Quantum Control and Computing with Neutral ^{88}Sr Atoms

Robin Eberhard

München 2025

Quantum Control and Computing with Neutral ^{88}Sr Atoms

Robin Eberhard

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Robin Eberhard
aus Aalen

München, 28.10.2025

Erstgutachter: Prof. Dr. Immanuel Bloch
Zweitgutachter: Prof. Dr. Christian Groß
Tag der mündlichen Prüfung: 9.12.2025

"In other studies you go as far as others have gone before you, and there is nothing more to know; but in a scientific pursuit there is continual food for discovery and wonder."

Victor Frankenstein

Mary Shelley

Zusammenfassung

In Quantencomputern wird die Überlagerung von Quantenzuständen genutzt, um Berechnungen durchzuführen, die theoretisch unmöglich oder nicht durchführbar sind, wie zum Beispiel das travelling salesman Problem für eine große Anzahl an Knotenpunkten oder unstrukturierte Suchen für große Datenbanken. Aufgrund ihrer Skalierbarkeit und Flexibilität sind neutrale Atome gefangen in optischen Pinzetten, eine vielversprechende Technologie, um Quantenberechnungen durchzuführen. Hier sind die Qubit-Konfigurationen nicht statisch, sodass beliebige Geometrien realisiert werden können und Algorithmen, in denen Qubits ausgetauscht werden, leicht verfügbar sind.

Für neutrale Atome wird die Quantenstruktur genutzt, um Qubits in zwei Niveaus zu kodieren, um die Basis für Quantenberechnungen zu schaffen. Wenn Atome in ihre elektronischen und Bewegungs-Grundzustände gekühlt werden, kann der Übergang zwischen qubit null und qubit eins mit hoher Genauigkeit getrieben werden. Ein Multi-Qubit-Gatter kann dann realisiert werden, wenn zwei oder mehr Atome miteinander interagieren, z.B. indem sie nahe zusammengebracht werden oder indem ihr Dipolradius vergrößert wird, wenn sie in einen Rydberg-Zustand angeregt werden.

Bisher wurden verschiedene Qubits mit neutralen Atomen realisiert, insbesondere in ^{87}Rb, wo der elektronische Grundzustand in zwei Unterzustände aufgeteilt wird, welche als Kodierung für das Qubit genutzt werden. Wir beschreiben die Fortschritte, um einen Quantencomputer basierend auf ^{88}Sr zu realisieren. Als Boson hat der elektronische Grundzustand keine Hyperfeinstruktur, sodass state preparation and measurement (SPAM) Fehler eliminiert werden können. Das Qubit ist im Grundzustand zusammen mit dem viel studierten, langlebigen Uhrenzustand kodiert, was das Treiben des Übergangs mit hoher Genauigkeit ermöglicht.

In dieser Arbeit wird ein Apparat präsentiert, der ein ultrahohes Vakuum erzeugt, da langlebige Atome notwendig sind für leistungsstarke Quantenalgorithmen. Wir zeigen die wichtigsten Ideen hinter den Lasersystemen, die zum Kühlen, Fangen und Treiben der Einzelqubit-Operation genutzt werden. Unsere Arbeit demonstriert, dass kohärente Bewegung von Atomen eine realisierbare Operation von Quantencomputern ist, welche erlaubt Qubits auszutauschen, sie in separate Register zu verschieben oder zusätzliche Phasengatter zu ermöglichen.

Die hier präsentierten Ergebnisse etablieren ^{88}Sr als einen geeigneten Kandidaten für Quantencomputer und legen den Grundstein für zukünftige fehlertolerante Quantensysteme.

Abstract

In quantum computers, the superposition of quantum states is used to perform calculations that are theoretically impossible or infeasible with classical computers, such as solving the traveling salesman problem for many nodes or conducting unstructured searches for large databases. Due to its scalable and flexible nature, neutral atoms trapped in optical tweezers are a promising technology for performing quantum computation. Here, qubit configurations are not static, allowing arbitrary geometries to be realized, and schemes where qubits are exchanged are readily available.

The quantum level structure of neutral atoms is used to encode a qubit between two states, forming the basis for quantum computing. As the atoms are cooled to their electronic and motional ground states, transitions between qubit zero and qubit one can be driven with high fidelity. A multi-qubit gate is then realized when two or more atoms interact, e.g., by moving them close together, or by increasing their dipole radius when they are excited to a Rydberg state.

Up to now, different qubits have been realized in neutral atoms, most notably ^{87}Rb, where the electronic ground state splits into two sublevels, which are used as encoding of a qubit. We describe progress on realizing a quantum computer based on ^{88}Sr. As a boson, its electronic ground state exhibits no hyperfine splitting, such that SPAM errors can be eliminated. The qubit is encoded in the ground state together with the highly studied, long-lived clock state, allowing for high-fidelity control of the transition.

In this thesis, we present how the apparatus creates an ultra-high vacuum, as long-lived atoms are necessary for high-performance quantum algorithms. We show the main ideas behind the laser systems used for cooling, trapping, and driving the single-qubit transition. Our work demonstrates that coherently moving atoms is a feasible operation in quantum computing, allowing for the exchange of qubits, the shuttling of qubits to separate registers, or enabling additional phase gates.

The results presented here establish ^{88}Sr as a viable candidate for quantum computing and lay the foundation for future fault-tolerant quantum systems.

Contents

1. Introduction

The discovery of quantum mechanics was a scientific breakthrough in the early 20th century, revealing that energy is distributed in a quantized way in atomic systems [1,2]. Richard Feynman discovered in 1982 that it is possible to simulate quantum systems on a new platform: Quantum computers [3–5]. The foundation of quantum computing is qubits, which are a superposition of two quantum states [6,7]. There are different ways of creating qubits, e.g., by encoding the information in the polarization of photons [8], in superconductors exploiting the Josephson effect [9], in the vibrational state of an ion chain [10], or in the electronic state of atoms [11,12]. The latter is the foundation for neutral atom quantum computing, where atoms are trapped and cooled individually, and their inherent quantum structure is used for quantum operations. DiVincenzo has derived five fundamental criteria [13] in order to build a quantum computer:

1. A quantum system containing qubits
2. The system needs to be initializable into a defined state
3. The quantum system needs to be isolated from its environment
4. Single-qubit and multi-qubit operations need to be available
5. A way to measure the result of the operations.

For neutral atom experiments, these are naturally fulfilled [11,14–17]. Atoms are cooled and trapped in a vacuum, where they are isolated from outside influence. Their electronic ground state provides a well-defined initialized state, and driving Rabi or Raman transitions to other states allows single-qubit operations. Multi-qubit operations can be realized by interacting atoms, either by moving them close together or using Rydberg states to increase their electric dipole radius. The atoms are trapped individually, and measurements are performed by collecting fluorescent photons on a transition with a high scattering rate.

Trapping of atoms is achieved by using the AC Stark effect, where the atom, as a dipole, interacts with an electric field, such as that from a laser beam. For single atoms, this was first achieved in optical lattices [18–21], where a laser beam of wavelength λ interferes with itself (e.g., by retro-reflecting it), creating a standing wave with lattice spacing down to $\lambda/2$. Creating a filled and defect-free array, however, is another challenge. In optical lattices, one way to achieve this is starting from a Bose-Einstein condensate (BEC)

[22] and utilizing the phase transition to a Mott insulator [18]. However, to get to a BEC, atoms are usually evaporatively cooled in an optical dipole trap, which is a time-consuming process [23–25]. Alternatively, atoms can first be loaded into an array of optical tweezers [26–28], which themselves are dipole traps with a tight focus. As their trap depths are much deeper than in a lattice, the atoms can be pre-cooled before they are transferred.

Alternatively, experiments can also be performed entirely in optical tweezers, where atoms are loaded directly from a magneto-optical trap (MOT) [29,30]. By creating an array of focused beams, geometries are more variable than in the case of optical lattices. The arrays are generally created by using spatial light modulators (SLMs) [31,32], digital micromirror devices (DMDs) [33], or acousto-optical Deflectors (AODs) [26], which are electronically controlled devices for spatially controlling laser beams. SLMs and DMDs are two-dimensional devices containing pixels that can be set to imprint a phase or amplitude pattern, respectively, onto a laser beam. In this way, it is possible to form a single initial laser beam into an array of output beams. Alternatively, an AOD works using the same principle as an acousto-optical modulator (AOM) however, the crystals usually have a slower acoustic velocity, which results in a larger deflection angle. Applying an electronic sine wave with a frequency and amplitude deflects an incident light beam. The angle of the deflected beam depends on the frequency of the electronic wave, and the laser power depends on the wave amplitude. By applying a superposition of waves, one beam is deflected into multiple output beams. As the device is one-dimensional, creating a two-dimensional pattern involves placing two AODs in series.

Systems of up to 6100 qubits [34] have already been demonstrated, and further scaling is very possible at the current state [35]. However, other challenges are being tackled right now, such as high fidelity gate operations, lifetime of qubits for running long algorithms, fast execution times, error correction, and connectivity [36–43]. The neutral atom platform has excellent prospects. For example, it is possible to create dynamic qubit configurations and connectivities that can change in real-time during a quantum algorithm, allowing, for instance, SWAP gates to be directly possible. Single-qubit addressing with minimal cross-talk can be implemented using tightly focused beams that only interact with a single site. In terms of cycle times, most current implementations are limited due to the initialization of the atoms, which need to be trapped and cooled in a MOT, then transferred to tweezers or lattices and cooled there again. However, new schemes where qubits are reinitialized after a run, keeping a consistent storage of replacement atoms or having a static MOT with a moving lattice to transport new atoms are all ideas that are ready to be implemented in modern neutral atom quantum computers [44–46].

In this thesis, a new ^{88}Sr-based neutral atom quantum computing platform is presented. With a single electronic ground state and its coupling to two long-lived states, the atom is a good choice for encoding qubits [47,48]. A compact vacuum system is presented, which we show can reach exceptional atomic lifetimes. Cooling lasers are modularized in a custom laser rack, minimizing space occupation around the chamber. Optical tweezers are generated using a 2D-AOD, where the generation of RF tones is explained in detail. Measurements towards generating singly-occupied tweezers and cooling atoms to their motional ground state are provided. A novel feature of this experiment is the ability to coherently transport atoms, as measured by [49] using interleaved randomized benchmarking (IRB).

A new software stack was developed, enabling digital, gate-based quantum computing and exploring error correction. Additionally, analog quantum simulation can be explored using tweezers directly or by implementing a lattice in the future.

The thesis is structured as follows:

Chapter 2 - Hardware of a neutral atom tweezer experiment

The ^{88}Sr atom is introduced, with an examination of its level structure, the general cooling mechanisms, and its role as a driver for quantum computing. We follow up with the mechanical system, including the vacuum chamber, the coils, the glass cell, and other smaller components.

Chapter 3 - Rack-based laser systems

A laser rack was prototyped as part of this experiment, containing shelves of laser breadboards. It conserves space, and temperature stability is maintained by using a water cooling cycle. We highlight the importance of modular setups and general rules that aid in debugging later on.

Chapter 4 - Generating and optimizing optical tweezers

In this experiment, atoms are trapped in an optical tweezer generated by a 2D-AOD. The laser system is explained, including the theory on the magnification of tones when using an AOD. The RF tones are generated using an M4i6631-x8 card. Its programming is explained in detail, and tools for debugging waveforms are provided. We present a procedure for generating a defect-free 2D-AOD array, and explain how to homogenize the array.

Chapter 5 - Trapping, cooling, and measuring single atoms

This chapter provides complete insight into the initialization sequence of the experiment: Trapping and cooling atoms in a dual-stage MOT, loading single atoms into optical tweezers, and cooling them to their motional ground state. All relevant measurements are provided, and we demonstrate the performance of the vacuum system by measuring the lifetime of atoms in tweezers.

Chapter 6 - Effect of motion on qubit coherence

An important feature of this experiment and the neutral atom platform for quantum computing is its ability to transport atoms. In this chapter, we present two waveforms for moving atoms: constant jerk and adiabatic sine. For both waveforms, we evaluate the effect on the motional state occupation, and we measure the loss of coherence for any superposition of the qubit.

Chapter 7 - Software for neutral atom experiments

This experiment features a new control system based on the ARTIQ software and Sinara hardware. An additional software stack was programmed on top of ARTIQ to fit the needs of our experiment. It includes fast storage and retrieval of data into a database, configurable user channels, and separates logic from parameters in the experiment code.

2. Hardware of a neutral atom tweezer experiment

Trapping and cooling neutral atoms and ions has evolved to a major field [17,50–57], where their electronic and motional quantum level structure is used to simulate physical models, and more recently utilizing them as qubits in a quantum computing architecture [14,35,58,59]. The physical setup of these experiments can be understood in terms of modules, consisting of a vacuum chamber, magnetic coils, and laser systems tailored to the specific application [60]. The exact arrangement varies from one experiment to another. As an example, some experiments may choose to use multiple species for various purposes [61] (e.g., cooling or studying molecular transitions). Some may contain counterpropagating laser beams to create a lattice [18,62,63].

In contrast, others may choose to trap atoms in optical tweezers [62,64,65] or in a dipole trap to create a BEC [54]. The atoms have to be cooled in order to study and leverage their quantum properties, so the experiments will always work in an ultra-high vacuum with pressures ranging from 1×10^{-10} mbar to 1×10^{-12} mbar. Atoms have to be in a gaseous form, which is generally achieved by heating a solid sample consisting of the desired species. Cooling the atoms can then occur transversally with a 2D-MOT stage and longitudinally using a Zeeman slower; however, some species can be loaded into a 3D-MOT immediately [66]. These initial cooling stages require magnetic fields, which can be generated using permanent magnets or via electric coils. The atoms are directed towards a science chamber (which can be a steel/titanium chamber with viewports or a glass cell) and trapped there in a 3D-MOT, cooling them in all three dimensions. Further preparation steps are very experiment-specific. The following is a more detailed explanation of the specific setup of this thesis. We explore the quantum level structure of ^{88}Sr and then go into detail about the mechanical setup of the experiment.

2.1. Atomic level-structure of ^{88}Sr

Strontium-88 has undergone much research since it is a prominent candidate for a new clock standard. There is also a good understanding of this isotope within the group, and therefore, we decided to use Strontium-88 for our quantum computing apparatus. Moreover, it aligns very well with DiVincenzo's criteria for a quantum computer. Naturally, all neutral atom experiments with single-atom resolution are suitable to fulfill the requirements. The atoms are trapped in an isolated environment (the vacuum) and cooled to their electronic ground state. Moreover, single-qubit operations are immediately available via simple Rabi or Raman transitions. Multi-qubit operations can then be performed using collisional gates or by using the Rydberg blockade effect. Here, ^{88}Sr is especially suitable, since as a boson it only contains a single electronic ground state, thus SPAM errors can be eliminated. Moreover, encoding the qubit in $|0\rangle = 1S0$ and $|1\rangle = 3P0$ is an excellent choice, as the clock state ($^3P_0$) has a very long lifetime and thus does not decay back to the ground state within reasonable durations of quantum computing operations. A highlight of the experiment in this thesis is the ability to coherently transport atoms during quantum operations, allowing qubits to be stored in specific registers, swapped, or implemented as simple phase gates when atoms are shifted parallel to a driving beam. A more detailed explanation of quantum computing with ^{88}Sr is provided below, where the level scheme in Figure 2.1 serves as a reference.

Reaching the quantum computation step of the experiment requires cooling and trapping single atoms. For ^{88}Sr, this means trapping and cooling in a two-stage MOT setup, where the broad "blue" cooling transition $(5s)^2\ {}^1S_0 \rightarrow 5s5p\,{}^1P_1$ is used for initial cooling to a few hundred µK. In a second MOT stage, the atomic cloud is further compressed and cooled to sub-µK temperatures on the narrow "red" cooling transition $5s\,{}^1S_0 \rightarrow 5p\,{}^3P_1$. Due to its high scattering rate, the ${}^1S_0 \rightarrow {}^1P_1$ transition is also used for absorption imaging of the atomic cloud and fluorescence imaging of single atoms. The temperature limits for the MOTs are found using the doppler temperature:

$$T_D(\Gamma) = \frac{\hbar\,\Gamma}{2k_B} \tag{2.2}$$

such that $T_{D,\,\mathrm{blue}} = T_D(190\,\mathrm{MHz}) = 727\,\mu\mathrm{K}$ and $T_{D,\,\mathrm{red}} = T_D(47.00\,\mathrm{kHz}) = 179\,\mathrm{nK}$ [70].

The ^{88}Sr atoms are then trapped in optical tweezers, which are an array of optical dipole traps at the clock-magic wavelength 813 nm, where both the ground state $5s\,{}^1S_0 = |0\rangle$ and the clock state $5s\,{}^3P_0 = |1\rangle$ experience the same light shift. Single atoms are then generated by applying a parity projection pulse, where molecular transitions for

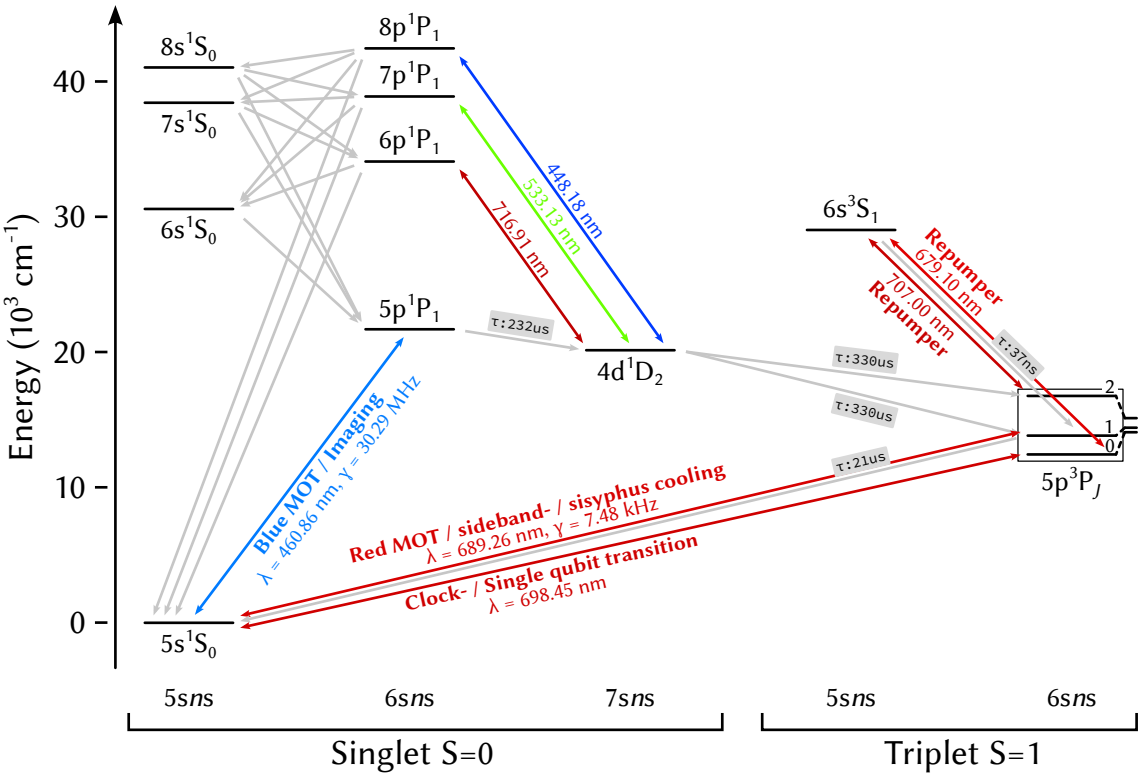


Figure 2.1: **^{88}Sr level scheme.** Decay rates and linewidths are shown for transitions that are relevant in this experiment. The Blue MOT linewidth is used to calculate the Doppler temperature, from where atoms can decay into the $^1D_2$ state, where they can be intercepted by repumpers, or decay into the triplet manifold ^3P_J. Here, atoms are repumped on the 679 nm and 707 nm lines, from which they can decay into the unstable $^3P_1$, so that they end up back in $^1S_0$. The $^1S_0 \rightarrow {}^3P_1$ transition is used for the red MOT, as well as parity projection, sisyphus, and sideband cooling. The $^1S_0 \rightarrow {}^3P_0$ "clock" transition is used for single-qubit operations. Data from [67], [68] and [69], more detailed information can be found in [70].

allowing this process exist close to the blue and red cooling transitions. Furthermore, the atoms are cooled to the ground state on the red cooling transition. Due to the encoding of the qubits $|0\rangle$ and $|1\rangle$, single-qubit transitions are simple Rabi oscillations on the clock transition $5s^1S_0 \rightarrow 5s^3P_0$. Multi-qubit gates are realized using Rydberg excitations, where the interaction radius of the atoms is artificially increased. There is a direct transition from the clock state to a Rydberg S-state, which is discussed in a future thesis [71].

