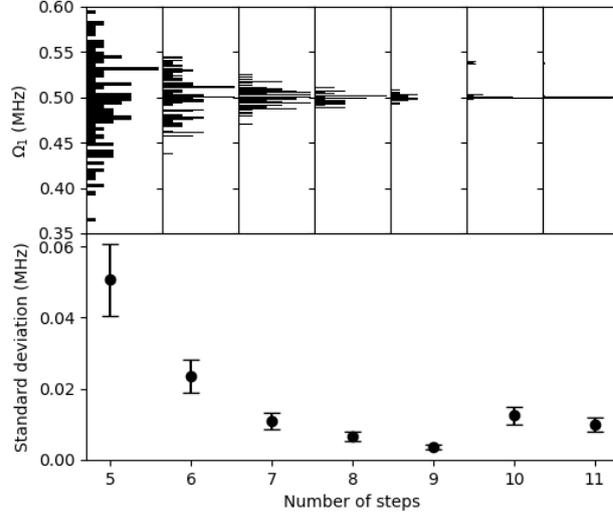


Figure 4.1: Rabi frequency estimation for a varying amount of steps. The upper plot shows histograms of the estimations for each step, while the lower plot shows the standard deviation for each of the samples.

As we can see, the accuracy of the protocol increases as more steps are performed. Thus, we should choose the maximum number of steps possible while still maintaining an acceptable run time. There is a slight increase in standard deviation for 10 as well as 11 steps. While the histogram is centered on 0.5 MHz, there are some outliers at around 0.55 MHz. These affect the standard deviation compared to for example 9 steps, where no such outliers exist. The protocols do not provide any protection against the estimation of very large or small outliers. This was also observed in later simulations. It was decided that when $|\hat{\Omega} - \Omega_{est}| \geq 0.2$ MHz, the simulated results would be discarded. This decision was based on the fact that in actual experiments results with such large differences to the input would be discarded as well and these outliers severely skew the standard deviation, as can already be observed in figure 4.1 for relatively small deviations.

The second set of simulations was performed by keeping the number of steps fixed at 8 steps, based on the previous results. We then performed a number of simulations using between 20 and 90 shots per step. The results are shown in figure 4.2

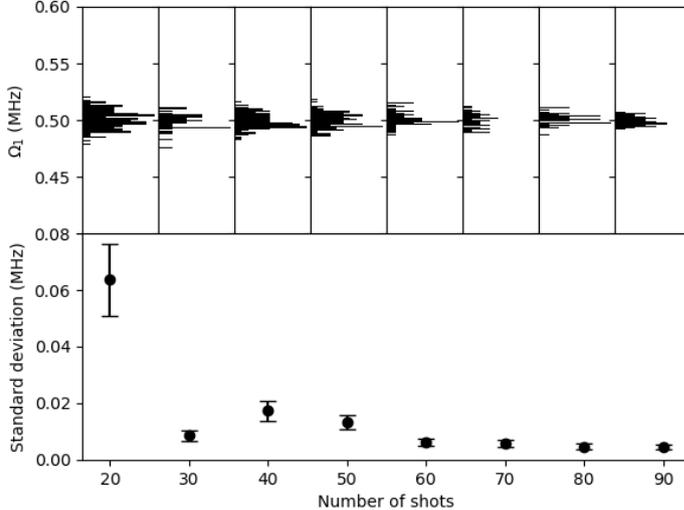


Figure 4.2: Rabi frequency estimation for a varying amount of measurement shots, keeping the number of phase approximation steps at 8 steps. The upper plot shows histograms of the estimations for each number of measurement shots, while the lower plot shows the standard deviation of each simulation.

We observe that although there is a dependency on the number of shots performed per step, the effect seems less dominant than the effect of increasing the number of steps. It should be possible to obtain good results with a low number of shots and a high number of steps. To determine the optimal number of steps and shots for our setup, this analysis should be repeated with real data. For the remainder of this section, 8 steps at 90 shots each were performed for each run of the protocol.

4.1.2 Effect of input parameter

Since the calculations within the protocol depend to a large part on the input parameter Ω_{est} , the effect of this input should be investigated. Since the protocol does not distinguish between Ω_1 and Ω_2 , one of the two Rabi frequencies can be fixed. In this subsection, we chose $\Omega_1 = 0.5$ MHz. The second Rabi frequency was chosen in the range $\Omega_2 \in \{0.5, 0.52, 0.54, 0.56, 0.58\}$ MHz. For each pair of Rabi frequencies, a number of simulations were performed, differing in the input parameter. The results are shown in figures 4.3 to 4.7