The combination of these features places ^{88}Sr at the forefront of quantum computing, as single-qubit and multi-qubit operations are straightforward single-laser interactions. Moreover, schemes for individual addressing of atoms are currently in preparation, thus allowing high control over each single-qubit. A novel feature in this experiment will

be to shuttle atoms, i.e., transporting them, which can be used as an operation during a quantum computing algorithm. It will be possible to move atoms closer together or further away, thus gaining increasing control when paired with Rydberg blockade effects. Being able to move atoms directly is a powerful tool that is not readily available in quantum computers using printed silicon wafers.

2.2. Preparing Strontium atoms

We have a solid sample of Strontium, which is not isotopically enriched and therefore contains isotopes ^{88}Sr (82.6%), ^{87}Sr (7.00%), ^{86}Sr (9.86%) and ^{84}Sr (0.56%) [72] due to the natural abundance of Strontium, the experiment was built around ^{88}Sr. The sample is heated in an oven included in an AOSense cold atomic beam system (OvenGen3-Sr) to 400 °C and expands in all directions due to its gaseous nature. A 2D-MOT setup cools the atoms transversally, creating a collimated beam of atoms. This section of the chamber contains many stray atoms, affecting the maximally achievable vacuum pressure. It is possible to obtain a high vacuum in the science chamber by adding a differential pumping tube to separate the chambers, ensuring they are connected only through a small aperture. Here, the differential pumping tube has a diameter of 6 mm and length 115 mm. The atoms are directed through the narrow tube and pass through two chambers, again separated differentially with tubes of a 10 mm diameter and 80 mm length. Each is connected to a respective combination pump. The first chamber has a NEXTorr Z 200, and the second chamber a NEXTorr D 500. It may be possible to reduce the system to only one combination pump; however, this was not experimentally tested. The atoms arrive in the final part of the setup, the glass cell. Once there, they are further trapped and cooled in a dual-stage 3D-MOT, after which they are transferred to an optical tweezer array, concluding the initialization phase.

The magnetic field of the 3D-MOT is generated using electric coils, and the tweezer light is focused using an objective. Thus, the experimental apparatus as depicted in Figure 2.2, is separated into five parts:

1. Oven / 2D-MOT
2. Main vacuum pumping
3. Glass cell
4. Coils and objective mounts
5. Rails to move the chamber

The following sections contain more detailed information on the specific parts of the setup.

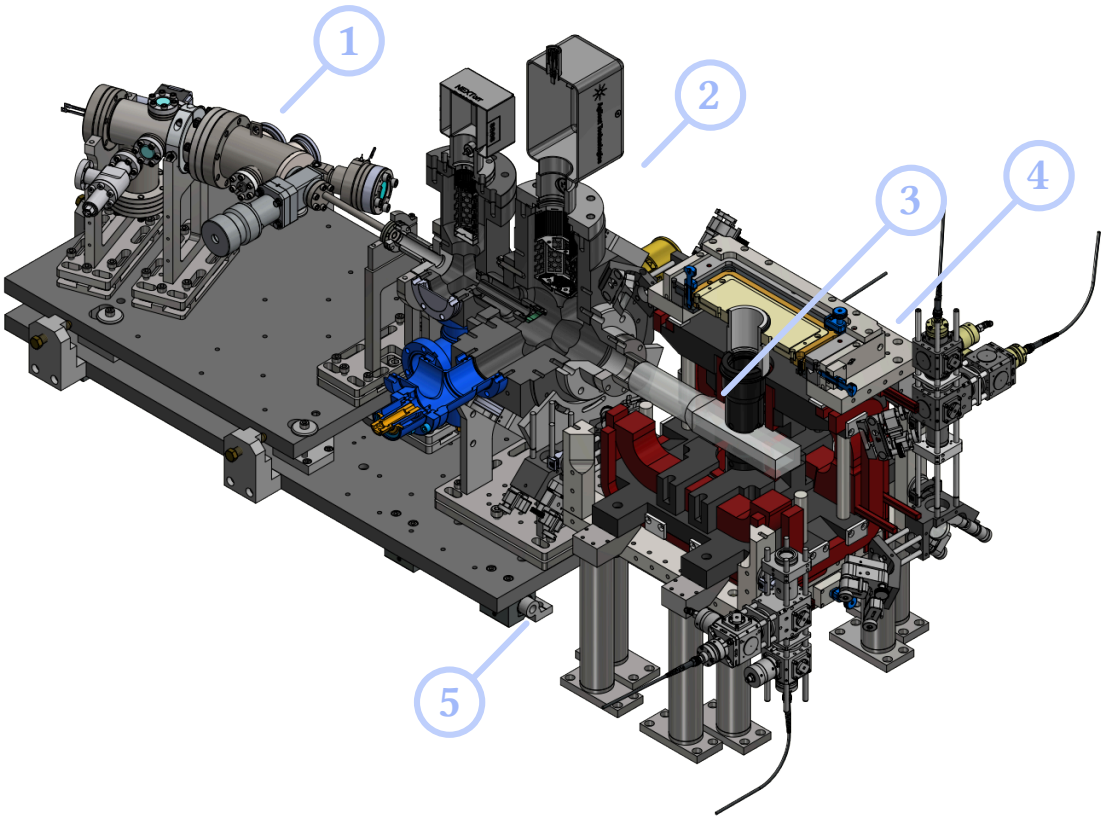


Figure 2.2: **Vacuum system.** The experimental apparatus consists of five main sections. (1) Oven: A solid sample of Sr is heated up such that the atoms expand as a gas. They are then cooled transversally and directed towards section 2. (2) Main pumping section: Ion pumps are used to preserve an ultra-high vacuum around 1×10^{-11} mbar. (3) Glass cell: The experiment is conducted in a glass cell, where the atomic trap is positioned in the center. (4) Magnetic coil and objective mount: Magnetic coils are used to trap atoms and drive various transitions. The objective is used for imaging and addressing of single atoms in dipole traps.

2.3. Rail system

The entirety of the vacuum chamber rests on rails (see Figure 2.3), allowing it to be moved for debugging or alignment purposes. The magnetic coil mount, including the objective mount, remains stationary, and only the vacuum parts (especially the glass cell) move out of the way. This system allows direct access to the atomic plane and is especially useful for all beams that pass through the objective. A glass plate can be inserted between the objective and the atomic plane to substitute the glass cell, and a target with µm-sized holes is used to simulate the atoms, as described in Section 4.6.

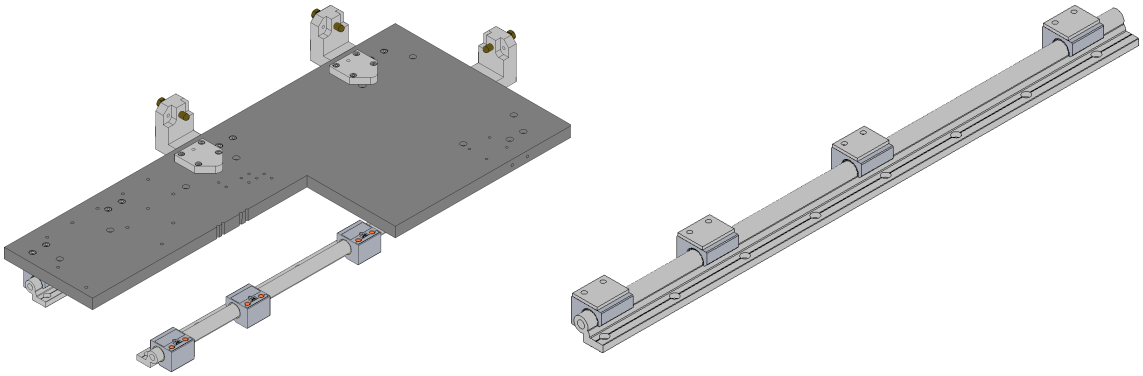


Figure 2.3: **Rail system.** The vacuum chamber is mounted on a breadboard that rests on rails. The left image shows a quarter-section view of the breadboard, the right image one of the two rails. The three sliders are connected to the breadboard, and the chamber can be moved easily unless the sliders are locked.

This setup allows for the observation and correction of aberrations, for example, when the imaging system or the optical tweezers are out of focus.

The rails we are using are Igus WS-20, and we have found that moving the system out and back in does not significantly alter the measurements with the atoms. The MOT retains similar atom numbers, and only a few hours of optimization get the system back to its original state. This is an invaluable tool that can save much work, which might otherwise be just searches in the dark.

2.4. Transverse cooling of gaseous atoms

The first section is an AOSense cold atomic beam system (OvenGen3-Sr). It is pre-pumped by a turbo pump connected to a valve on the side, and an ion pump is permanently connected to further pump and sustain the vacuum. As shown in Figure 2.4, the system contains an oven for heating the Strontium sample, as well as a 2D-MOT section for transverse cooling and a Zeeman slower section for longitudinal cooling. Permanent magnets generate magnetic fields, and laser light is coupled in through two windows for the 2D-MOT and one for the Zeeman slower.

The beams are shaped and coupled according to the manual. Following the instructions, it took us only a few hours to see an atomic beam illuminated by the lasers through the 2D-MOT windows. There are three additional CF-16 windows at the beginning of the section, which can be used to get a first signal of an atomic beam. With the beams constantly on and the lab darkened, we managed to tune the wavelength by looking

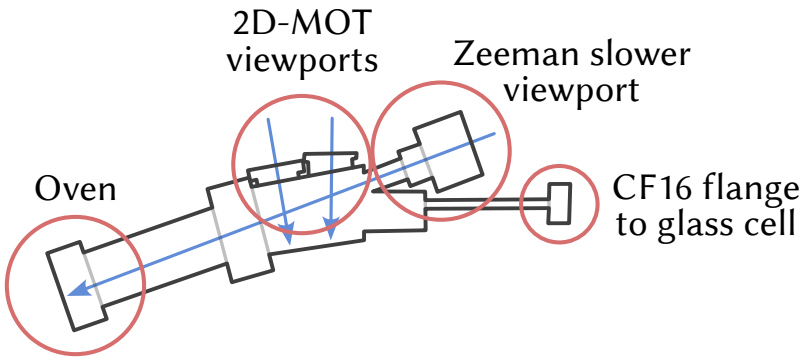


Figure 2.4: **AOSense cooling section.** Atoms are heated in the oven, from where they evaporate uniformly in all directions. They are captured transversally using the 2D-MOT beams and longitudinally cooled using the Zeeman slower. With the remaining velocity, they exit through the narrow tube to the glass cell.

through one of the three windows until an atomic beam emerged. Afterwards, the optimization was a combination of turning the mirrors and changing the wavelength while monitoring the 2D-MOT windows, either by eye or using a camera (a simple webcam works well enough). The final result was a clearly visible atomic beam, as can be seen in Figure 2.5 (a).

To prevent hot atoms from entering the glass cell during an experiment, the 2D-MOT section and everything following are angled relative to the oven section. Thus, atoms will only exit the system when they are redirected by the 2D-MOT beams. Another

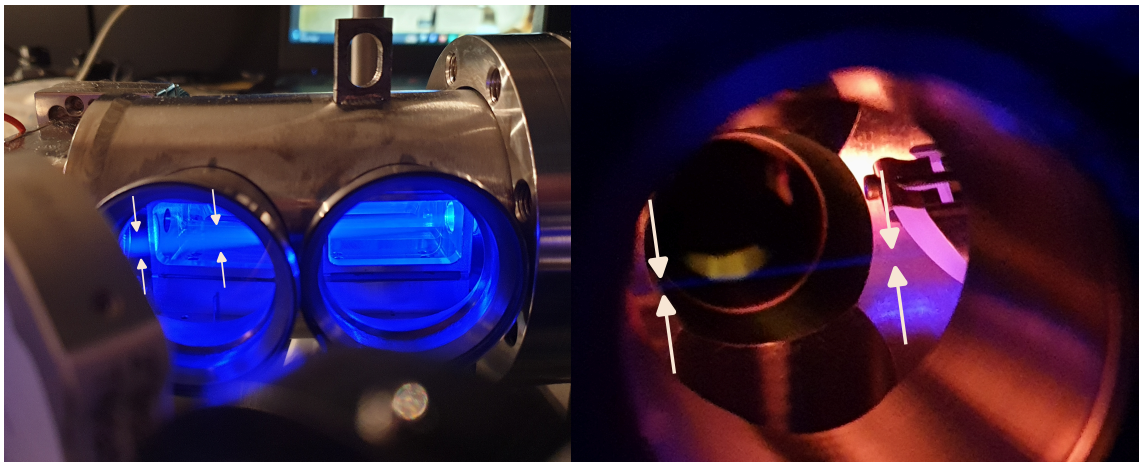


Figure 2.5: **Oven and main pumping section.** The atomic beam is visible in both pictures when the atoms are cooled using blue lasers and consequently fluoresce. The atomic beam in the left image is not yet aligned, as it does not pass into the differential pumping tube and is mostly illuminated by the Zeeman slower beam.

solution used in other experiments [73,74] involves, for example, an atomic shutter that physically blocks the atomic beam.

2.5. Vacuum pumping stage

The oven section ends with a CF16 flange connected to a thin vacuum tube (6 mm diameter), which differentially separates the low vacuum in the oven section, caused by the evaporated Strontium oven, from the high vacuum in the glass cell. The main pumping section has two stages, which again are separated by a differential pumping tube of diameter 10 mm. As in Figure 2.6, the first stage has a NEXTorr Z200 ion getter pump, as well as windows on the side and a CF40 flange with a valve to connect a pre-vacuum pump. A NEXTorr Z200 pumps the second stage and is octagonal. The top face of the octagon connects to the ion pump, the front face has a connection for a hot cathode gauge (Leybold IE 514), and the back face has a CF40 connector with a valve. We also installed an in-vacuum mirror connected to the differential pumping tube, which deflects beams coming from the front surface of the glass cell downwards through a viewport on the bottom surface of the Octagon stage. All other faces of the stage are covered by windows that can be used for imaging the atomic flux. The in-vacuum mirror is coated for 313 nm (>98%) and 670-707nm (>99%) and has a hole of 11 mm, such that the atoms can pass through.

In this configuration, we can achieve pressures converging towards 1×10^{-11} mbar, as measured by the hot cathode gauge in Figure 2.7. The possibility of reaching such low pressure values is partly also due to our choice of vacuum chamber material, which is Titanium Grade 2, further discussed in Section 2.7.

2.6. Glass cell

Atoms are observed in a glass cell, which, compared to a steel or titanium chamber with windows, is more compact and offers the advantage of a dynamic laser layout, as it does not need to be defined beforehand. Its properties are listed in Table 2.1. The cell, as depicted in Figure 2.8, is rectangular, with the back surface open and connected to a CF40 flange. Its design allows for a large amount of optical access, where the inner dimensions are 125 mm × 14 mm × 28 mm.

The dimensions of the glass cell were chosen to allow laser angles of 45 °.

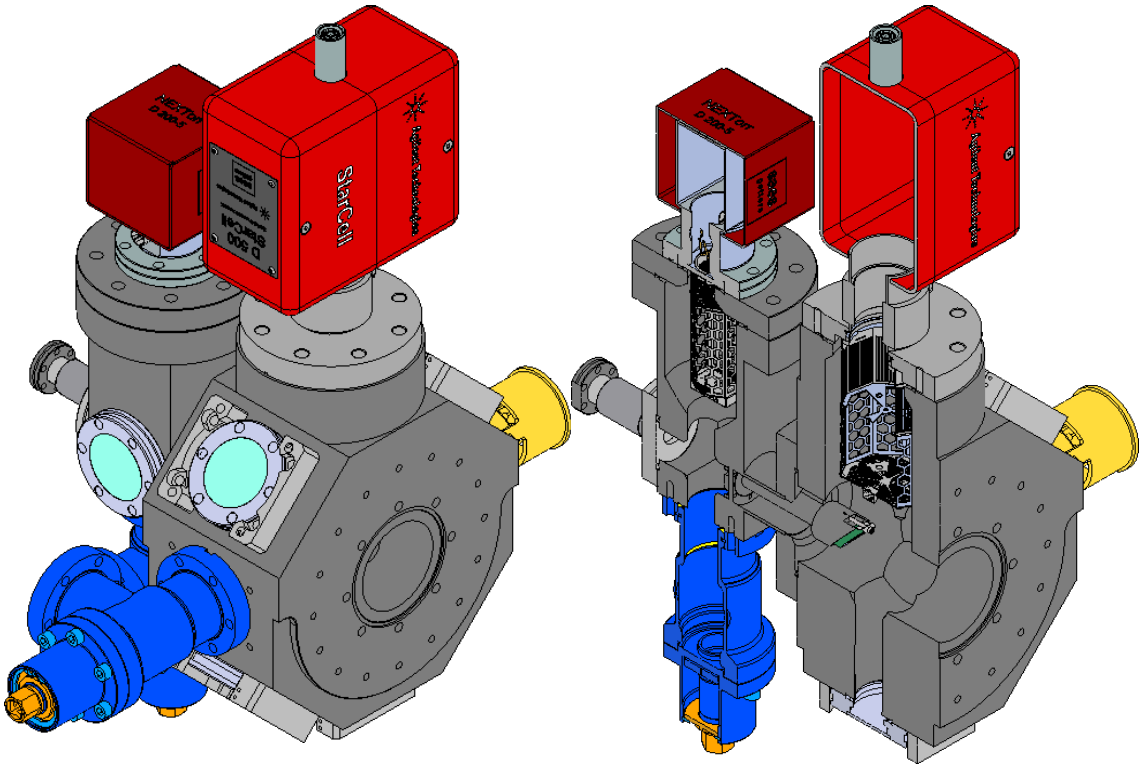


Figure 2.6: **Pumping section.** The vacuum pumping section consists of two ion pumps (see main text). It has connections for a gauge and a pre-pump. The inside features an in-vacuum mirror to deflect beams passing through the front surface of the glass cell.

We specified flatness, parallelism, and optical contact bonding. However, we later discovered that the glass faces are actually heated to enhance bonding, and the flatness is achieved only by polishing the outside face. Nevertheless, using a white-light interferometer built as a Bachelor's project by Jonathan Menssen [75], the glass cell thickness was measured as 6.074 mm, and the relative angle between opposing surfaces is 235 urad. The parallelism is important because back reflections are used to align the objectives, which must be parallel to overlap tweezer configurations from both sides properly. Moreover, when lattices are implemented, non-parallel surfaces can lead to inhomogeneities in the lattice due to interference.

2.7. Choice of material for the pumping stage

Competitive quantum computers require a large number of qubits, and while scalability for tweezer or lattice experiments is, in general, very possible, the likelihood of losing any single atom scales with their total number. Thus, at a large scale, quantum comput-

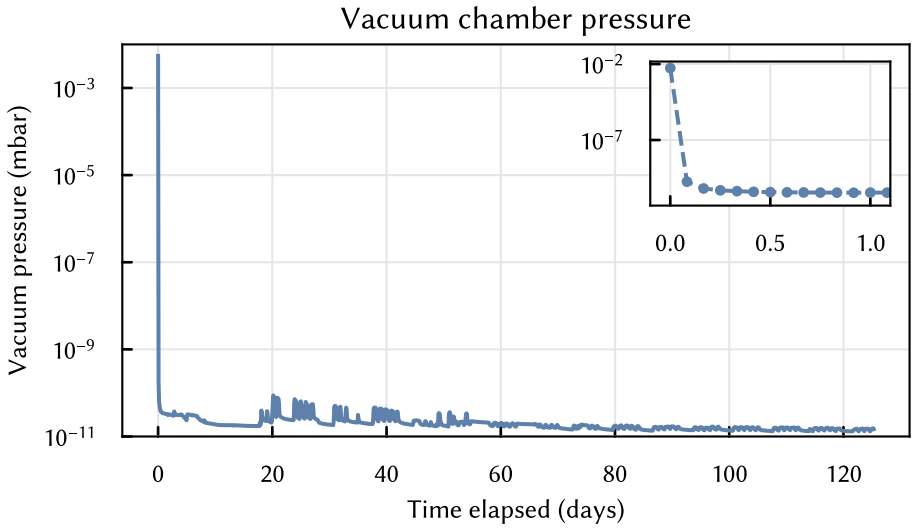


Figure 2.7: **Vacuum pressure after turning on the ion pumps.** The vacuum pressure is monitored using a hot cathode gauge. Data is regularly uploaded to a Grafana dashboard. Irregularities in the data usually occur when the oven is turned on or off. The inset shows the evolution of the pressure on the first day of activation.

ing operations are still limited by atom loss. The leading cause is background pressure from stray (hot) atoms in the vacuum chamber, which collide with the trapped atoms. To reduce this effect, it is important to achieve an excellent vacuum, taking into account the outgassing rates of materials, as indicated by their diffusion coefficients. It is well understood that in stainless steel chambers, hydrogen atoms that are embedded in the material outgas and increase the pressure in the chamber [76]. It is for that reason that the material is heated (fired) before assembly, up to temperatures of 350°C [77,78].

A material study was conducted to determine if it is possible to further reduce the pressure in the system, and Titanium Grade 2 is a viable candidate. It has lower magnetic permeability than Stainless steel (AISI 316 LN), and contains much less hydrogen. A detailed comparison is given in Table 2.2.

Apart from the AOSense oven section and the flange connecting the glass cell, all parts are of Titanium Grade 2. During assembly of the chamber, some leaks were found at the connections between the two materials, which were easily fixed by placing new copper or silver gaskets. These leaks can occur, for example, when materials with different expansion coefficients are combined and then heated (as is the case here).

After the leaks were fixed, we observed no limitations in choosing Titanium as the material for the pumping stage. We can reach 1×10^{-11} mbar pressure in the chamber,

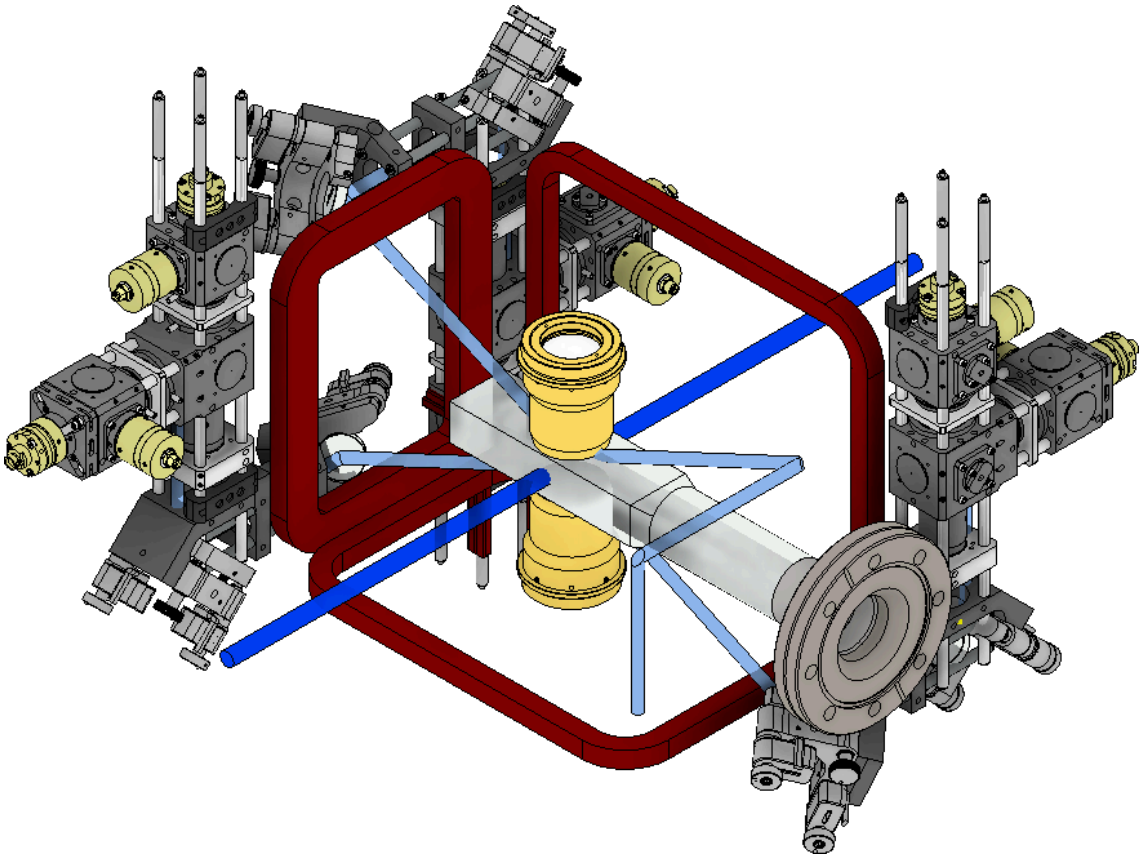


Figure 2.8: **MOT beam arrangements.** The Schäfter+Kirchhoff cube system has fiber ports for both red and blue MOT beams, which are combined in an internal dichroic. They pass through mirrors and then enter the glass cell diagonally on the xz-plane. The horizontal MOT beams are prepared on a breadboard, which also contains other beams that enter the xy-plane. Three coils and one S+K cube system were removed here for visibility; however, the system is symmetric, as seen in e.g. Figure 2.2 or Figure 2.9.

which is already limited by the hot cathode gauge IONIVAC IM 540 (see Figure 2.7). We opted to turn off the hot cathode gauge because the reading is outside its measurement range, and its hot nature would only contribute to more outgassing.

2.8. Magnetic coils

Apart from driving the atoms with lasers, magnetic coils are used in the experiment for cooling and trapping in the MOT, as well as cooling during sideband cooling. Moreover, the clock transition, which is used in the qubit encoding, is forbidden unless a magnetic field is applied, since then there is a mixing between the $^3P_0$ and $^3P_1$ states [83–85]. It

Manufacturer	**JapanCell**
Connectorized by	Precision Glassblowing
Material	OHARA SK-1300
Outside dimensions	131 mm x 26 mm x 40 mm
Glass thickness	6 mm
Coating	None
Bonding	Optical contacting
Parallelism to opposite wall	<0.1 mm
Flatness	< $\lambda/10$

Table 2.1: **Specifications for the glass cell.** As the glass cell is the main section where experiments are performed, its specifications are critical. Especially the material, the flatness as well as parallelism all have implications for beams entering the cell.

is desirable to have a high field, due to the linear scaling of the Rabi frequency on this transition with magnetic field:

$$\Omega_{\text{clock}} = \alpha\sqrt{I_{\text{clock}}}\ |\mathbf{B}|\cos(\theta) \tag{2.4}$$

where $\alpha = 2\pi \times 202.1(2)\frac{\text{Hz}}{\text{T}\sqrt{\text{mW/cm}^2}}$ [70,86,87].

To trap atoms in a MOT, opposing coils need to be in an anti-Helmholtz configuration, whereas for all other measurements, the coils would be in a Helmholtz configuration. It is possible to switch between these settings using an H-Bridge electronic circuit; however, we decided to use two separate pairs of coils for the time being. Here, the MOT coil pair will be called “Bias Y”, and “Bias Z” is used to enable the clock transition.

Additionally, it is helpful to cancel outside fields, so there are an additional three pairs of offset coils in the x,y, and z directions. These are also useful to slightly move the position of the MOT. Geometric properties are given in Table 2.3, so that the coils are assembled into the configuration Figure 2.9. Electrical properties were measured and are compiled into the table Table 2.4. Fields were measured using a magnetic field sensor placed approximately at the center between coil pairs with an applied current of 5 A. Measurements for voltage drop, resistance, and inductance were taken from a single coil using an LCR meter. Theory simulations were carried out using Comsol.

The fields can be modeled analytically using the Biot-Savart law [88]:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0 I}{4\pi}\int \frac{d\mathbf{l}' \times \hat{\mathbf{r}}}{r^2} \tag{2.6}$$

	Stainless steel 316 LN	**Titanium Grade 2**
Composition (percent)	C: ≤ 0.012	C: 0.007 to 0.012
	Si: 0.33 to 0.49	N: ≤ 0.01
	Mn: 1.59 to 1.82	Fe: 0.08 to 0.12
	P: 0.23 to 0.31	O2: 0.1 to 0.14
	S: 0.001 to 0.0018	other total: ≤ 0.17
	Cr: 17.16 to 17.38	Ti: remainder
	Mo: 2.6	
	Ni: 12.24 to 13.55	
	N: 0.1636 to 0.1636	
	Co: ≤ 0.04	
	Fe: remainder	
Magnetic permeability	1.008	1.00005
Specific heat capacity	0.500	0.523
at 20°C (J g^{-1} °C)		
Thermal conductivity (W m^{-1} K)	16.3	16.4
Hydrogen (H2)	7.0e-10	2.5e-12
outgassing rate (Pa L s^{-1} cm^{-2})		
Hydrogen concentration (ppm)	≤ 150[1]	5-150 [79]

Table 2.2: **Material properties of 316LN stainless steel vs Titanium Grade 2.** Titanium has a much lower H2 outgassing rate and magnetic permeability, which makes it attractive in low-pressure applications, where Eddy currents can build up due to strong magnetic fields. Properties from [80–82] unless otherwise noted.

Simply put, the magnetic field is integrated over segments $d\mathbf{l}'$ of the conductor in the direction of current flow I, measured from the position $\mathbf{r}$. The offset fields are rectangular, allowing the integral to be split into four wire segments with finite length $\frac{L}{2}$. This results in the following equation [89], where the wire is oriented along the z-direction. The equation is given in cylindrical coordinates, with $r = \sqrt{x^2 + y^2}$ and the field distribution along φ is uniform:

$$B_{\text{wire}}(r, z; L) = \frac{\mu_0 I}{4\pi r}\left(\frac{z + L}{\sqrt{r^2 + (z + L)^2}} - \frac{z - L}{\sqrt{r^2 + (z - L)^2}}\right) \tag{2.8}$$

[1]The specification ASTM B265 / SB265 lists ≤ 150ppm as the hydrogen concentration in Titanium grade 2, however we opted to request materials which had measured concentrations ≤ 100ppm

Coil pair	Width (mm)	Height (mm)	Radius (mm)	Thickness axially (mm)	Thickness radially (mm)	Number of windings	Separation (mm)
Offset X	155	204		13	26	2x4	264
Offset Y	250	200		13	13	2x2	182
Offset Z	250	172		13	13	2x2	174
Bias Y	130	250		35	35	2x 2x5	97
Bias Z			91	65	13	5x1	61

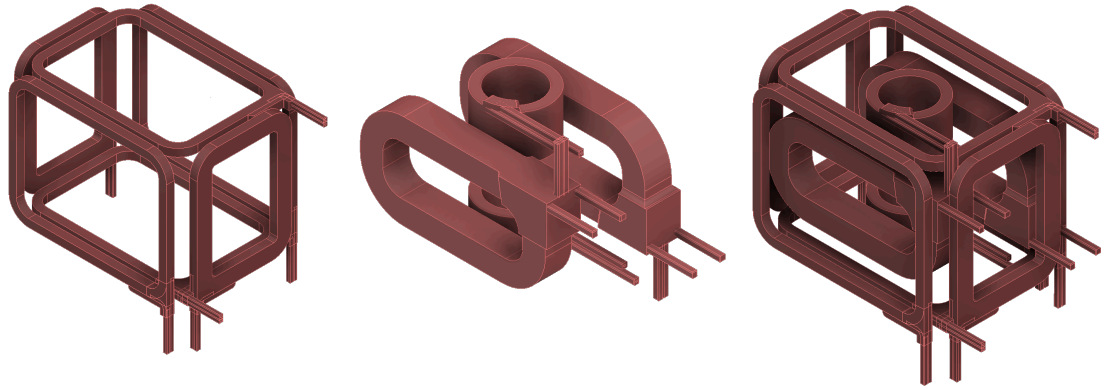
Table 2.3: **Geometrical properties of the magnetic coils.** The offset coils are rectangular, whereas the Bias Y coil is mostly rectangular and the Bias Z coil is circular. The thickness is mostly given by the quadratic wire profile ($5.3\,\mathrm{mm} \times 5.3\,\mathrm{mm}$) plus some buffer for the epoxy. The windings are given as the number of windings in the axial direction times the number of windings in the radial direction. The height, width, and radius are outside dimensions. The separation is the inner separation of two coil pairs

The rectangular coils are two pairs of two identical wires, such that the equation for the magnetic field along the central axis of a rectangle with width w and height h becomes:

$$B_{\mathrm{rectangle}}(z;\mathrm{w},\mathrm{h}) = 2B_{\mathrm{wire}}\left(\sqrt{(\mathrm{w}/2)^2+z^2},0;\mathrm{h}\right) + 2B_{\mathrm{wire}}\left(\sqrt{(\mathrm{h}/2)^2+z^2},0;\mathrm{w}\right) \tag{2.10}$$

For circular loops of radius R, the equation becomes [88]:

$$B_{\mathrm{circle}}(z;\mathrm{R}) = \frac{\mu_0 I}{2}\frac{R^2}{(R^2+z^2)^{3/2}} \tag{2.12}$$



(a) Offset coils (b) Bias coils (c) Complete coil configuration

Figure 2.9: **Configuration of the coils.** Dimensions are given in Table 2.3.

Coil pair	Voltage drop (V)	Resistance mΩ	Inductance (uH)	Magnetic field (G)	Magnetic field vs Current, measured (G/A)	Magnetic field vs Current, theory (G/A)
Offset X	0.315	63	22.1	0.779	0.156	0.168
Offset Y	0.360	72	15.1	1.143	0.190	0.188
Offset Z	0.368	73.6	13.5	1.137	0.198	0.198
Bias Y	0.275	55	25.1	9.34	1.83	1.875
Bias Z	0.310	62	32.5	6.50	1.27	1.24

Table 2.4: **Measured electrical properties of the magnetic coils.** Measurement of the field was performed at 5A current in the center of the coil pair in Helmholtz configuration using a Hall sensor.

Using the analytical expression for the coils, measured magnetic field values are compared to this result in Figure 2.10 . There is some deviation, likely due to measurement inaccuracies. In fact, as seen in the figure, small changes in the position of the sensor already have significant effects on the measurement outcome. The Python code incorporating the analytical functions is given in Appendix A..

2.8.1. Fast coil switching

The blue MOT is the first three-dimensional cooling stage; however, its feature is quite broad, limiting cooling to a few hundred μK. Atoms are transferred to a second red MOT stage, which can cool down to a few hundred nK [70]. Transferring the atoms quickly is key to preserving high atom numbers and densities. The limiting factor in the fast transfer is reducing the MOT coil currents from its initial 230 A (420 G) down to 15 A (28 G).

To achieve this, one method is to use an insulated gate bipolar transistor (IGBT) as a fast switch, unloading the current into a capacitor, as shown in Figure 2.11. When the switch is turned off, the circuit becomes floating. Then, the inductor (the coil) generates a negative voltage to oppose the sudden change in current [88].

The conductor is still charged with the original current, which is drained by a capacitor. To prevent the current from oscillating between the capacitor and the inductor, a diode is placed after the capacitor (taking into account the polarity of the voltage, which has

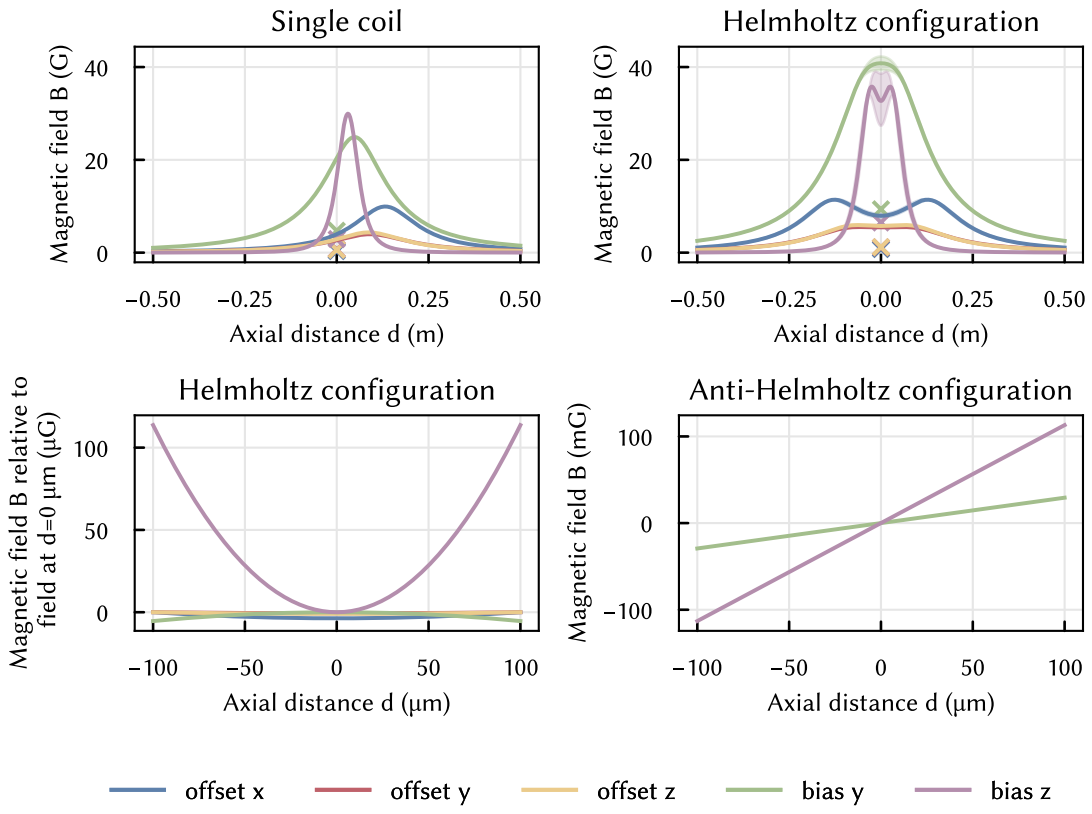


Figure 2.10: **The magnetic field of the coils as a function of axial distance.** The plots show the magnetic field according to theory, at an applied current of 5 A. The measured field values are indicated with arrows, where the probe remained in the center of the coil pair for both cases. In the left plot, only a single coil was turned on, followed by the activation of both coils in a Helmholtz configuration. A zoom-in of the Helmholtz configuration is shown, where the total field was subtracted by the field at $d = 0\,\mathrm{m}$ for each coil pair, respectively. The plot to the right shows the field gradient in anti-Helmholtz configuration, which is $0.015\,\mathrm{G\,m^{-1}}$ for bias y and $0.056\,\mathrm{G\,m^{-1}}$ for bias z.

changed). This means the capacitor will now charge up, and the inductor's current falls off. To discharge the energy, a resistor is placed parallel to the capacitor, which dissipates the current in the form of heat.

When the switch is flipped, the inductor generates a reverse voltage to compensate for the current. This voltage reaches a maximum when the current is zero, a condition that the IGBT has to withstand. To understand the magnitude of the voltage spike, we need to solve the differential equation of the system when the circuit is floating. First, without the resistor, the system follows the equation:

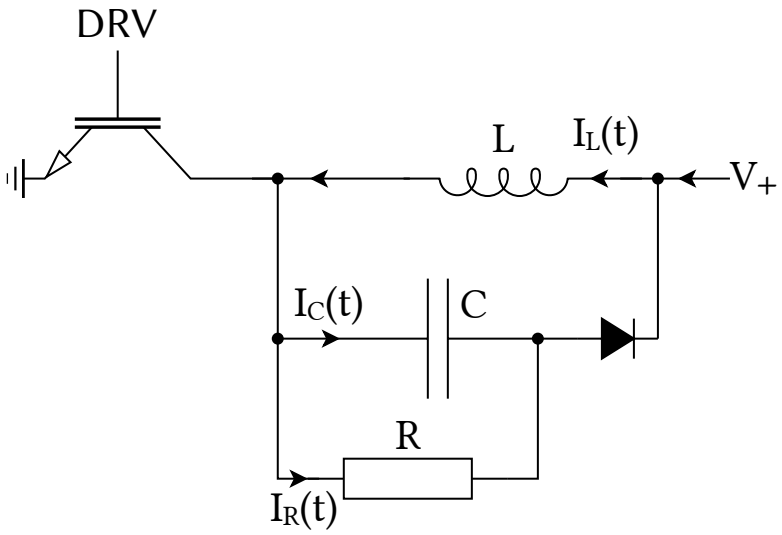


Figure 2.11: **Circuit for fast switch-off of High field coils.** Current flows from V_+ to GND, when the switch is open. When it is closed, the capacitor drains current from the inductor (the coil) and the resistor dissipates the energy of the capacitor into heat. When the circuit is floating, the direction of the current flow is inverted, so the diode prevents backflow from the capacitor to the inductor.