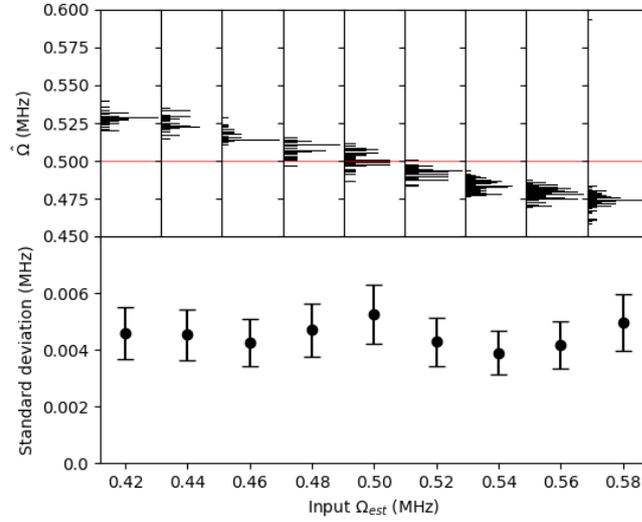


Figure 4.3: Rabi frequency estimation for a varying input Ω_{est} for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.5$ MHz. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input. The horizontal red line indicates the desired estimation result of 0.5 MHz.

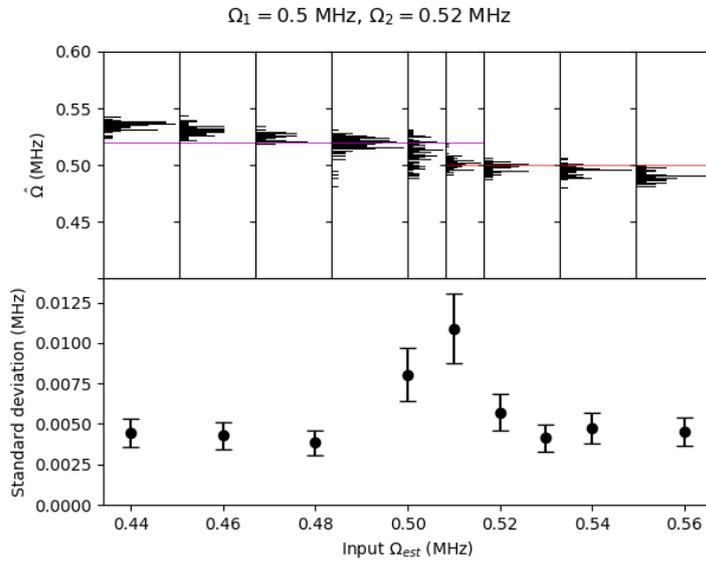


Figure 4.4: Rabi frequency estimation for a varying input Ω_{est} for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.52$ MHz. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input. The purple and red horizontal lines indicate the desired estimation results of 0.52 and 0.5 MHz respectively.

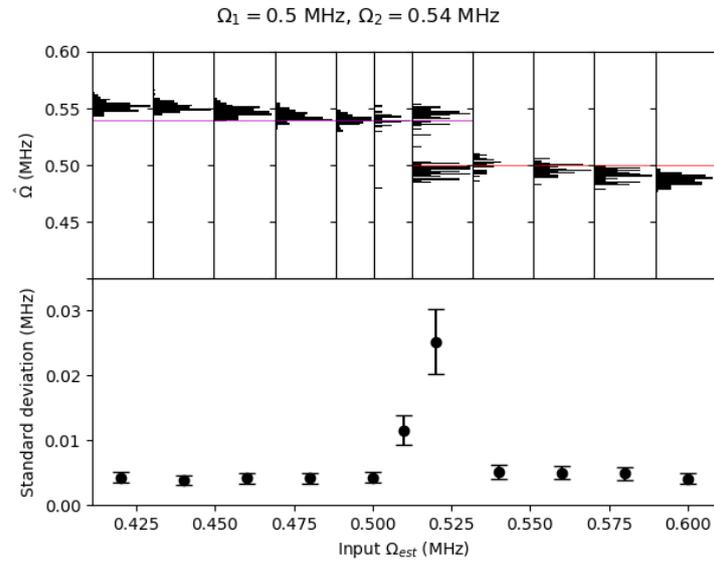


Figure 4.5: Rabi frequency estimation for a varying input Ω_{est} for $\Omega_1 = 0.5 \text{ MHz}$ and $\Omega_2 = 0.54 \text{ MHz}$. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input. The purple and red horizontal lines indicate the desired estimation results of 0.54 and 0.5 MHz respectively.