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{dt}^2} I_{L(t)} + \omega_0^2 I_{L(t)} &= 0 \\ U_L(t) - L\frac{\mathrm{d}}{\mathrm{dt}} I_L(t) &= 0 \end{aligned} \tag{2.14}$$

Where $U_L(t)$ is the voltage across the inductor, $\omega_0^2 = \frac{1}{LC}$. The initial conditions are $I_L(0) = I_{\max}$ and $U_L(0) = 0$. Then the solutions to the differential equations are:

$$\begin{aligned} I_L(t) &= I_{\max} \cos(\omega_0 t) \\ U_L(t) &= -U_{\max} \sin(\omega_0 t) \end{aligned} \tag{2.16}$$

$U_{\max}$ is the voltage spike when the current is zero; thus, the maximum voltage rating of the IGBT needs to be $U_{\max} + V_+$. Using the oscillation frequency, we can find the minimum capacitance of the capacitor:

$$\begin{aligned} \omega_0 &= \frac{1}{\sqrt{LC}} = \frac{U_{\max}}{L I_{\max}} \\ C &\geq L\frac{I_{\max}^2}{U_{\max}^2} \end{aligned} \tag{2.18}$$

Using a conservative inductance of 100 uH and a maximum target current of 400 A, we can settle on $U_{\max} = 1200$ kV and find $C \approx 10$ uF.

The switch-off time is one-fourth of the period, so $t_s = \frac{\pi}{2\omega_0} \approx 50$ us.

Including the resistor in the differential equation adds a damping term:

$$\frac{\mathrm{d}^2}{\mathrm{dt}^2} I_{L(t)} + L\frac{\omega_0^2}{\omega_{\mathrm{LR}}}\frac{\mathrm{d}}{\mathrm{dt}} I_L(t) + \omega_0^2 I_{L(t)} = 0 \tag{2.20}$$

With the oscillation frequency $\omega_{\mathrm{LR}} = R/L$. We are only interested in finding a maximum rating for the IGBT, and since the damping term reduces the voltage, this does not affect the main result. However, the switch-off time t_s changes to

$$t_s(R) = \frac{\pi}{2\sqrt{\frac{1}{LC} - \frac{R^2}{4L^2}}} \tag{2.22}$$

which for $\frac{1}{LC} > \frac{R^2}{4L^2}$ means it is increasing. Using the values from above, however, the resistance R must be less than 6 Ω and our choice of $R = 5\,\Omega$ increases t_s by a factor 1.8.

This results in the switch-off times we see in Figure 2.12, which is ≈ 300 µs / 420 G.

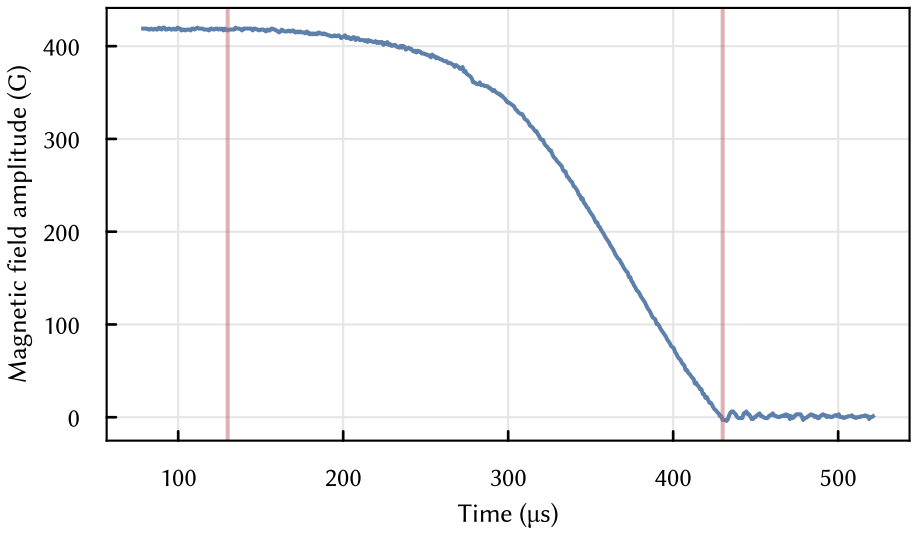


Figure 2.12: **High field coil rampdown.** Shown is the time it takes the high field coils to ramp down from 420 G to zero. The data was taken with a transducer around the wire to the coil and thus does not show residual currents around the chamber. The rampdown duration between vertical lines is 300 µs.

2.9. Objectives

In optical tweezer experiments, atoms are trapped in the focus of an array of laser beams, where the potential depth depends on the beam's waist due to the Stark effect. Generally, objectives are used in these experiments to correct for aberrations for short focal distances and achromatic shifts when working with multiple wavelengths. The potential is inversely proportional to the squared waist of the laser beam:

$$U(r) = U_0 * \left[1 - 2\left(\frac{r}{w_0}\right)^2\right] \tag{2.24}$$

which means it is proportional to the square of the numerical aperture NA:

$$\mathrm{NA} \approx \frac{\lambda_0}{\pi \omega_0} \tag{2.26}$$

Hence, it is desired to have an objective with a high numerical aperture. As a large NA means a lower working distance, there is a theoretical limit to how high the NA can be. Ultimately, the largest NA possible is achieved for an in-vacuum objective, which is not feasible for glass-cell type experiments. Moreover, if multiple laser beams are passing through the objective, it must correct for chromatic aberrations and may need to be coated for these wavelengths as well. Its size has to fit into the experimental setup, especially when there are additional beams that have to closely pass the objective (as is the case here with the MOT beams, see Section 2.10). Even though the Stark effect is leveraged to trap atoms, it also becomes problematic in cases where a transition is not magic, meaning the two states experience different shifts. Therefore, objectives should also introduce as little wavefront error as possible. Three objectives were characterized in Table 2.5.

Part number	Strehl ratio	Transmission
57-31-24M 461-813 16161A-1	0.827	0.89
57-31-24M 461-813 16161A-2	0.913	0.88
57-31-24M 461-813 16161A-3	0.891	0.88

Table 2.5: **Objective characterizations.** The objectives used in the experiment were characterized in a separate setup. A collimated beam enters a pinhole and is imaged by the objective onto a camera. After careful alignment of the pinhole and the objective with respect to the pinhole, the Strehl ratio can be extracted.

The experiment uses two objectives, one above and one below the glass cell. The lower one is used to create the tweezer array, as well as to collect fluorescent photons off of atoms illuminated by the imaging laser. It is also designed to focus on addressing beams either on the $^1S_0 \rightarrow {}^3P_1$, transition, or on the $^1S_0 \rightarrow {}^3P_0$ transition. Similarly, the upper objective can be used in the future to focus a multi-qubit addressing laser and to add an imaging stage to double the number of collected photons.

2.9.1. Objective mount

Aligning the objective requires careful attention - its face must be parallel to the glass surface of the cell and as close as possible. In this case, the distance between the two surfaces is 1 mm. In the case of the lower objective, its focus must be at the position of the MOT. More complicatedly, the foci of a laser array through the upper objective must perfectly overlap a similar array from the lower objective.

These considerations provide five degrees of freedom: two angles for aligning the objective with the glass cell, and three positions. The mounts were designed so that these degrees of freedom can be easily tuned as independently as possible. Piezo actuators are necessary because manually turning a screw applies too much force and affects other degrees of freedom. However, since they can generate a large electric field, it must be evaluated whether leaving them on affects the Rydberg physics.

The design is depicted in Figure 2.13. It incorporates an outer frame to connect the structure to the optical table. Two screws are used for xy translation. The three other screws are used (in combination) for tip/tilt and z-translation. For all components around the glass cell, eddy currents are always a concern. As such, slits are introduced, and the screws connecting the plates are isolated. Springs are made of non-magnetic steel, while all other metal parts use titanium due to its low magnetic permeability (see Table 2.2). For the stage, MACOR was deemed preferable because it is much stiffer and so should not change its shape over time. However, machining MACOR is a delicate task, and we couldn't acquire the part in time. Instead, we settled on PEEK as an alternative.

The lenses in the objectives are made of fused silica, and we use aluminum oxide (Al2O3) for the housing, a ceramic with high electrical isolation and a low temperature coefficient. The choice of materials makes sure that magnetization is minimized, since that is a concern due to the high fields during the MOT stage and for driving the $^1S_0 \rightarrow {}^3P_0$ and $^1S_0 \rightarrow {}^3P_2$ transitions and would deform the stage considerably otherwise. PEEK and MACOR are also stiff and very light, reducing long-term instabilities.

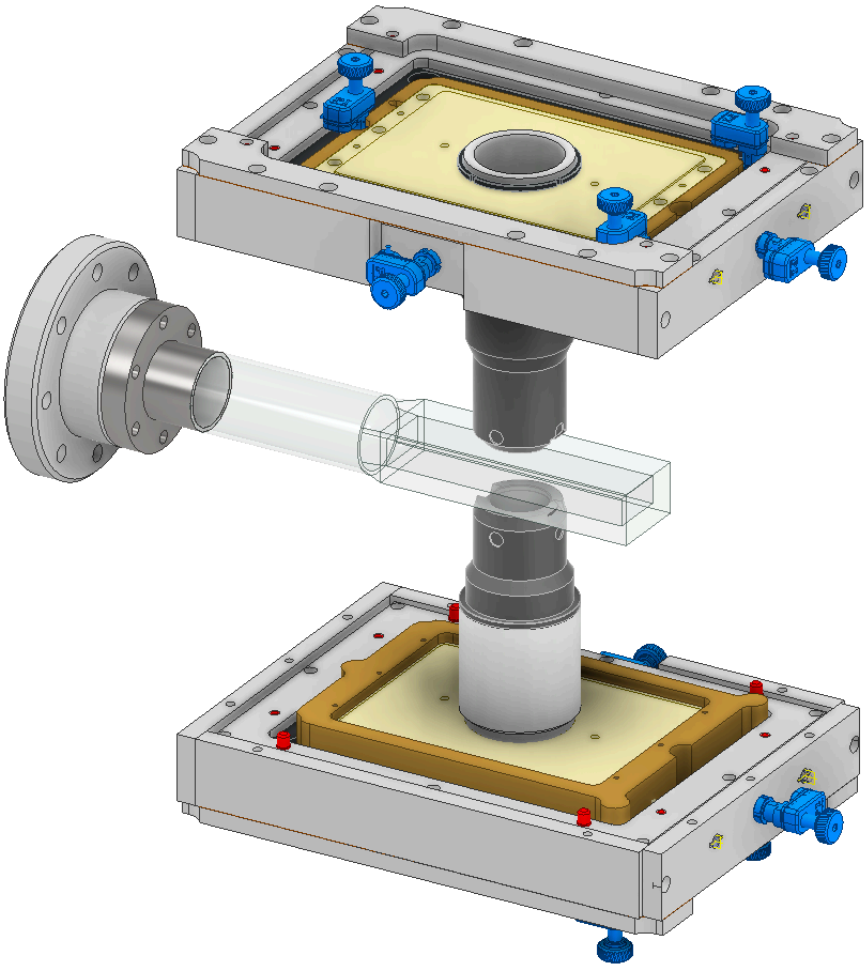


Figure 2.13: **A five-axis stage to accurately position the objectives into the focus of the atomic plane.** The stage consists of translation, as well as tip/tilt components.

The objective was also designed to have carvings on the side for better optical access to the diagonal MOT beams. It also contains mounting positions for mirrors that can be glued to the side, allowing for interferometric stabilization of the objective vibrations later on.

2.10. Compact MOT laser launcher

Part of the project is having a compact, modular setup that will have future iterations. MOTs is now a standard part of neutral atom experiments and is a solved problem in

experimental physics. Since the project aims to produce a compact, maintenance-free system, we decided on a fixed setup of laser beams to trap the atoms, which is mounted on the structure surrounding the glass cell.

Also, the top and bottom sides of the glass cell are not accessible due to the objectives. Only one face on the x-axis is open for laser access, while the other connects to the vacuum chamber. Thus, we place the beams in the xz-plane, and even though there is some reflective loss, it is a good compromise to have the beams out of the way for more important lasers later. The MOT beam launchers, designed in collaboration with Schäfter+Kirchhoff, are screwed into the side of the titanium cube. They use the S+K multicube system, where the beam from a fiber collimator passes a beam sampler into a photodiode, and the polarization can be adjusted using a $\lambda/2$ waveplate. Two of these are combined in a dichroic (one for red, one for blue), and the beams are then launched into the chamber via two coupling mirrors. These mirrors are motorized, which aids in the alignment procedure and maintains the potential for an automated MOT alignment process in the future. This setup has the advantage of being robust, since it is isolated from the main breadboards surrounding the chamber, which are occasionally accessed to install new optics and lasers. Moreover, it makes efficient use of the space and does not occupy important space. The launchers are detailed in Figure 2.14, depicted as a cage setup of two fibers connected to two mirrors, which redirect the diagonal beams into the glass cell.

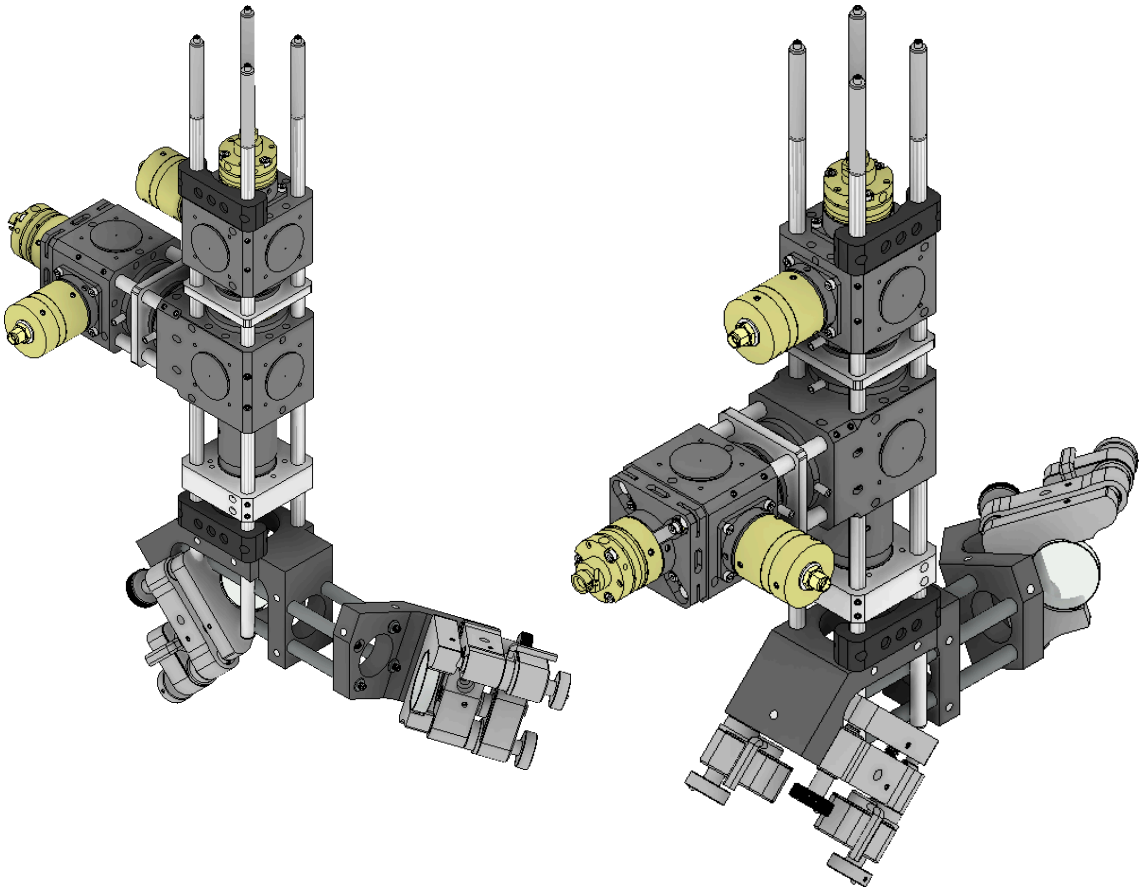


Figure 2.14: **Schäfter+Kirchhoff cube system for MOT beams.** The system is a compact way to combine and launch diagonal MOT beams into the glass cell. They are mounted directly on the titanium structure around the glass cell (see Figure 2.2). It contains two fiber ports, one for 461 nm and one for 689 nm, each corresponding to one of the two MOT stages. The beams pass through a beam sampler, which sends a fraction of the light to a photodiode, then adjustable $\lambda/2$ waveplates before being combined on a dichroic. Afterwards, two mirrors are used to align the beam centrally into the glass cell.

3. Rack-based laser systems

Trapping and cooling neutral atoms is a well-known technique, especially for ^{88}Sr, because it is used as a clock standard. Moreover, a neutral-atom-based experiment has a strong focus on the technical aspects. Thus, a goal of this project was to make the initialization phase as maintenance-free as possible. With the current technologies, there are some limits. Diode lasers are easy to build; however, they degrade over time and have to be exchanged. Temperature and mechanical instabilities will reduce couplings, such as into AOMs or fiber couplers. Moreover, care must be taken in selecting optical elements, as a focused, high-power beam can burn surfaces due to dust particles or glue used to combine optics.

A custom laser rack was built to create setups that minimize outside influence and have beam paths that are easy to debug.

Apart from the lasers used in the initialization phase, there is also a laser used for trapping, one for driving the clock transition (used in single-qubit rotations), and another for driving the Rydberg transition (used for multi-qubit gates).

The following chapters provide insight into all the laser setups required for ultracold strontium tweezer experiments. Locking schemes are collectively explained at the end of this section.

3.1. General idea of the rack-based setup

We use Toptica ECDLs as seed lasers for the red and blue light that is used in the initialization of the atoms. In the case of the blue light, the same 461 nm seed is used for:

- Blue 2D MOT
- Blue 3D MOT
- Zeeman slower
- Imaging

Similarly, for the red light at 689 nm, the same seed is used for:

- Red 3D MOT

- Sisyphus cooling
- Sideband cooling
- Addressing

Using one seed for multiple purposes is advantageous because it provides a constant frequency reference. AOMs are consequently used to shift the resonance relative to the base frequency. The output power of the blue diode laser is 150 mW, and the red laser produces 5 mW of coherent light. The light is amplified in separate stages to produce enough output power for all the cooling beams. In our case, this is achieved through home-built injection lock amplifiers (ILAs), where we use one seed for multiple ILAs to disentangle the different projects, thus making it easier to debug problems.

All the following laser setups are installed in a custom laser rack, which is a modification of a standard server rack. The breadboards are water-cooled to maintain a constant temperature, and the doors are also closed to minimize air currents. The breadboards rest on extendable rails, and all cables and fibers exit toward the inside of the rack. There, they enter energy chains, which are used to maintain a well-defined position for the cables and act as strain relief on the outputs. The setups on the rack breadboards are designed with the following ideas in mind:

- All fiber outputs must contain a $\lambda/2$ and polarizing beam splitter (PBS) to clean the polarization (converting polarization noise into intensity noise), followed by a beam sampler into a photodiode for monitoring.
- All fiber inputs must have $\lambda/4$ and $\lambda/2$ waveplates to align the polarization axis into the fiber
- All fibers and cables must exit toward the rack center into energy chains connected to the breadboard.

Pictures of the rack and rack setups are in Figure 3.1 and Figure 3.2.

3.2. Lego optics mounts

Space for laser optics is a scarce resource in quantum optics experiments, especially in front of the experiment chambers, where multiple lasers have to enter the system. While most groups still use clamped mounts with M6 screws, a custom optical mounting system was developed at the Max-Planck Institute for Quantum Optics (MPQ) (see [73]), which uses mostly 1/2″ optics where possible and fixed posts with holes for M3 screws, to avoid clamping the optics on cylindrical posts.

Figure 3.1: **Laser rack prototype.** Picture of the first laser rack prototype, which is currently used in this experiment.

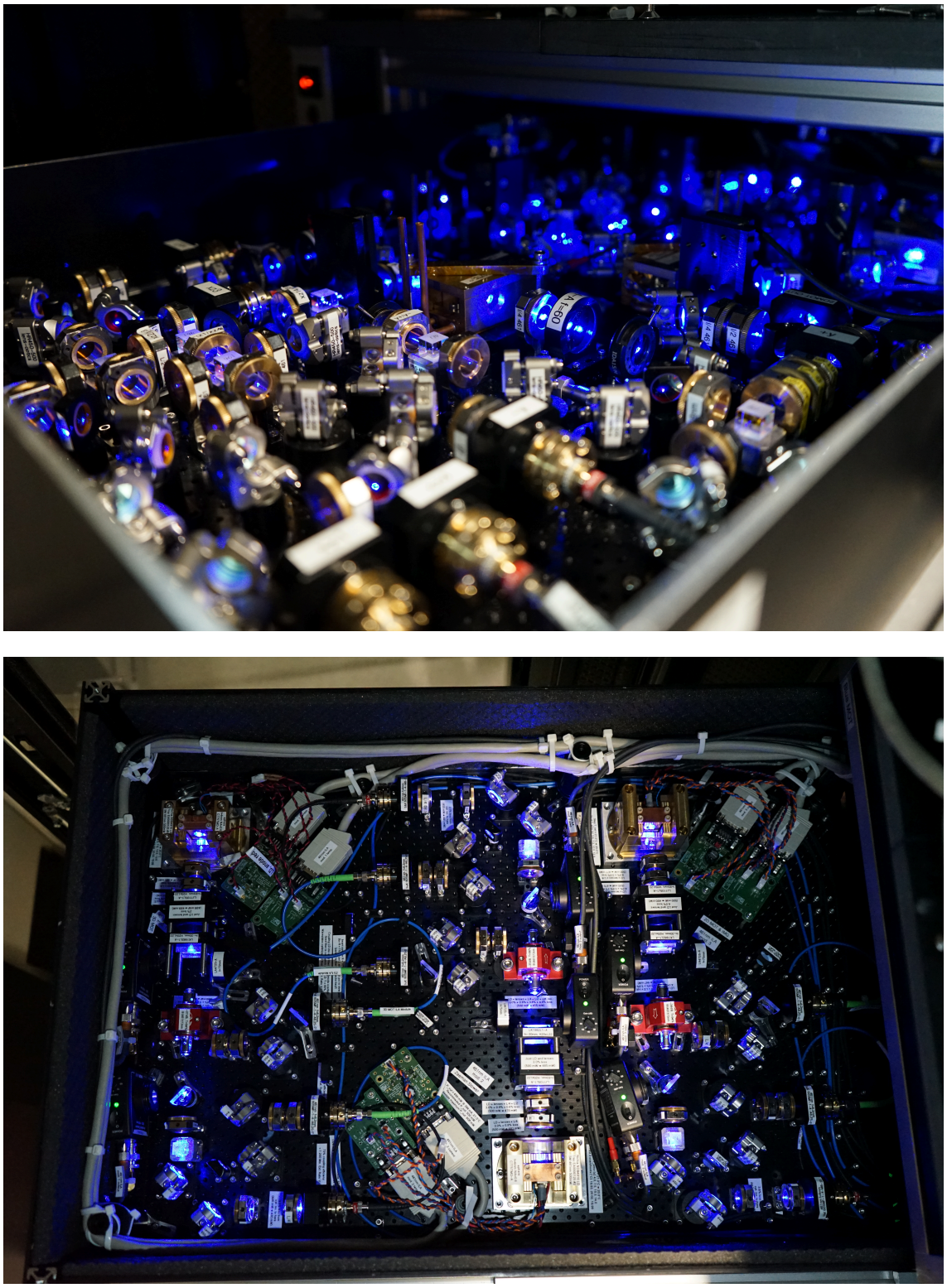


Figure 3.2: **Laser setups in the rack.** Picture of the blue cooling lasers in the laser rack (top) and of the blue injection locks (bottom).

3.3. Amplifying via injection locking

Part of the idea of having maintainable setups was to modularize parts that should be easy to exchange. Due to time constraints, it was left as a task for future machines, now that we have a better understanding of what the setups may look like. However, in the case of the injection locks, a standard setup was established. The injection locks for blue are on one breadboard, just like the red lasers.

The setup itself is depicted in Figure 3.3. The beam enters through a fiber port and is split to a monitoring photodiode. To reduce laser noise, the laser diode is preceded by an isolator with the direction pointing away from the diode. Light entering the isolator in the backward direction is set to S-polarized light by the first polarizer, then rotated by 45° and filtered out by the second polarizer, which is rotated by 135° with respect to the first polarizer. In contrast, light entering through the reflective port of the first polarizer is P-polarized and is therefore transmitted through the second polarizer. The main beam path thus enters the isolator through the reflective port of the first polarizer. The beam is shaped using two cylindrical lenses to match the diode's mode, and the polarization is adjusted with waveplates.

The injection-locked diode outputs light back via the same path, and the light is transmitted through the isolator. The light is sent to a monitoring diode directly after the isolator, where only one beam remains. It is then further split towards a port for the wavemeter and another one as the main output.

The wavemeter has proven to be a valuable tool to check the locking status of the injection-locked diode. If the wavelength or the output of the interferometer in the wavemeter does not match the seed's mode, then the diode is not locked. The current is then adjusted until there is a match.

The injection lock is used for 461 nm and 689 nm. The isolators used are Thorlabs IO-5-440-HP and IO-5-670-HP, respectively. For blue, we use a Nichia NDB4916 laser diode, and for red, a Ushio HL69001DG.

3.3.1. Automatic injection relocking

In addition to using the wavemeter to check the lock of the ILA, it is also possible to monitor the laser power on a photodiode near the output of the injection lock. As the current of the diode is modulated, bumps can be seen whenever the diode becomes single-mode, as less power gets distributed into other modes. This observation can be

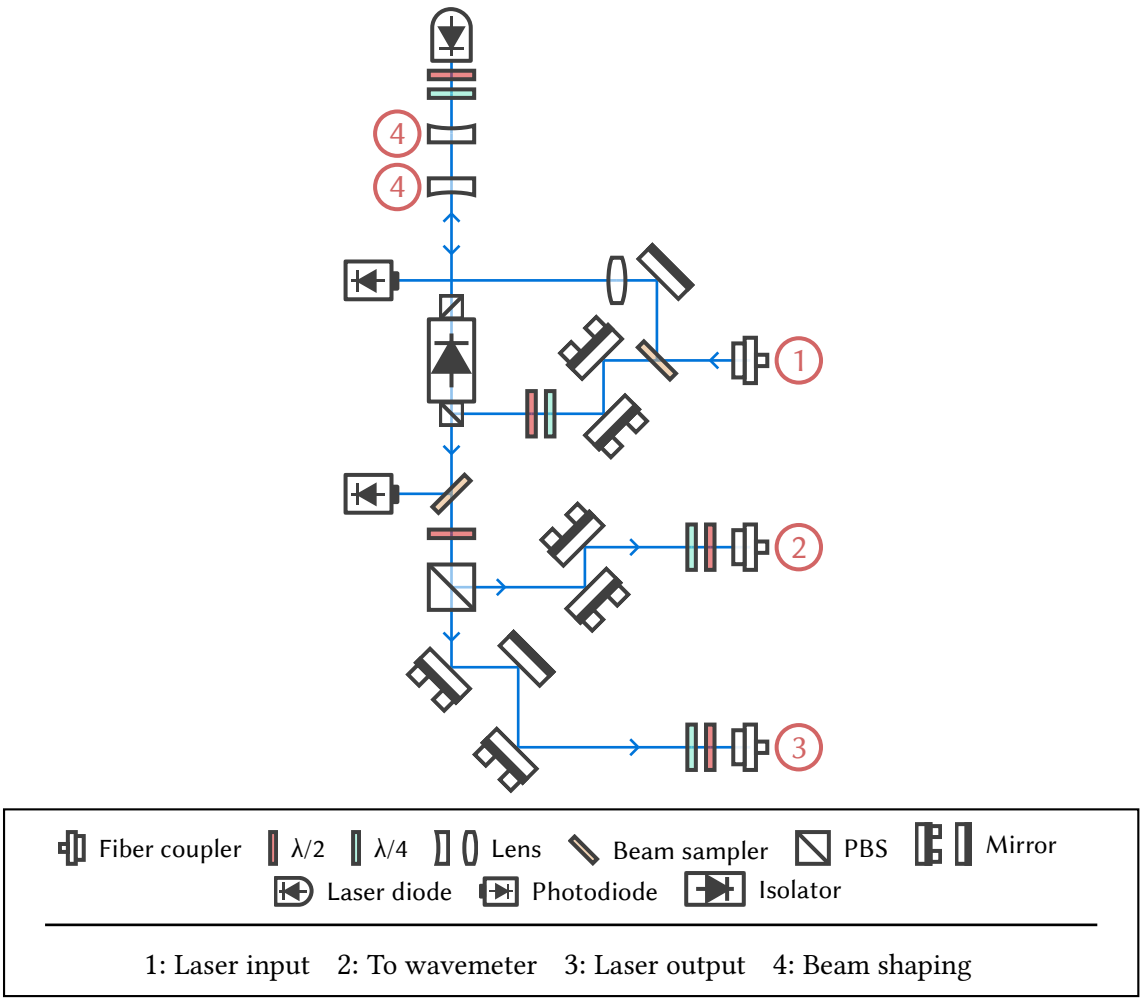


Figure 3.3: **The laser setup that is used for all injection locks.** The seed enters at (1) and is redirected on a PBS of the isolator towards the diode. The beam is shaped (in the case of the blue injection lock, it is sufficient to adjust the vertical axis) to match the mode of the diode. Light that is emitted from the diode has its polarization set to transmit through the previous PBS and is split into a fiber towards the wavemeter and the experiment.

used in an automatic locking scheme, where the current is swept over the bump to detect it and set the lockpoint accordingly. It has been previously accomplished in [90], and the automatic relocking scheme follows the publication. We use an Arduino Nano Matter, for the fact that it has four digital analog converters (DACs), however, as it later turned out, the DACs cannot be sufficiently decoupled, so after all, we can only use two channels on the microcontroller.

The microcontroller is used in the following way: DAC0 and DAC2 are the two channels we can use to connect to the current modulation of the diode's current controller, which

is a Thorlabs PRO8000. A2 and A3 are used to read out the photodiode. We solder a 220 nF decoupling capacitor between DACx and GND. To protect the DAC, we insert a 470 Ω between DACx and the SMA cable connected to the controller. We only need a 200 mV scan range to scan over the lock point, so we scan from 0.0 V to 0.2 V. Initially, we set the voltage to 0.1 V and lock the ILA, to center the lockpoint. When sweeping the voltage, we have a 10us delay between setting the value and reading the photodiode voltage. The minimum can be detected by taking a moving variance - when the variance is maximal, we are at the lockpoint. Then we follow the sequence as described in [90]:

- Sweep over the lockpoint 30 times, from high to low voltage. This thermalizes the position of the lock point
- Sweep from low to high voltage over the lockpoint
- Sweep from high to low voltage over the lockpoint and remember the position
- Sweep from low to high voltage over the lockpoint
- Sweep from high voltage to the lockpoint

The sequence itself takes less than 40 ms. At this point, the ILA is locked. However, since there is a delay between setting the value on the current controller and obtaining the reading on the photodiode, we can feed-forward the compensation voltage. This results in some setup-dependent offset that needs to be considered for each ILA. In [90], the ILA is locked at the beginning of each sequence; however, it can also be detected whether it is currently locked. For this, we take note of the photodiode voltage when the laser is locked and unlocked. If the photodiode voltage exceeds some fraction of the difference of these voltages, we can consider the laser unlocked and relock it according to the above sequence. The complete code of the Arduino is given in Appendix B..

3.4. Toptica rack

The injection locks need to be seeded by a master laser. This beam needs to be frequency locked and ideally provide enough power such that the injection locks remain stable. Diode lasers are a good choice when phase noise is not exceptionally crucial, so we decided on lasers from Toptica. Due to its compact form factor, the laser rack T-RACK is a good choice for us. We have two racks, containing all necessary wavelengths for the initial cooling phases (apart from the 689 nm laser of the $^1S_0 \rightarrow {}^3P_1$ transition, which needs to be narrow for future applications), as well as two HighFinesse wavemeters. A WS8-2 VIS is used for locking, and a WS7 NIR for monitoring laser beams. Therefore, all the main lasers and the locking are self-contained, making them easy to monitor and maintain.

Quantum computers are ideally constantly accessible, so redundancy is a concern so that in case of a failure, parts can be quickly swapped out. Due to that, the rack contains two instances of the 461 nm beam and a 813 nm laser with a tapered amplifier, used as a backup in case the primary 813 nm tweezer light fails (generated via titanium-sapphire laser (TiSa)). Lasers to be used as repumpers on the 448 nm, 679 nm, and 707 nm lines are included, as per Figure 1, and a 408 nm laser for ionizing Rydberg atoms.

3.5. Menlo rack

As a quantum computer, this project aims to achieve high fidelities in both the single-qubit and multi-qubit cases. Single-qubit gates are realized on the $^3P_0$ clock transition, but it is also possible to use $^3P_1$ for selectively applying single-qubit gates, by using it to apply a light shift on the $^1S_0$ state. As such, all noise sources of lasers on the transitions $^1S_0 \rightarrow {}^3P_{0,1}$ reduce the maximally achievable fidelity. The same is true for the multi-qubit gate transition $^3P_0 \rightarrow 60^3S_0$, where a UV laser drives the transition. In this case, the frequency is generated by two seed lasers at 1061 nm and 1570 nm, so errors on the seeds propagate to the resulting UV laser.

To reduce frequency noise, lasers can be mode-locked to a ultra-low expansion (ULE) cavity in a vacuum. Locking many lasers to the same cavity is possible using multi-bore ULE cavities. However, if high fidelity locks are desired, another option is to use a frequency comb, which involves a single seed laser locked to a ULE cavity. We use a set of three racks by Menlo, which contains locking on a frequency of the two red lasers at 689 nm and 698 nm, as well as the two IR lasers at 1061 nm and 1570 nm.

The first rack is a Menlo ORS system, containing a 1550 nm laser locked to a ULE cavity. The temperature is stabilized by a Pound-Drever-Hall (PDH) lock, and an oscilloscope is available for monitoring the lock. A second rack contains diode lasers for the 689 nm lasers and, since space was available, also for the repumpers at 679 nm and 707 nm. The lasers for the Rydberg and clock transitions are sourced externally. The frequency comb, as well as the fiber noise cancellation setups for the 1550 nm ULE seed. Space is available to mount the UV seed lasers. All necessary electronics for driving the system are contained in a third rack, along with the control computer that interfaces with the optical hardware.

Naturally, the ULE cavity can drift on the scale of Hz/day, which is compensated by a PID feedback loop to an AOM. Moreover, the frequency is logged digitally and fed back to the control system to compensate for the frequency on the lasers locked to the comb.

3.6. Blue laser preparation

3.6.1. Distributing the seed

In the case of the blue light, a Toptica DL-Pro is used as a seed laser with an output power of 150 mW out of the fiber. The seed laser is sent to a distribution board in the rack, which splits the light into three ILAs using a Nichia NDB4916 laser diode with a maximum output power of 600 mW. The purpose of the three ILAs is as follows:

Identifier	Purpose	Measured output power ex fiber (mW)
ILA module 1	3D MOT	329
ILA module 2	Zeeman slower	284
ILA module 3	2D MOT + imaging	282

The distribution setup is depicted in Figure 3.4. The seed's polarization is cleaned, and the output is monitored on a photodiode. In this instance, the monitoring allows tracking the degradation of the laser diode and fiber coupling of the DL-Pro (although it is worth noting that the couplings were relatively stable over the course of 1 year). The seed is then split three times using PBSs, creating a total of four outputs. The beam is then split into the three injection locks.

The outputs of the ILAs are entered into a separate breadboard, where the light is shifted in frequency to accommodate the different purposes. A shutter is installed in the beam path, and the light is split into multiple beams (if necessary). Even though all ILA outputs enter the same breadboard, the setups are self-contained and therefore drawn separately below.

3.6.2. 3D-MOT laser stage

The light for the 3D MOT enters diagonally into the glass cell on the xz-plane. As the glass cell is not coated, at an angle of 65°, the transmission into the vacuum is only 70% (see Figure 3.5), and therefore, through both surfaces, it is 49%. Even though it is still possible to create a 3D MOT by retro-reflecting this lossy beam, it provides less control over the alignment, which is crucial when it needs to overlap with the optical tweezers.

Therefore, instead of using retro-reflected beams, six separate beams enter the glass cell. The injection lock with the highest output, ILA module 1, was chosen as the input for

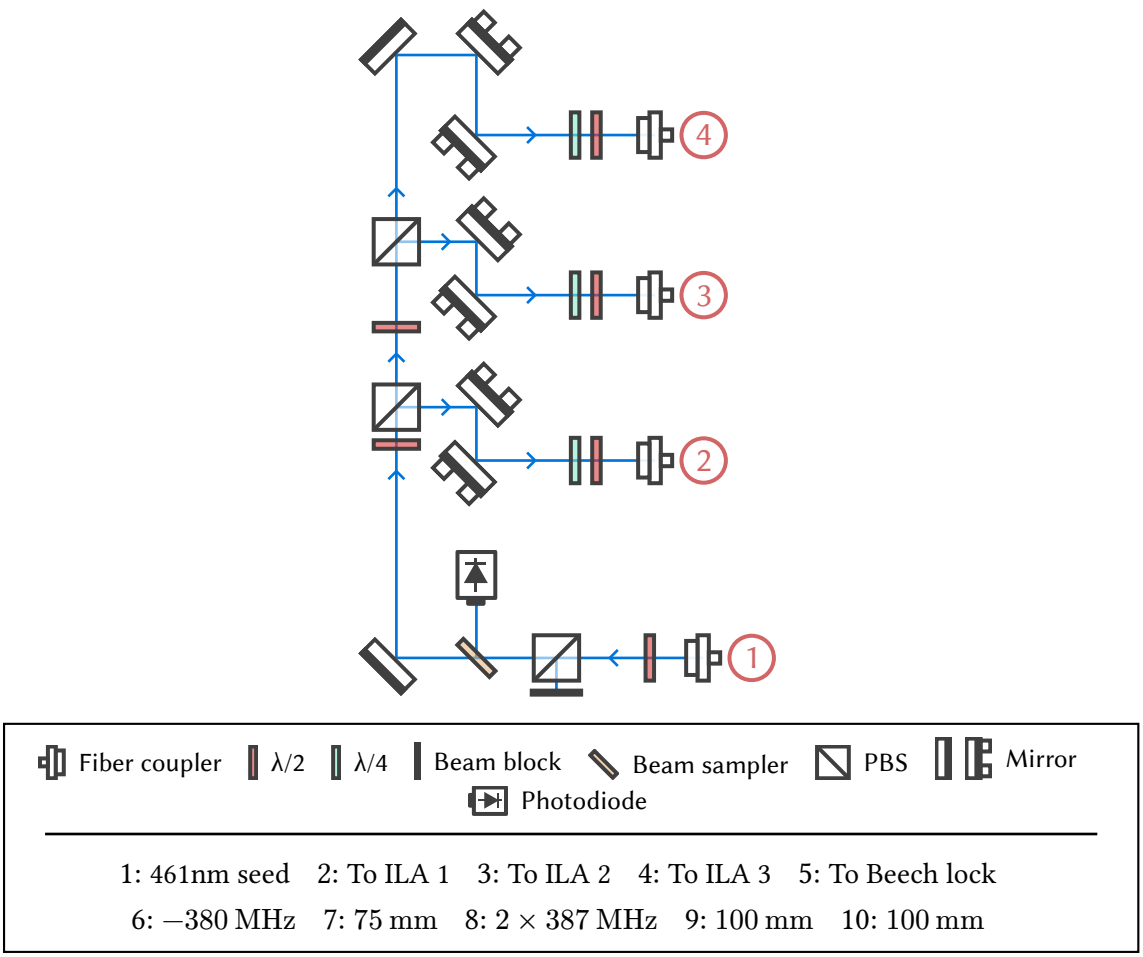
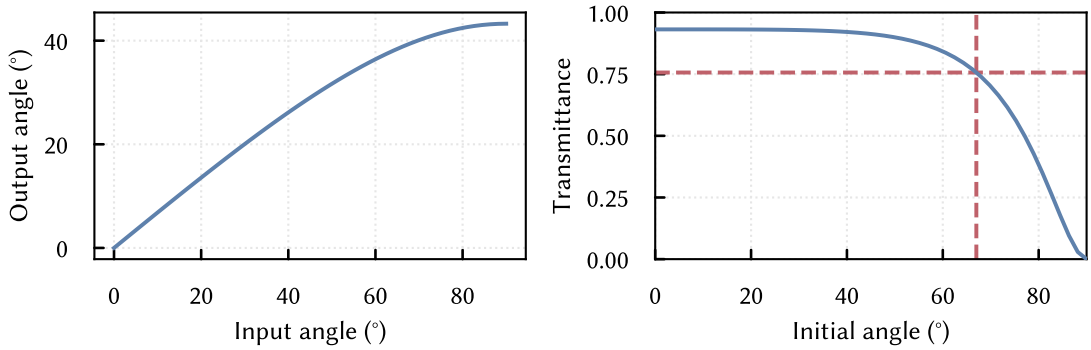


Figure 3.4: **Distribution of the blue seed.** The seed laser of the 461 nm light is split into three identical injection lock amplifier setups. The laser can be locked either via the wavemeter, where the light is directly split from the DL-Pro, or using an AQT Beech system, where the light is split off here. Ideally, the power is distributed evenly across the three ILAs, where above a certain threshold, the lock does not depend on input power anymore. However, as of recently, the imaging ILA was replaced by a Toptica system, the Zeeman slower ILA is not used as the Zeeman slower window of the AOSense stage is coated and does not transmit enough power anymore.

this setup. Figure 3.6 illustrates the setup of the distribution board for creating the six beams. The input first enters a double-pass AOM and is then split into six beams, with opposing pairs sharing one beam splitter, as the intensities of opposing beams need to be balanced. There were plans to include piezo-mirrors into one arm of the beams, since perfectly in-phase opposing beams can create interference fringes [73,92,93]. However, in this setup, the fringes were never observed. The mirrors are still on the breadboard as placeholders, but they may not be necessary in a future iteration.