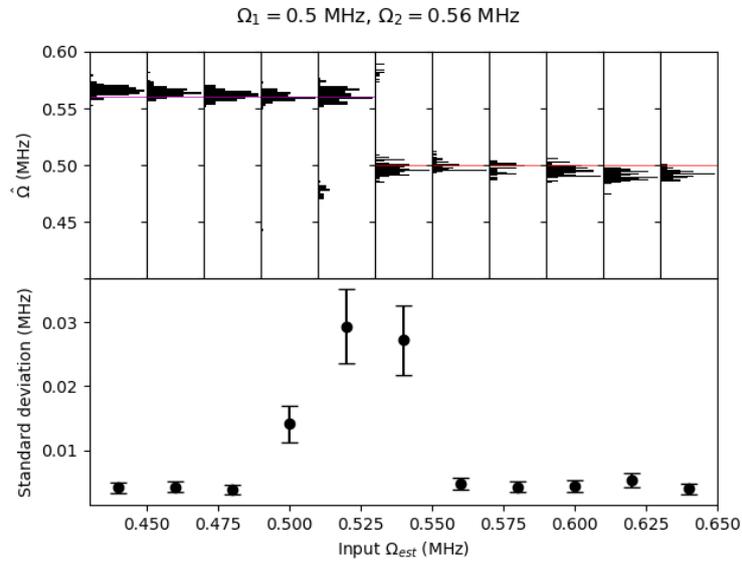


Figure 4.6: Rabi frequency estimation for a varying input Ω_{est} for $\Omega_1 = 0.5 \text{ MHz}$ and $\Omega_2 = 0.56 \text{ MHz}$. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input. The purple and red horizontal lines indicate the desired estimation results of 0.56 and 0.5 MHz respectively.

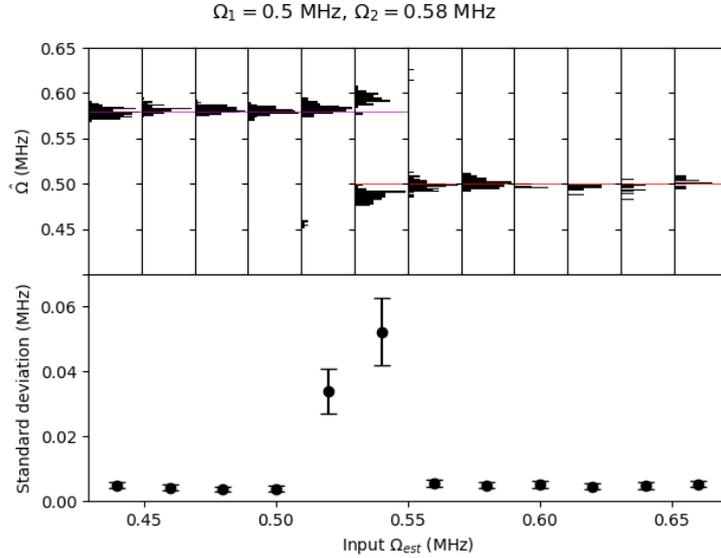


Figure 4.7: Rabi frequency estimation for a varying input Ω_{est} for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.58$ MHz. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input. The purple and red horizontal lines indicate the desired estimation results of 0.58 and 0.5 MHz respectively.

Figure 4.3 shows the case $\Omega_1 = \Omega_2$. We see that for $\Omega_{est} = 0.5$ MHz, the results are centered around 0.5 MHz as expected. We can also observe that if the input deviates from 0.5 MHz, the protocol stays accurate, however, the estimation does not converge to the correct result. Instead, it seems to be biased towards the input.

To calculate the accuracy of the estimations, we define an estimation as successful if $|\Omega_i - \hat{\Omega}_i| \leq \frac{0.2}{100} \Omega_i$. We chose a threshold of 0.2% since this is the typical accuracy to which we wish to calibrate the Rabi frequencies in the laboratory. Counting the number of correct estimations for each input Ω_{est} , we obtain the second column of table 4.1. This data also leads to the conclusion that while the estimation works well if $\Omega_{est} = 0.5$ MHz, for deviating inputs there are quickly no estimates within an acceptable range of the actual Rabi frequency.

Table 4.1: Number of correct estimations (out of 100) depending on input value. A plot of this table can be found in figure 4.8

Ω_{est} (MHz)	$\Omega_2 = 0.50$ MHz	$\Omega_2 = 0.52$ MHz	$\Omega_2 = 0.54$ MHz	$\Omega_2 = 0.56$ MHz	$\Omega_2 = 0.58$ MHz
0.42	0	-	34	-	-
0.44	0	13	48	71	96
0.46	2	49	80	66	90
0.48	54	91	94	75	89
0.50	96	7	96	74	81
0.51	-	47	1	-	-
0.52	62	90	39	0	72
0.53	-	97	-	-	-
0.54	4	87	97	84	17
0.56	0	41	90	97	91
0.58	0	-	67	93	95
0.60	-	-	28	84	91
0.62	-	-	-	49	92
0.64	-	-	-	58	87
0.66	-	-	-	-	99

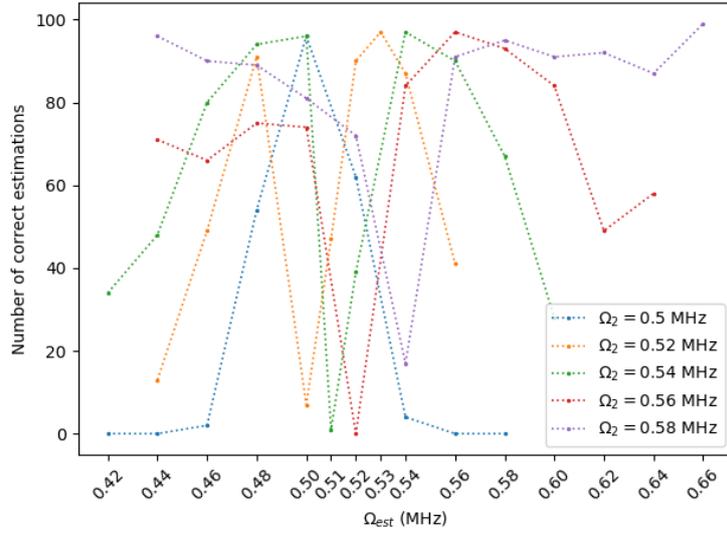


Figure 4.8: A plot containing the values of table 4.1

Figures 4.4 and 4.5 show the behavior of the algorithm if the Rabi frequencies Ω_1 and Ω_2 are within at most 0.04 MHz of each other. For both data sets, we see that for Ω_{est} far from Ω_1 or Ω_2 , the calibration approximates the value of one of the two Rabi frequencies, however, the peaks are shifted away from the correct values. This suggests that although the algorithm converges to some value, it does not converge to the correct result, at least not to an accuracy which would be desired in experiments. Calculating the

number of correct estimations using the same criterion as before leads to the results given in columns three and four of table 4.1, we see this fact clearly.

For $\Omega_{est} \in (\Omega_1, \Omega_2)$, the protocol seems to not be able to converge to a single value, which is more evident in figure 4.5. Instead, the algorithm cannot decide between estimating Ω_1 and Ω_2 , leading to an overall worse accuracy than outside this regime. The same becomes clear from the standard deviations. The standard deviation becomes large in (Ω_1, Ω_2) , pointing towards the fact that the algorithm has worse convergence in this regime. For all other inputs Ω_{est} , the standard deviation remains low, however, only for inputs close to Ω_1 or Ω_2 are most estimated values close enough to the actual Rabi frequency for experimental purposes as evident from table 4.1. These results suggest that for $\Omega_1 \approx \Omega_2$, the calibration is very sensitive to the input Ω_{est} .

Figures 4.6 and 4.7 show the results for large differences between Ω_1 and Ω_2 . These plots show that when $\Omega_{est} \ll \Omega_1$ or $\Omega_{est} \gg \Omega_2$, the algorithm manages to converge to the correct value with good accuracy. However, if $\Omega_{est} \approx \Omega_1 + \frac{\Omega_2 - \Omega_1}{2}$, there is a jump in the standard deviation, and neither of the two Rabi frequencies are estimated accurately. This seems to be the same behavior as observed in the previous plots. If the input is chosen to be near $\Omega_{est} \approx \Omega_1 + \frac{\Omega_2 - \Omega_1}{2}$, the algorithm does not converge well to either of the two Rabi frequencies and instead oscillates between the two values. These findings are supported by the number of correct estimations in columns five and six of table 4.1.

In conclusion, we can say that this protocol is very sensitive to the input Ω_{est} as well as the difference between the two Rabi frequencies. For small differences, a desired accuracy seems only achievable if $\Omega_{est} \approx \Omega_i$. For large differences, the protocol converges well as long as the input is not chosen such that $\Omega_{est} \approx \Omega_1 + \frac{\Omega_2 - \Omega_1}{2}$.

4.1.3 Iterative protocol

As shown in subsection 4.1.2, whether the estimation succeeds is dependent on the input Ω_{est} . Choosing an estimate in the range (Ω_1, Ω_2) seems undesirable. Choosing inputs far away from the actual Rabi frequencies seems to also be not advisable, at least if $\Omega_1 \approx \Omega_2$. In this subsection, we present the results when the output from the protocol is used as input to another protocol, regardless of whether this result was correct. This process was repeated a total of four times.

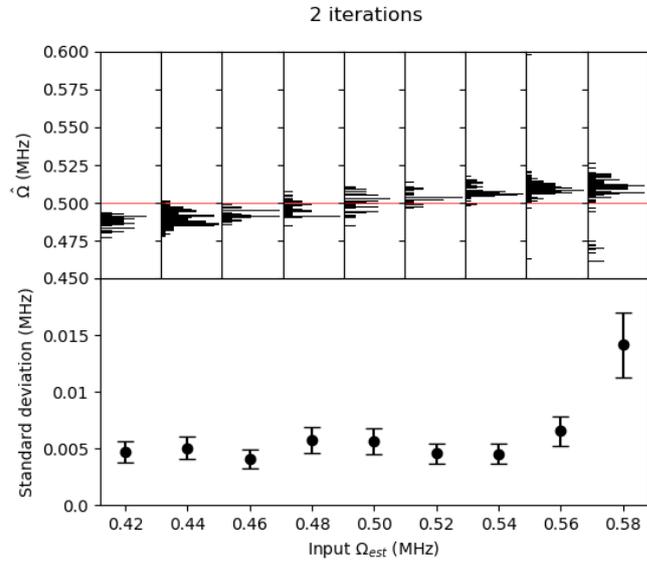


Figure 4.9: Rabi frequency estimation for an iterative protocol with a varying first input Ω_{est} , for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.5$ MHz. 2 iterations were performed. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input.