(a) Refraction of fused silica (b) Total transmittance through glass cell

Figure 3.5: **Refraction and transmittance of light between 460 nm-698 nm via the glass cell.** The refraction of fused silica in this wavelength range is known [91], and using Fresnel equations, the transmittance is calculated. The vertical line in the second plot shows the angle of the diagonal MOT beam in the xz-plane.

One of the beams is additionally overlapped with a repumper at 448 nm, using a PBS. After the cube, the polarizations of the two beams are orthogonal, which can be matched by using a waveplate that is $\lambda/2$ for 448 nm and λ for 461 nm. Consequently, these waveplates are then used to match the fiber polarization to minimize temperature-dependent polarization drifts (which are converted to intensity drifts by placing a PBS after the fiber).

3.6.3. 2D-MOT, Imaging and Zeeman slower paths

The atoms are first cooled using a 2D-MOT setup and a Zeeman slower in the oven section of the vacuum. This process requires two additional beams of different frequencies and a final setup to image the atoms. To keep all options open, the imaging beam has three outputs, which can be used for absorption imaging of the MOT in all three axes. The output of ILA module 2 is used for the Zeeman slower, the setup is shown in Figure 3.7, where the beam is frequency shifted on a double-pass AOM. As in the 3D-MOT case, the beam is overlapped with the 448 nm repumper.

The ILA module 3 output is then split into the 2D-MOT setup, which contains a single-pass AOM, and the imaging setup, which has a double-pass AOM, as shown in Figure 3.8. For imaging, the beam is split on PBSs to the three outputs.

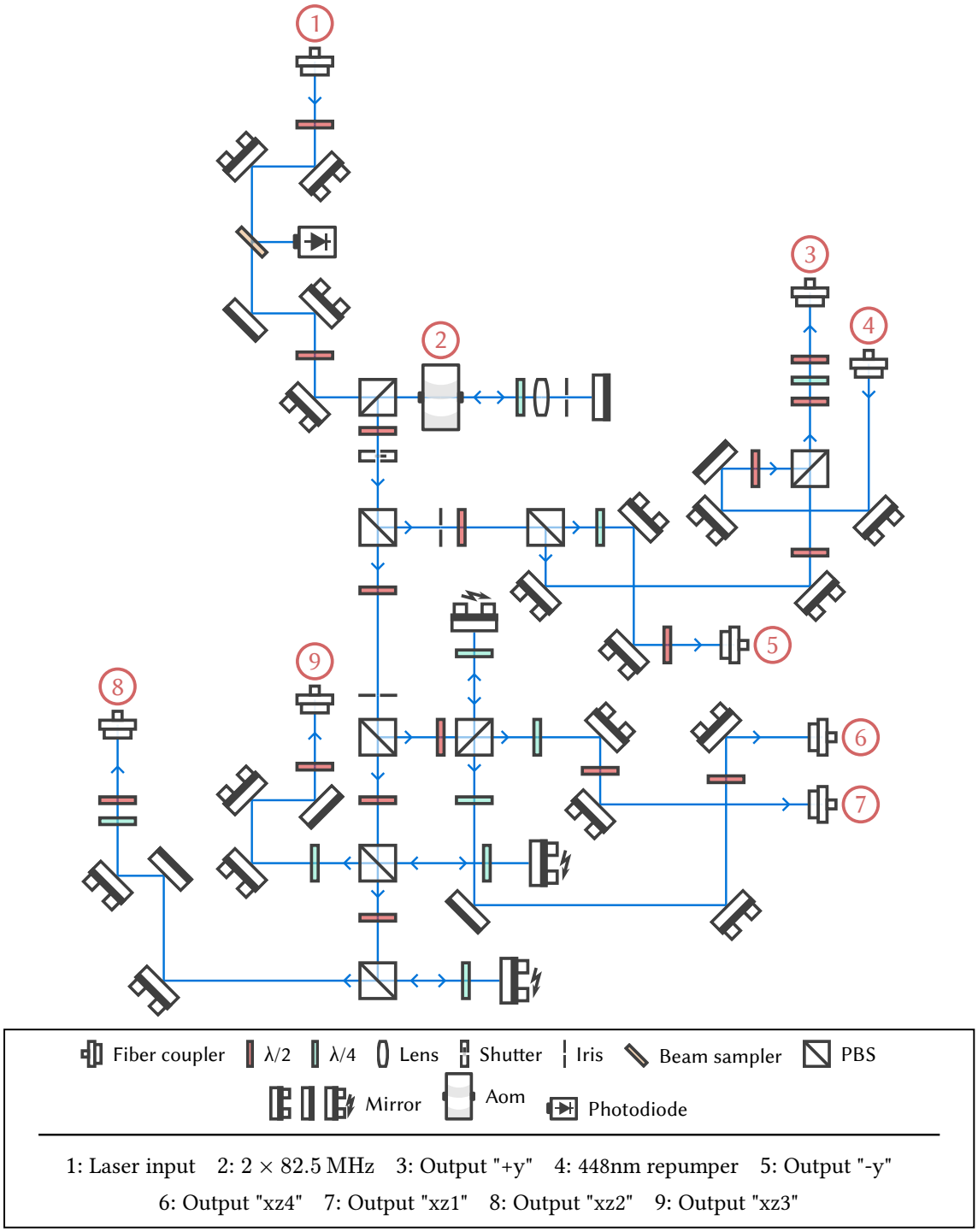


Figure 3.6: **Blue MOT beams preparation.** This setup aims to provide six beams that are fiber-coupled towards the experiment. The initial seed light is frequency shifted via a double-pass AOM and then split into the six components. Opposing beams on the experiment table always share a PBS, allowing for power balancing relative to each other. Piezo mirrors are planned to counter interference fringes, but they have not been used yet. When we first align the MOT, the ratio of power between a diagonal and a horizontal beam is 7/5. This ratio changes during optimization of the MOT signal and is not consistent between optimization runs.

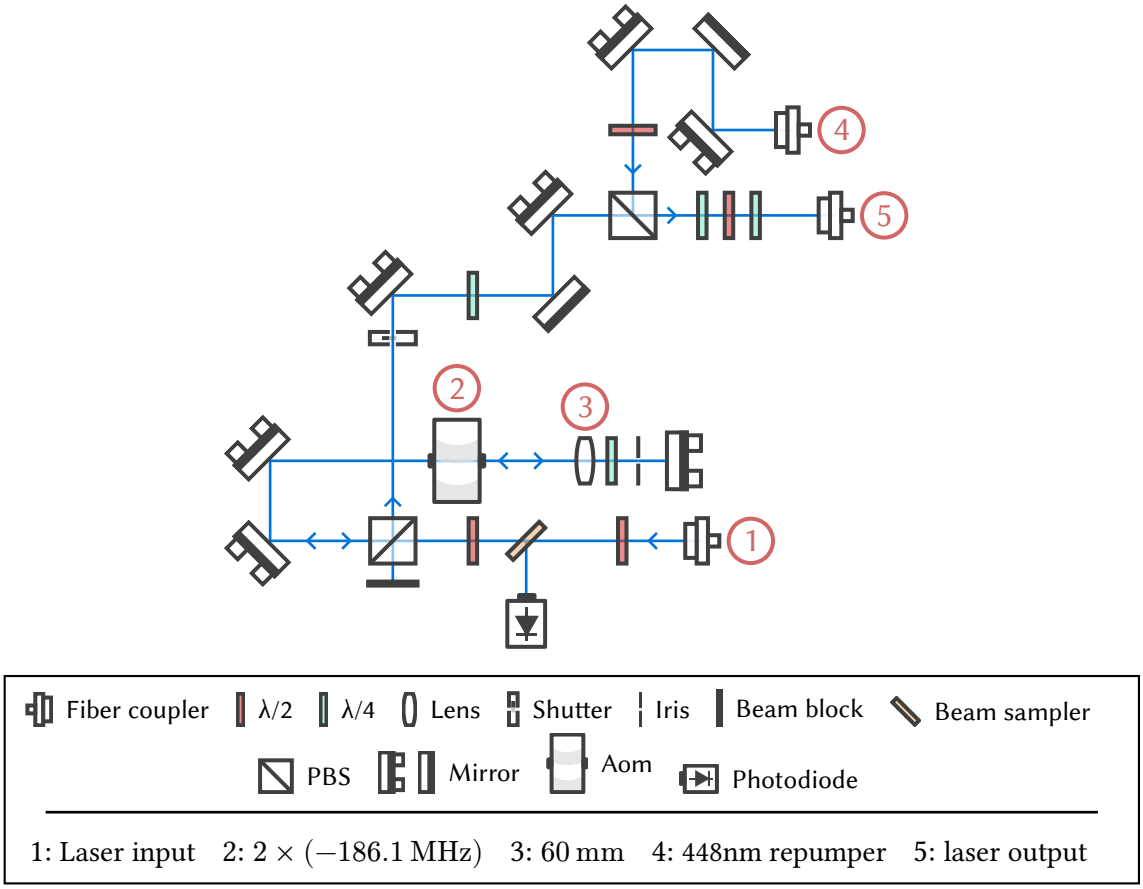
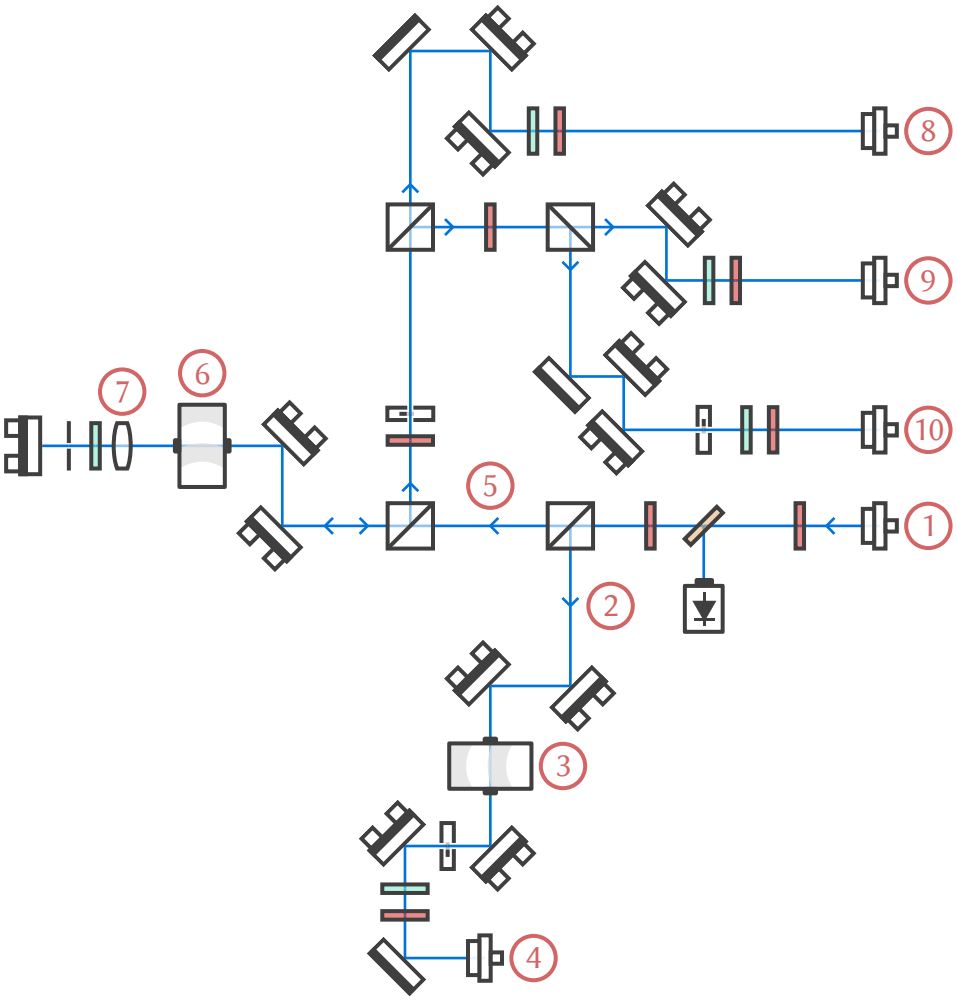


Figure 3.7: **Zeeman slower beam.** The setup for the Zeeman slower has a double-pass AOM at $2 \times (-186.1\,\mathrm{MHz})$ and exits to a fiber towards the experiment. The 448 nm repumper is added to the light using magic waveplates, which allows the overlap of both polarizations.

3.7. Red laser preparation

Following the blue MOT stage, atoms are further cooled on the $5s^1S_0 \rightarrow 5p^3P_1$ transition at a wavelength of 689.45 nm. The transition is used in a 3D-MOT, Sisyphus cooling, and sideband cooling (SBC).

The same injection lock module is used as in the blue case; the laser diode here is Ushio HL67221DG with a maximal power of 200 mW. An Optoquest diode laser is used as a seed, which is locked to a Menlo comb Section 3.5. Two modules are available, which are used as:



Fiber coupler λ/2 λ/4 Lens Shutter Iris Beam sampler PBS
Mirror Aom Photodiode

1: Laser input
2: 2D MOT path 3: 194.2 MHz 4: 2D MOT output 5: Imaging path 6: 2 × 70 MHz
7: 60 mm 8: Imaging 1 output 9: Imaging 2 output 10: Imaging 3 output

Figure 3.8: **2D-MOT beam.** This setup is a combination of imaging beams and the 2D MOT beam. The output of the injection lock is split towards each section. The 2d MOT beam is shifted using a single-pass AOM. The beam for the imaging light is shifted using a double-pass AOM and then split into three fibers, allowing for imaging from three cardinal directions.

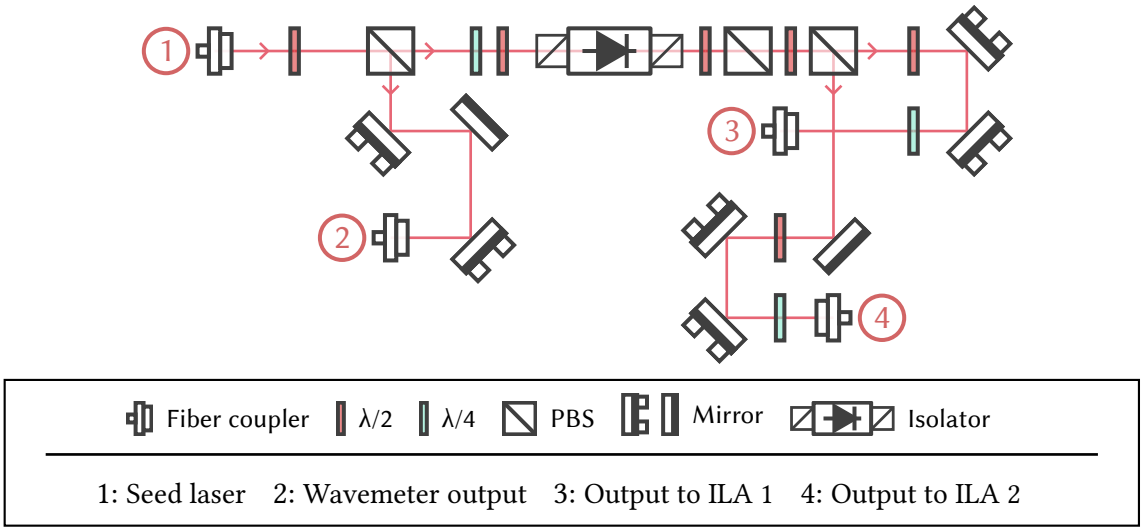


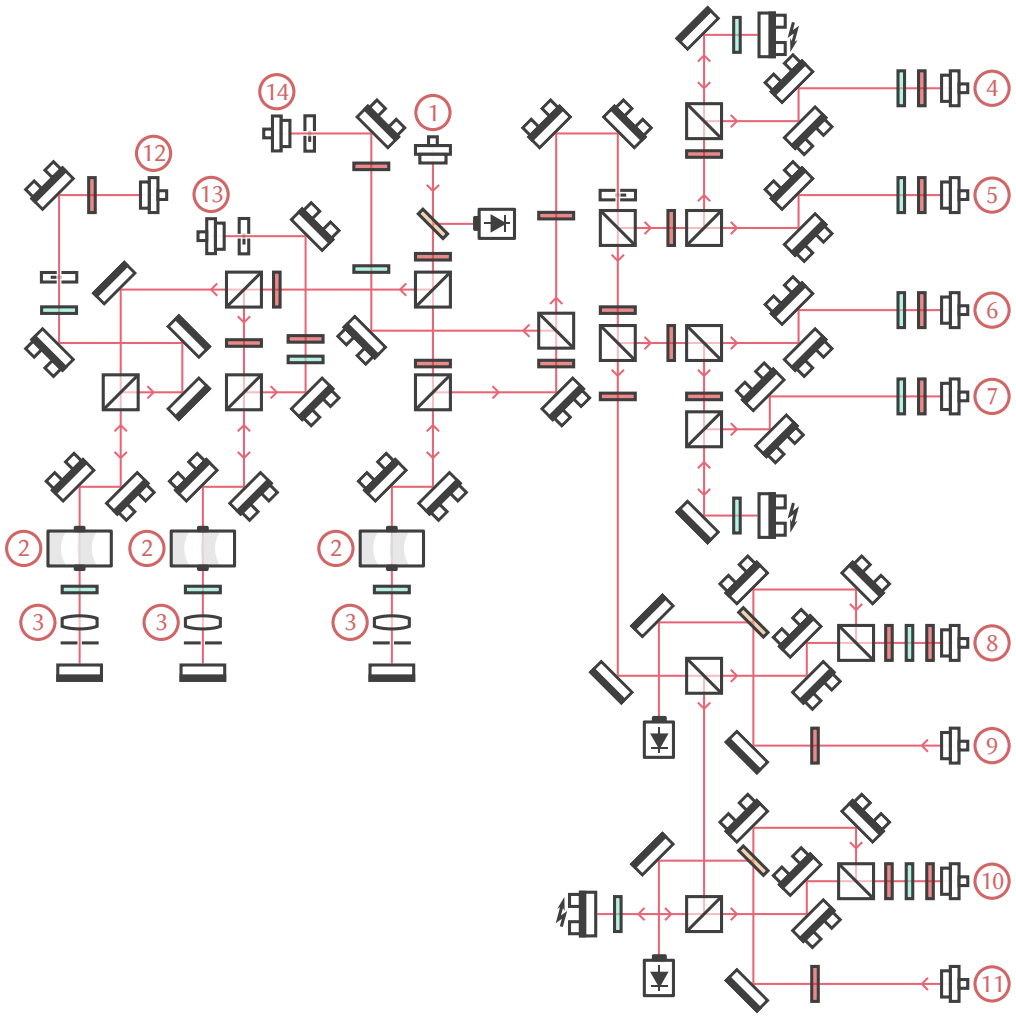
Figure 3.9: **Red MOT seed distribution.** The seed light is split into two outputs for the ILAs, and another monitoring output to the wavemeter (the red light is locked via the Menlo comb).