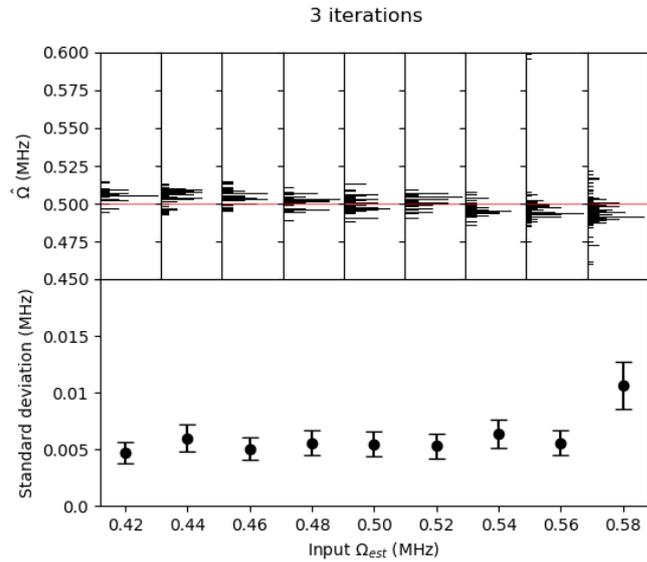


Figure 4.10: Rabi frequency estimation for an iterative protocol with a varying first input Ω_{est} , for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.5$ MHz. 3 iterations were performed. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input.

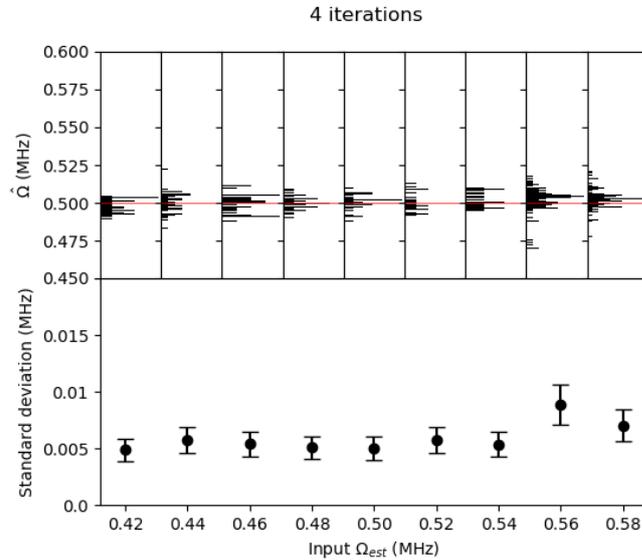


Figure 4.11: Rabi frequency estimation for an iterative protocol with a varying first input Ω_{est} , for $\Omega_1 = 0.5$ MHz and $\Omega_2 = 0.5$ MHz. 4 iterations were performed. The upper plot shows histograms of the estimations for each input, while the lower plot shows the standard deviation as a function of this input.

For few steps, such as in figure 4.9, the protocol still does not estimate the correct value of 0.5 MHz, when the input Ω_{est} is not close to 0.5 MHz. However, already for two steps the dependency on the input seems to be a bit relaxed, the peaks of each histogram are much closer to 0.5 MHz than in figure 4.3. We can also observe that when performing more steps such as in figure 4.11, the dependency vanishes almost completely. Similar behavior can be found when $\Omega_1 \neq \Omega_2$ as shown in table 4.2. While a simple protocol according to section 3.3.2 does not perform well enough to be used for calibration, it appears that iterations can be used for this purpose. However, this behavior has to be confirmed in experiments. If these results are found in real data as well, the optimal number of iteration steps should be determined.

Table 4.2: Number of correct estimations (out of 100) depending on input value after 4 iterations, for each starting value. A plot of this table can be found in figure 4.12

Ω_{est} (MHz)	$\Omega_2 = 0.50$ (MHz)	$\Omega_2 = 0.52$ (MHz)	$\Omega_2 = 0.54$ (MHz)	$\Omega_2 = 0.56$ (MHz)	$\Omega_2 = 0.58$ (MHz)
0.42	92	-	62	-	-
0.44	94	74	62	67	89
0.46	94	84	46	68	69
0.48	98	75	58	70	77
0.50	95	88	68	65	77
0.51	-	82	76	-	-
0.52	93	83	77	67	64
0.53	-	88	-	-	-
0.54	95	87	82	66	85
0.56	79	90	71	71	91
0.58	78	-	88	71	92
0.60	-	-	71	74	86
0.62	-	-	-	74	88
0.64	-	-	-	78	92
0.66	-	-	-	-	87