Identifier	Purpose	Output power ex fiber (mW)
ILA module 1	Single-qubit addressing	130-150
ILA module 2	Red MOT + sideband cooling	138

The addressing is part of the future projects of Zhao Zhang [94]. The initial seed laser is split into two ILA modules, as shown in Figure 3.9. The output of ILA module 2 is sent to a separate breadboard and is split into three paths, as shown in Figure 3.10.

The first and second paths are for sub-Doppler cooling beams 1 and 2, and each contains a double-pass AOM. The beams are coupled into a fiber after traversing a shutter. On the third path, the laser is frequency shifted by a double pass AOM, which is used for both the third sub-Doppler beam and the red 3D MOT. The beam is split into the respective setups right after. In the case of the third sub-Doppler beam, it goes through a shutter and is then coupled into a fiber. For the red 3D-MOT, the beam is split into six beams, and the setup is similar to that of the blue 3D-MOT. The outputs "+y" and "-y" are combined using dichroic waveplates with the 707 nm and 679 nm repumpers, respectively.

A trace of the motional states on the $^1S_0 \rightarrow {}^3P_1$ transition reveals the resonance Figure 5.11 frequency, where the double-pass AOM frequency of the sideband cooling beam is 188.274 MHz. These shifts place the base level of the laser at 434.829121308 THz, resulting in Figure 3.11.



Fiber coupler λ/2 λ/4 Lens Shutter Iris Beam sampler PBS
Mirror Aom Photodiode

1: ILA 2 output 2: center frequency at 2 × 200 MHz 3: 50mm
4: Output "xz2" 5: Output "xz3" 6: Output "xz4" 7: Output "xz1" 8: Output "+y"
9: Repumper 707nm 10: Output "-y" 11: Repumper 679nm 12: SBC 1 output
13: SBC 2 output 14: SBC 3 output

Figure 3.10: **Red MOT and SBC beams preparation.** The distribution setup of the red light for the red MOT is practically the same as for the blue MOT, however, it contains additional AOMs and outputs for sub-Doppler cooling beams (such as Sisyphus cooling and sideband cooling). The main laser is split into six beams for the experiment, and the repumpers at 679 nm and 707 nm are added to two outputs. All AOMs are used for cooling, so that different frequencies in three directions are possible (which is important for sideband cooling). Additionally, the rightmost AOM is also used for the 3D-MOT.

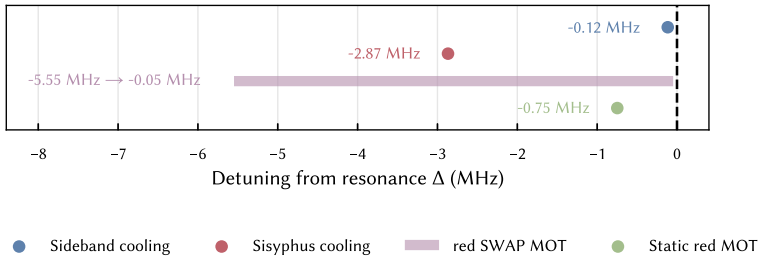


Figure 3.11: **Red laser frequency detunings.** The red lasers are shifted according to this plot, where the resonance was inferred from Figure 5.11.

3.8. Outputs to the experiment

On the experiment side, the 2D-MOT and Zeeman slower beam are shaped and split on a breadboard next to the AOSense chamber, as per the manual, and are provided again in Table 3.1. The schematics are given in Figure 3.12.

Device	$1/e^2$ beam diameter	Detuning from $^1S_0 \rightarrow {}^1P_1$	Laser Power
Zeeman slower	6-8 mm x 6-8 mm	−580 MHz	>50 mW
2D MOTs	25 mm x 5 mm	−40 MHz	2:1 ratio, 25 mW total (1st:2nd trap)
2D MOTs	25 mm x 5 mm	−40 MHz	2:1 ratio, 25 mW total (1st:2nd trap)

Table 3.1: **AOSense beam specifications.** Beam specifications necessary to use the AOSense oven, Zeeman slower, and 2D MOT section. The 2D MOT has two beams of equal size, but different powers. Polarization is right-handed circular for all beams.

The Zeeman slower exits the fiber collimator with a diameter of 5.9 mm, is then polarization cleaned and monitored, and then directly coupled into the AOSense chamber. In the case of the 2D-MOT beam, it exits the fiber with a diameter of 4.4 mm, and is then polarization cleaned. The x-axis extended to a diameter of 24 mm by a telescope of a 30 mm plano-concave lens and a 150 mm plano-convex lens.

On the side of the glass cell, the blue and red beams are polarization cleaned, monitored, and then combined using a dichroic mirror, which is reflective for red and transmissive for blue. The setup is the same for all outputs; however, in the horizontal case (outputs “+y” and “-y”), it rests on a breadboard positioned around the chamber. This is shown in

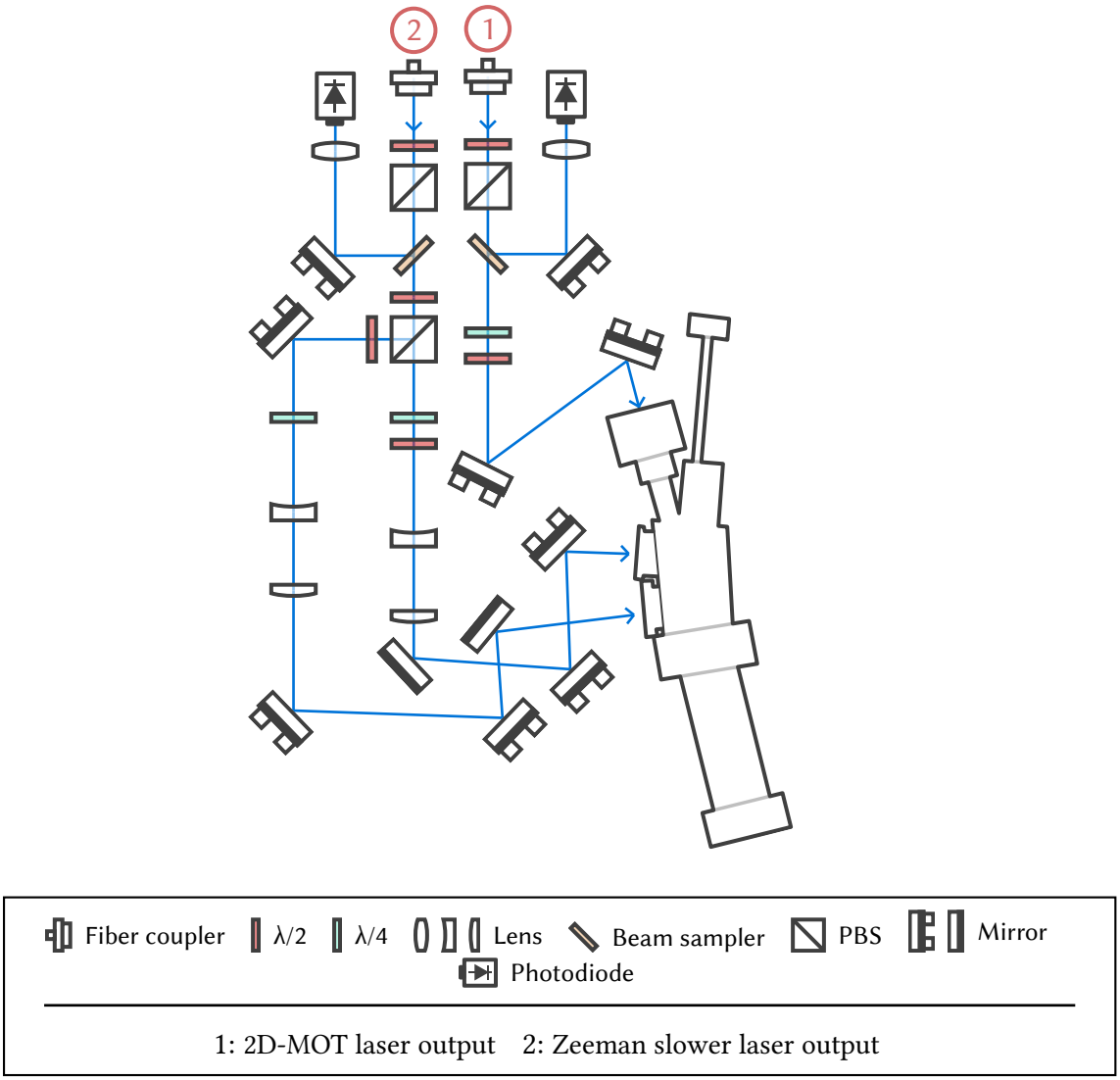


Figure 3.12: The beams for the 2D-MOT and Zeeman slower light are shaped according to the specifications in Table 3.1. The mirrors are carefully placed such that the beams enter the setup perpendicular to the viewports.

Figure 3.13. For the outputs in the xz-plane, the beams are combined in a custom MOT-launcher Section 2.10.

The other lasers around the glass cell are arranged as shown in Figure 3.14. All beams are launched from fiber couplers, polarization cleaned, and monitored. Two mirrors before the glass cell are used to align the beams, and waveplates are inserted to set the correct polarization. The imaging laser is used for absorption imaging, where the beam illuminates the MOT cloud, and the camera collects photons on the other side of the glass cell. It is used for fluorescence imaging, where the photons are collected through an objective.

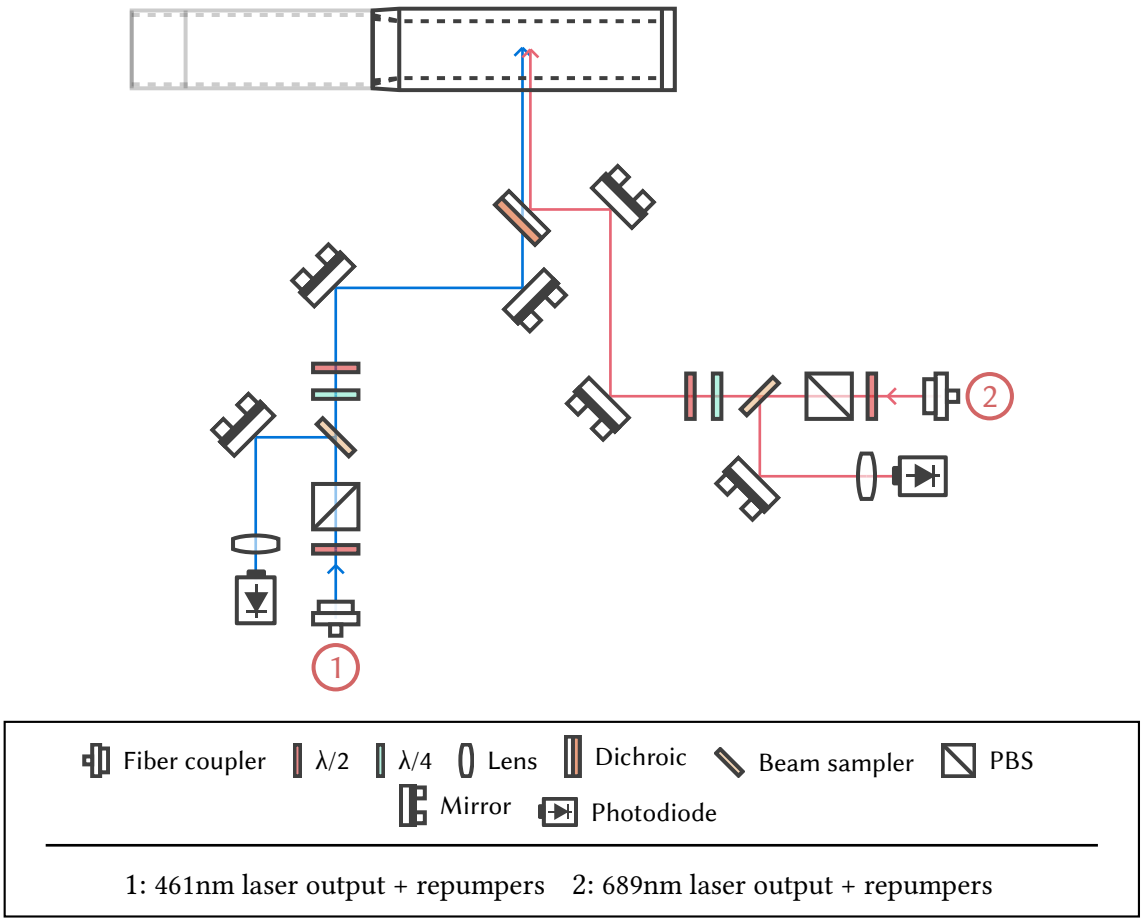
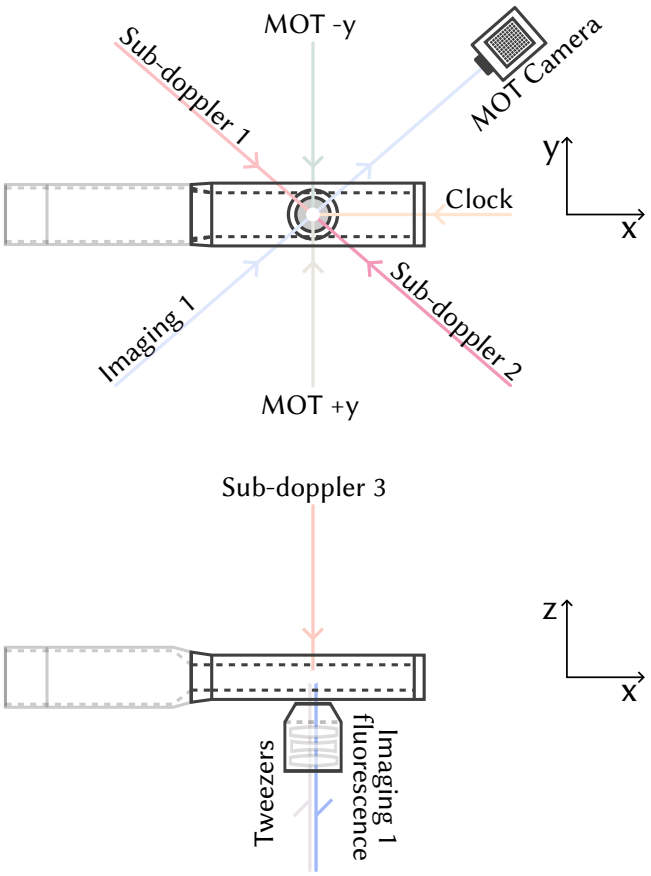


Figure 3.13: **Combined MOT beams.** The lasers for the blue and red MOT are combined using a dichroic mirror before the glass cell.

3.9. Imaging single atoms

All quantum circuits eventually conclude by measuring all or specific qubits. In optical tweezers, we can measure the quantum states simply by taking an image. In the most basic case, atoms are in the ground state, and since the $^1S_0 \rightarrow {}^1P_1$ has a high scattering rate, absorbed photons are quickly rereleased due to spontaneous emission. In fact, every experiment starts by taking an initial image of the ground state population to establish a ground truth for comparison in the next step. The first image is necessary in experiments where 100% filling cannot be guaranteed; however, one can imagine a situation where the atom loss from the imaging process outweighs the stochastic probability of defects in the array, in which case taking a first image might be obsolete.

Imaging states other than the ground state usually require a method on a case-by-case basis. In this experiment, we are especially interested in measuring the $^3P_0$ (clock) state,



Identifier	Direction
MOT +y	+y
MOT -y	-y
Sub doppler 1	+x and -y
Sub doppler 2	-x and +y
Sub doppler 3	-z
Imaging 1	+x and +y
Clock	-x
Tweezers	+z

Figure 3.14: **The glass cell viewed from the top and the side.** The beam used to image the MOT clouds is angled with respect to the glass cell, since the MOT lasers come in perpendicular. A camera is used for absorption imaging with the imaging beam. An objective is below the glass cell, where tweezers, light enters, and fluorescent photons from the atoms exit toward an Orca camera.

as well as the Rydberg state 5s ns^{3}S$_1$. In either case, a destructive measurement removes all ground-state atoms, then transfers the remaining atoms back to the ground state and images them. Rydberg atoms have a high principal quantum number, allowing electrons to be excited even further across the ionization threshold. This process removes Rydberg atoms, enabling the measurement of the remaining ground state atoms. Non-destructive measurements can be realized by shelving the atoms of interest into a quantum state, which is not part of the imaging cycle. Naturally this is not the case for the clock state, as atoms can decay into that state if they decay via the $^1P_1 \rightarrow {}^1D_2$ channel, however atoms on that path can be intercepted via repumper light at 448 nm, 533 nm or 717 nm, see Figure 1 and [95].

Fluorescent photons are emitted from the atoms and can be collected with an objective focused at the position of the atoms. After the objective, the collimated photon beam is reflected from a dichroic mirror (which is transmissive for the tweezer light, which also has to pass through the objective). Two mirrors align the beam into a focusing lens, and another pair of mirrors centers the image onto the camera. The full beam path is displayed in Figure 3.15 The focusing lens and the objective form a telescope, so a suitable focal length must be chosen, which in turn affects the length of the imaging path. This experiment uses a lens with a focal length of 750 mm, that images onto an Orca Quest camera, resulting in a total path length from the atoms to the camera of ~1.55 m.

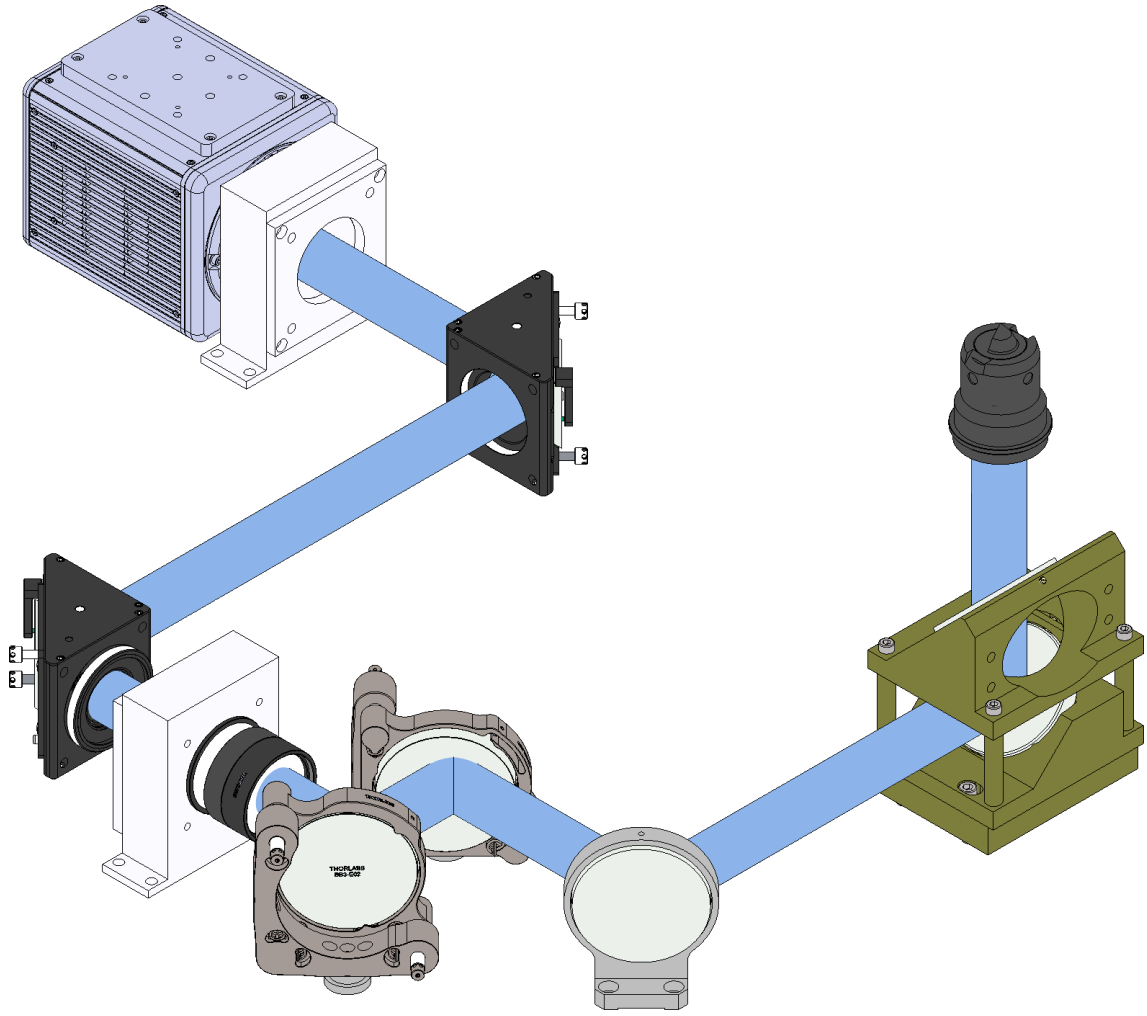


Figure 3.15: **Imaging setup.** The imaging beam passes through the objective, then a beam sampler, and is then reflected off a dichroic mirror towards the camera.