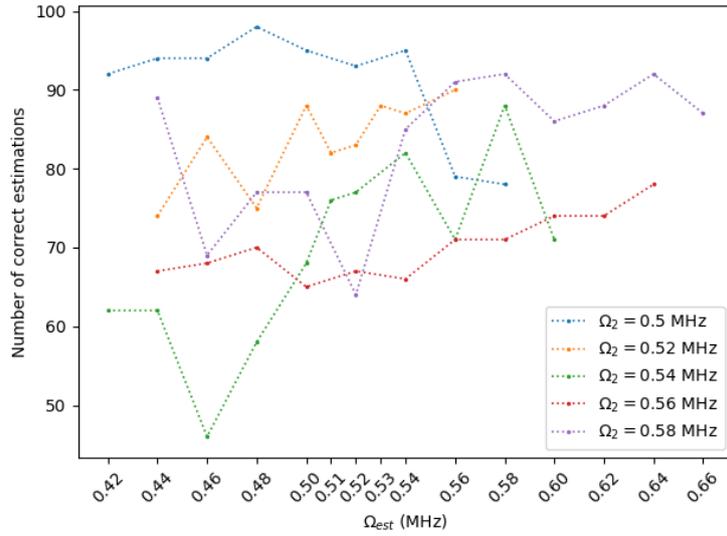


Figure 4.12: A plot of the values given in table 4.2

4.2 Estimation of two frequencies

It is not possible in our current setup to perform the estimation of both Rabi frequencies simultaneously with the protocol derived in section 3.6. However, the plan of our group is to soon implement a way to distinguish the two ions. Therefore, it is useful to compare the protocol which estimates

two Rabi frequencies, presented in section 3.6 to the one which can currently be implemented ¹.

The accuracy of this protocol does not change depending on the value of one single Rabi frequency, it only depends on the difference between Ω_1 and Ω_2 . It is, therefore, possible to fix one of the two values. For this simulation, the first Rabi frequency was fixed to $\Omega_1 = 0.5$ MHz. Since the protocol is symmetric, it does not matter whether Ω_1 or Ω_2 is fixed. The second Rabi frequency Ω_2 was chosen to be in the set $\{0.5, 0.52, 0.54, 0.56, 0.58, 0.60\}$ MHz. The number of Fourier coefficients, the factor to increase the standard deviation after a step and the time of the laser pulse were chosen the same way as in section 4.1. The number of steps was set to 9. Unlike in the previous simulations, we chose to perform 200 measurement shots per step. This was due to the fact that if we chose a value closer to the one used for the simulations for the estimation of one frequency, for example 90 shots per step, the algorithm had not yet converged. For each pair of Rabi frequencies, 100 estimations were performed.

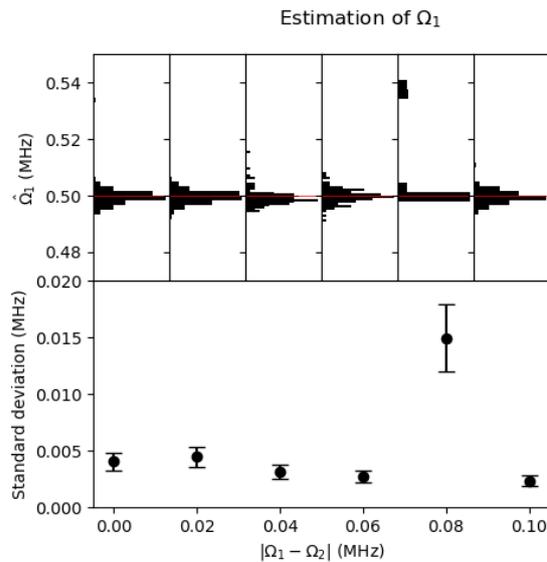


Figure 4.13: Rabi frequency estimation for a two-dimensional protocol. The upper plot shows histograms of the estimations of Ω_1 for each difference $|\Omega_1 - \Omega_2|$, while the lower plot shows the standard deviation as a function of this difference. The horizontal red line indicates the desired result of the estimation of 0.5 MHz.

¹After performing these simulations, some minor bugs in the implementation of the protocol were discovered. The results presented here might therefore slightly differ from results obtained from the corrected version of the protocol

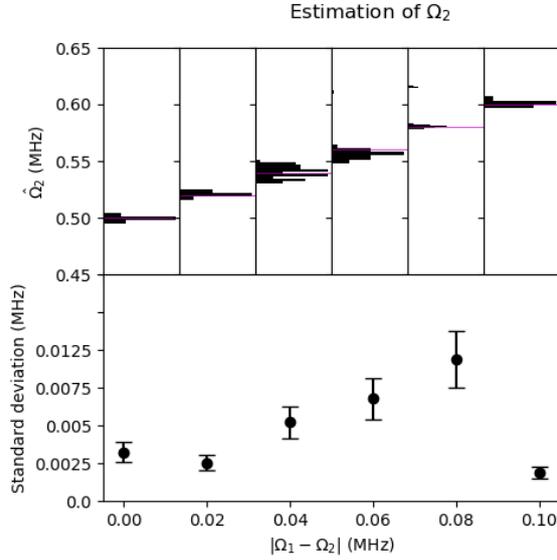


Figure 4.14: Rabi frequency estimation for a two-dimensional protocol. The upper plot shows histograms of the estimations of Ω_2 for each difference $|\Omega_1 - \Omega_2|$, while the lower plot shows the standard deviation as a function of this difference. The horizontal purple line indicates the desired result of the estimation which is equal to Ω_2 .

Figure 4.13 shows the estimations of Ω_1 . As we can observe, the histograms are always more or less centered on 0.5 MHz which corresponds to Ω_1 . As the histogram for $\Omega_2 = 0.58$ MHz suggests, there is still the possibility for wrong estimations, however, it seems to be much less probable than for a one-dimensional protocol. Figure 4.14 supports this conclusion. Table 4.3 shows how many estimations were accepted as correct. As in section 4.1, the estimated Rabi frequencies were compared to the actual values and the result was assumed to be correct if $|\Omega_1 - \hat{\Omega}_1| \leq 0.02\Omega_1$ and $|\Omega_2 - \hat{\Omega}_2| \leq 0.02\Omega_2$. Note that the threshold for acceptance was here chosen to be $0.02\Omega_i$ instead of $0.002\Omega_i$. This was due to the fact that for a smaller threshold, almost no results were accepted anymore. An acceptable threshold for experimental purposes needs to be determined before this estimation can be used for calibration.

Ω_2 (MHz)	Number of correct estimations (out of 100)
0.50	93
0.52	92
0.54	87
0.56	84
0.58	67
0.60	51

Table 4.3: Number of correct estimations for a two-dimensional protocol

Chapter 5

Outlook

This thesis derived multiple protocols which estimate one or two Rabi frequencies in a two qubit setup. Though the results need to be confirmed experimentally, it seems that an iterative one-dimensional protocol can estimate one of two Rabi frequencies accurately, independent of the difference between the two Rabi frequencies as well as the input value. A one-dimensional protocol without iteration appears to be too dependent on the input and difference to be used for calibration. A two-dimensional inference scheme is dependent on the difference between the two Rabi frequencies, giving it a lower accuracy than a one-dimensional equivalent for large differences. Although it could be useful to consider a mixture of the one- and two-dimensional protocol, an iterative one-dimensional protocol seems to be sufficient for calibration.

However, although these protocols allow for the estimation of Rabi frequencies, this does not provide a full calibration. Only when both Rabi frequencies are equal, applying a laser pulse for some fixed time will rotate both of the ion states by the same amount. For the calibration of a quantum computer it is, therefore, necessary to change the two Rabi frequencies after the estimation.

Luckily, it is straightforward to change the values of the Rabi frequencies, since they only depend on the dipole moment of the transition, which does not change for the time scales considered, the polarisation of the beams and the alignment between laser and the ions. Moving the laser will, thus, change the Rabi frequencies. The easiest way to perform laser alignment is to run one of the protocols continuously and to move the laser manually after an estimation is completed.

It would of course be desirable to automate this calibration completely. Assuming a Gaussian profile for the laser beam, it can easily be calculated to which position the laser should be moved such that both Rabi frequencies are equal. If a protocol written for indistinguishable qubits is implemented,

it is probably desirable to iterate for multiple steps, alternating between estimation and moving the laser, to obtain an accurate result.

In our setup, the movement of the laser is controlled by a piezoelectric ceramic. Piezoelectric materials show hysteresis effects. To move the laser, a voltage is specified which is applied to the piezo and depending on the voltage, the piezo contracts or expands and moves the laser. Unfortunately, due to hysteresis, there is no linear relation between the laser position and this voltage. To move the laser to a certain position, it is necessary to measure the hysteresis curve of the material. This curve can then be fitted with an invertible model, giving a relation between voltage and position. Then it should be possible to perform an automated alignment.

Each of the protocols presented in chapter 3 is capable of calibrating the two Rabi frequencies of an ion trap given laser movement. Since one and two qubit gates are sufficient in order to implement a universal quantum computer, it should not be necessary to extend the estimation to more qubits. Extending the protocols to more than two qubits will quickly become impractical when choosing a direct approach via Bayes' theorem. Not only are such protocols difficult to derive, the computational complexity of such algorithms will also limit their usefulness. Depending on the exact problem, it might be more practical to consider Monte Carlo based inference algorithms or possibly likelihood-free inference methods such as approximate Bayesian computation. Overall, the protocols presented in this thesis are an important stepping stone to the automation of the calibration of ion trap quantum computers. They also provide a more efficient way to calibrate compared to conventional methods. However, to fully achieve automation, further considerations are still necessary.

Appendix

The code for the simulations I performed can be found on the physics department Gitlab in the TIQI Projects folder. The files `fixed_frequency_arpe.h` and `fixed_frequency_arpe.cpp` implement the phase estimation. They are thoroughly documented. The file `main.cpp` contains a function `perform_arpe`, which performs one phase estimation, given some input parameters. This function is documented as well. For convenience, an example is provided in the `main()` method. Please note that to obtain deterministic results, the random number generator has to be seeded using `srand(double seed)`. A similar setup can be found for the estimation of two Rabi frequencies. I also implemented an experiment which should run the one-dimensional protocol for indistinguishable qubits. This can be found on Gitlab in `exp_arpe_2Be1Ca.cpp`. There were important alterations to the `frame_CoolDet_2Be1Ca.cpp`, which can be found in `frame_arpe_2Be1Ca.cpp`.

Namely, a variable `sequence_experiment_number` is saved on the frame. In the `read_out_pmts()` method, this integer is reduced by one every time the PMTs get read out. This implements the steps of the protocol. The integer is initialized for each step as the number of shots that the user specified in the experiment. When the counter reaches 0, the frame checks with the experiment whether a new step should be initialized and if so, it calls the `change_sequence` method which starts a new cooling, state preparation and qubit manipulation. This ensures that the steps are run properly.

In `exp_arpe_2Be1Ca.cpp`, the experiment is implemented. There are a number of parameters which can be specified by the user. They are documented in the header file. One can set whether the experiment should be run again if the estimation fails. Additionally, one can set the number of shots per step. The step number is currently limited to 8 steps, however, this can easily be increased. The user can also provide a known π - and known $\frac{\pi}{2}$ -time, which corresponds to the Rabi frequency given as input, or θ_{known} . The number of Fourier coefficients can also be adjusted. One can, in addition, provide a threshold which is used to decide whether the calibration was successful or not. The estimated value will be compared to the provided Rabi frequency and accepted if the difference between the two is within this threshold. Finally, one can provide the protocol with an integer specifying how many iterations of the protocol should be run. If this is set to 0, no iteration is performed. However, this last feature was not tested and might contain bugs. The experiment first calls the `Init()` method, which initializes all variables which are needed for the phase approximation. The `QubitManipulation()` method is overloaded to perform the three laser pulses described in this thesis. The `GetDefaultPmtCounts()` method is overloaded so that after each measurement in the experiment, the Fourier coefficients are updated and the FPAs are provided with the new phases, calculated from the mea-

surement angle α . The `get_sequence_experiment_number()` updates the `sequence_experiment_number` of the frame based on the number of shots specified by the user for the current step. Whether or not the experiment should be continued after a phase estimation is decided in the `PostSequenceCalculations()` method. If the phase estimation is not yet completed, i.e. there are more steps to be performed, the experiment is continued. Otherwise, the `stopExperiment()` method is called, which calculates the estimated π -time and Rabi frequency. If the Rabi frequency is not accepted according to the threshold set by the user, this method checks whether the user desires to rerun the experiment. Depending on this, either the variable `_exp_loop` is set to a value higher than allowed, making the experiment terminate, or the method communicates that the experiment should be run again.

If the experiment is not terminated after this method, `postCalibrationandReset()` is called. This method first estimates the Rabi frequency. Depending on the estimated value, the global variable corresponding to the π -time is updated. After this method has finished running, the `postCalibrationandReset()` method will either re-initialize the experiment if multiple experiments should be run or the experiment is terminated using the `endProtocol()` method.

Bibliography

- [1] O. Benson. Quantenoptik - chapter 13. <https://www.physik.hu-berlin.de/de/nano/lehre/quantenoptik09/chapter13>. Accessed: 2.5.2018.
- [2] E. Bertolazzi. One-Dimensional Minimization. <http://www.ing.unitn.it/~bertolaz/2-teaching/2011-2012/AA-2011-2012-OPTIM/lezioni/slides-m1D.pdf>. Accessed: 26.4.2018.
- [3] N. Bohr. On the Constitution of Atoms and Molecules. *Philosophical Magazine*, 1913.
- [4] D. DiVincenzo. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*, 48:771–783, 2000.
- [5] A. Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Annalen der Physik*, 17:132–148, 1905.
- [6] R. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.
- [7] H. Häffner, C. Roos, and R. Blatt. Quantum computing with trapped ions. *Physics Reports*, 469:155–203, 2008.
- [8] E. T. Jaynes. Information Theory and Statistical Mechanics. *Statistical Physics*, 1963.
- [9] J. Kelly. A Preview of Bristlecone, Google’s New Quantum Processor. <https://research.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>. Accessed: 26.4.2018.
- [10] A. Lebedev. Meteorology Report. Private communication.

- [11] B. MacDonald-de Neeve. Calibrating an Ion Trap Quantum Computer. Thesis written in the trapped ion quantum information group in 2017.
- [12] Y. Manin. Vychislimoe i nevychislimoe (computable and noncomputable). *Sov.Radio*, pages 13–15, 1980.
- [13] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information (2nd ed.)*. Cambridge University Press, 2010.
- [14] J. Pita. Design, Development, and Operation of Novel Ion Trap Geometries. <http://www.imperial.ac.uk/media/imperial-college/research-centres-and-groups/ion-trapping/public/Design-Development-and-Operation-of-Novel-Ion-Trap-Geometries.pdf>. Accessed: 2.5.2018.
- [15] M. Planck. Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum. *Verhandlungen der Deutschen Physikalischen Gesellschaft*, 2:237–245, 1900.
- [16] G. H. Low S. Kimmel and T. J. Yoder. Robust Calibration of a Universal Single-Qubit Gate set via Robust Phase Estimation. *Physical Review A*, 92, 2015.
- [17] P. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.