4. Generating and optimizing optical tweezers

Trapping single atoms in individual potentials is one of the main features separating the neutral atom platform from other quantum computing architectures [14,35,36]. The potentials generated by laser beams (either as a lattice or optical tweezers) are highly dynamic, such that arbitrary geometries, arbitrary trap depths, and mid-circuit transport are all possible. Here, we use optical tweezers with an option of integrating a lattice in the future.

The tweezers have a depth of 700 µK, and as the atoms are cooled to the Doppler temperature of 179 nK [70] after the red MOT, each tweezers can contain many atoms. In Figure 5.3 0.5×10^6 atoms are estimatead to be in the red MOT, which leaves 0.34×10^6 atoms in the 1σ radius. So a very rough estimate is dividing the number of atoms in the 1σ radius by the number of tweezers (100), leaving > 3000 atoms per tweezer, keeping in mind that as the atoms move, they are attracted to the tweezers. Loading atoms into optical traps will initially result in many atoms occupying the same trap. Molecular transitions are used to heat out pairs of atoms, resulting in a ~50% probability of retaining a single atom [96]. To create a quantum computer, however, it is necessary to fill every trap within a given geometry. This can be achieved by defining a region of the laser array as the target and the remaining traps as the reservoir. The target region is then filled with atoms from the reservoir, either by moving the traps themselves or by using an additional laser beam to pick up atoms. The latter configuration is the most common, where the pattern is generated by an SLM and atoms are moved around by a 2D-AOD, since it has a real-time response.

The ability to transport atoms is a clear advantage over other quantum computing applications; however, it requires a method to change laser positions quickly, which is not yet possible with only an SLM. Thus, we choose to implement tweezers with AODs. Only in later stages of the experiment, an SLM is used in combination with an AOD. The setup is explained below, including details about the shuffling algorithm and pitfalls found during the experiment. Generation of SLM arrays is not discussed here, but will be part of a future thesis, and references are available [97,98].

4.1. Splitting, projecting, and focusing

Optical dipole traps are generated by focused Gaussian beams and trap atoms by means of the Stark effect. The trap geometry is given by the Gaussian beam geometry defined by the intensity $I(r)$ of the laser beam, as well as the waist w_0 in the radial direction and the Rayleigh range z_R in the axial direction.

When the energy of the atoms, given by their temperature T is much lower than the trap depth, the trap can be approximated by a harmonic potential and atoms can be thought to oscillate in the trap. From [54], the trap depth is inversely proportional to the waist ω_0' of the focused beam, so minimizing it is desirable.

In the approximation of Gaussian beams, it is straightforward to show that a minimal waist after a lens follows from maximizing the radius of the laser before the lens. The lens equation is [99]:

$$\omega_0' = M\omega_0 \tag{4.2}$$

with the Magnification M as:

$$M = \frac{\left|\frac{f}{z_0 - f}\right|}{\sqrt{1 + r^2}} \tag{4.4}$$

Here, ω_0 is the waist of the beam before the lens, ω_0' the waist after the lens, z_0 the Rayleigh range, r the radius of the beam at the position of the lens, and f the focal length of the lens.

The magnification is minimal when either the focal length of the lens is minimal or the radius is maximal. As the system uses a glass cell, the minimally achievable focal length is limited by the distance from the center to the outside of the glass cell. Moreover, the radius of the objective is mechanically limited by the system, as the space around the experiment is usually restricted. Instead of using a single lens to focus the beam, an objective consisting of many lenses can be used to optimize various factors, such as chromatic and spherical aberrations and focal length.

In the following setup, the tweezer beam is expanded to fill the objective. However, care must be taken, as diffraction rings will appear when the beam is too large. As can be seen in Figure 4.1, the tweezer light exits a fiber. It is polarization cleaned, with its intensity monitored and stabilized, and then the beam is coupled into a 2D-AOD setup. We use an AA DTSXY-400-800 in its standard configuration. A lens images both first diffracted orders (the focus is at the center of both AODs) and the beam expands until it reaches the second lens of the telescope. It passes through a dichroic, which transmits the 813nm

light and reflects the 461 nm imaging light, and then enters the objective, where it is focused to the center of the glass cell.

Between the two lenses is a flip-mirror, which is used to view the beam pattern on an auxiliary camera or a photodiode.

The telescope before the AOD expands the beam enough to fit into a 7mm aperture. The beam exits the fiber with a $1/e^2$ diameter of $d_0 = 1.5\,\mathrm{mm}$. The collimation was verified by using an infinity-corrected camera[2]. The lenses have focal lengths of 50 mm and 125 mm, resulting in magnification of $M_1 = \frac{125\,\mathrm{mm}}{50\,\mathrm{mm}} = 2.5$ and a beam diameter before the AOD of $d_1 = M_1 \cdot d_0 = 3.75\,\mathrm{mm}$.

After the telescope, the lenses have focal lengths of 100 mm and 750 mm, resulting in magnification of $M_2 = 7.5$, the beam diameter before the objective is then $d_2 = M_2 \cdot d_1 = 28.125\,\mathrm{mm}$. The objective has an aperture of 31.25 mm, resulting in a power efficiency of $\sim 91\%$ by considering the clipping of a Gaussian beam by the aperture.

Initially, simulations were run to find the optimal parameters for the lenses, and we optimized for high trap frequencies. The highest trap frequencies are reached when the $1/e$ diameter of the beam matches the aperture of the AOD (and therefore apodization $A = 1$). However, this results in significant clipping, which leads to diffraction rings. That configuration was used in the beginning with magnifications $M_1 = 4.17$ and $M_2 = 4.29$. Although we managed to trap atoms, optimizing the homogeneity of the traps was limited. Instead, we opt for an apodization $A < 1$ and choose the magnification such that we make maximal use of the bandwidth of the AOD, assuming a certain grid size. The relation between displacement in the atomic plane Δx and deflection angle $\Delta\theta$ from the AOD is given by

$$\Delta x = \frac{f_{\mathrm{obj}}}{M_2}\Delta\theta \tag{4.6}$$

The objective has an effective focal length of $f_{\mathrm{obj}} = 24.4\,\mathrm{mm}$, so that a tweezer spacing of 3 µm results in a deflection $\Delta\theta = 0.92\,\mathrm{mrad}$. The maximal deflection is given by the manufacturer as $\Delta\theta_{\mathrm{max}} = 44\,\mathrm{mrad}$, so that this will result in a maximal number of tweezers along one axis of $N_{\mathrm{max}} = \Delta\frac{\theta_{\mathrm{max}}}{\Delta}\theta = 47$, which is certainly enough for the first phase of the experiment, however a different set of magnifications M_1 and M_2 may be considered in the future if more tweezers are required.

[2] A infinity corrected camera is assembled in the following way: Take a regular monochrome machine vision camera, add a filter and then a lens, preferably with long focal length (e.g. 250 mm). Try to find an object far away (e.g., by looking out the window) and focus on it by moving the lens. Once that step is complete, the camera can be used to collimate a laser beam, where the only objective is to minimize the spot size on the infinity-corrected camera.

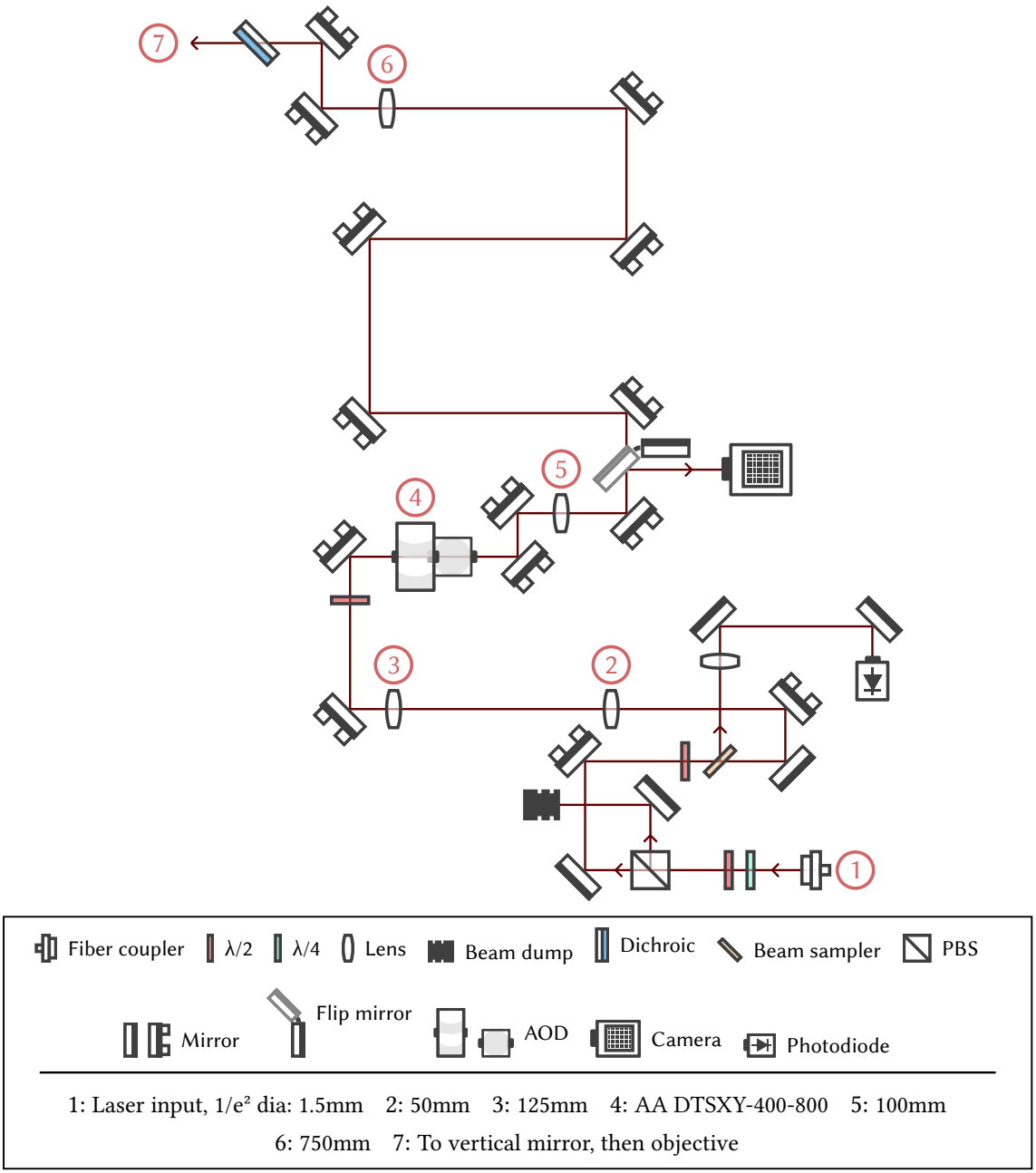


Figure 4.1: **Laser setup of the 813nm tweezer beam.** The laser is first polarization cleaned, and a pick-off plate samples a part of the laser power for monitoring purposes on a photodiode. The beam then enters a telescope to reach the desired apodization on the AOD aperture. The following optics consist of two lenses with focal lengths f_1 and f_2, respectively, and an objective with focal length f_3. The first lens is f_1 away, expanding the beam, which is then recollimated on a second lens at a distance $f_1 + f_2$. The objective is then positioned at $f_2 + f_3$. The collimated beam on the objective needs to be large in order to reach a tight focus in the atomic plane, resulting in a long beam path.

4.2. Acousto-optical deflector as a driver for tweezers

For trapping single, neutral atoms, there are two most prominent choices: either by utilizing an optical lattice [100,101], where spacings can be small and are periodic, or by using dipole traps [26,27], where atoms are constrained in the focus of a single laser beam. To create an array of dipole traps, however, further choices are available, each with its pros and cons. SLMs are often used [31], which are freely configurable holographic devices consisting of pixels made up of liquid crystals. There is a wide range available as they are common in commercial projectors. Similarly, DMDs consists of microscopic, tunable mirrors that can be turned on or off. They deflect parts of a beam, and by choosing the right phase conditions, it can be used to create an array of individual beams [102]. Both devices are a good choice for static patterns; however, for real-time modulation, AODs is currently a better option. There, a single laser beam is deflected using the acousto-optical effect and can be split into many beams [31,39]. The pattern is generated by keeping the RF on constantly on the device, and by modulating the RF, the pattern can be changed as well. Thus, each site has a slightly different frequency, which can become problematic for driving specific transitions.

Still, however, the AOD is a good choice for implementing a simple solution with the intention of movable optical tweezers, which can be used to create a defect-free array, as will be seen in the following chapters.

4.2.1. Applying RF to an AOD

The working principle of an AOD is effectively the same as for AOMs however, AOD typically has crystals with a smaller speed of sound to increase the deflection angle. Moreover, the devices are optimized for maximum diffraction efficiency on the first order, since the intended use is to deflect a beam and not to shift its frequency.

The deflection angle is given by the Bragg condition, which depends on the RF with frequency f applied to the AOD, the wavelength of the light λ, and the speed of sound in the crystal v_s:

$$\theta_{\pm} \simeq \frac{f}{v_s}\lambda \tag{4.8}$$

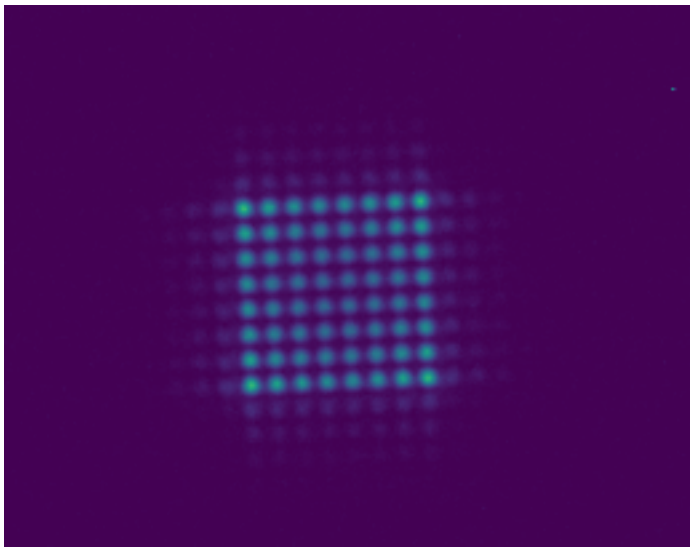


Figure 4.2: **Camera image of intermodulation effects.** The intermodulation effects are visible as the nonlinear transfer function of the AODs mixes the input signals. Constructive interference can be seen outside the desired region, and destructive interference is seen inside the desired region.

To split a laser beam, multiple RFs can be applied; however, intermodulation effects will affect the quality of the output beams [26]. Due to the nonlinear transfer function of the AOD, when two frequencies at f_1 and f_2 are applied, this results in higher-order intermodulation frequencies. The first order frequencies are at $f_1 \pm f_2$. Our AOD has a center frequency of $f_c = 100\,\text{MHz}$ and bandwidth of $\Delta f = 40\,\text{MHz}$. Therefore, these orders are not visible; however, they seed second orders at $2f_1 \pm f_2$ and $f_1 \pm 2f_2$. Due to interference, the resulting pattern has enhanced and diminished spots at the desired frequencies, as in Figure 4.2. The effect becomes more prominent as more tones are added.

The effect can be compensated for by varying the amplitudes and phases of the tones. The algorithm used in this work is described below and adapted from [103].

4.2.2. Homogenizing tweezer amplitudes

The waveform applied to the AOD is a multitone waveform of the following form:

$$\Phi(t) = \sum_i a_i \sin(2\pi f_i t + \varphi_i) \tag{4.10}$$

The amplitudes are given in peak-to-peak voltage V_{pp}. If the RF generator is a DAC, it is desired to make use of the full dynamic range of the device. In that case, the waveform is normalized to the range [1, −1] by dividing by the maximum of the absolute value of the waveform:

$$\Phi_{\mathrm{norm}}(t) = \frac{\Phi(t)}{\max(|\Phi(t)|)} \tag{4.12}$$

On the device, the output amplitude is then set in V_{pp} to $\max(|\Phi(t)|)$. A scaling factor and an offset may have to be applied depending on the supported format of the generator.

It is important to note that dividing by the maximum value is the correct way to do it, and instead setting the individual tones as $a_{i,\mathrm{norm}} = a_i/n$ does not work, because interference will reduce the total amplitude to < 1 and approaches zero for a large number of tones. Moreover, an optimization algorithm optimizing on this approach will fail, because the amplitudes in this case are not decoupled, as the full output amplitude is a non-trivial combination of all amplitudes and then $a_{j\neq i}$ depends on a_i.

Using the correct normalization, homogenization happens in two stages. First, the tweezer pattern is generated and observed on an auxiliary camera without atoms. In the second stage, atoms are loaded into the tweezer, and a suitable sequence runs to get feedback on the homogeneity of the array.

In the first stage, phases are optimized according to [104]. The equation is given as:

$$\theta_n = \theta_1 - 2\pi \sum_{l=1}^{n-1} (n-l)p_l \tag{4.14}$$

where p_l are the relative powers of the tones. This is further reduced when the power spectrum is flat, and thus $p_l = \frac{1}{n}$, then

$$\theta_n = \theta_1 - \frac{\pi n^2}{N} + \frac{\pi n}{N} \tag{4.16}$$

The amplitudes are optimized with feedback from an auxiliary camera by using the algorithm described in [103]. Both axes are optimized at the same time.

Initially, all amplitudes a_{ij} are set to the same value, and such that the total output power is below the damage threshold of the AOD. In this optimization stage, $M_{\mathrm{ij}} = 1$ for all i, j. The waveform with the phases from above is applied, and an image is acquired from the auxiliary camera. The pixels are counted around a region of interest (ROI) of each tweezer spot to set the values of C_{ij}. As in the reference, the balancing errors E_{ij} are then:

$$E_{\mathrm{ij}} = \frac{M_{\mathrm{ij}} C_{\mathrm{ij}} - \langle MC \rangle}{\langle MC \rangle} \tag{4.17}$$

where $\langle MC \rangle = [1/(N_1 N_2)] \sum_{\mathrm{ij}} M_{\mathrm{ij}} C_{\mathrm{ij}}$. This is then converted to 1D errors for the row and column:

$$E_i^{\mathrm{row/column}} = \frac{1}{N_2} \sum_{\mathrm{ij}} E_{\mathrm{ij}} \tag{4.18}$$

and finally, the amplitudes of the tones are then updated according to:

$$A_{\mathrm{new}}^{\mathrm{hor/vert}} = A_{\mathrm{old}}^{\mathrm{hor/vert}} - pE^{\mathrm{row/column}} \tag{4.19}$$

The proportionality factor p is determined to increase convergence speed. Additionally, we update p in each iteration to $p_{\mathrm{new}} = 0.98 p_{\mathrm{old}}$, as less proportionality gain is needed as the solution gets closer to a fully homogenized array.

The phases are updated according to the new amplitudes and the loop repeats. The ending condition of the algorithm is at a maximum of $n_{\mathrm{max\_iter}}$ iterations or when the standard deviation σ_E of E_{ij} falls below a threshold. In the experiment, values of $n_{\mathrm{max\_iter}} = 50$ and $\sigma_E = 0.01$ were sufficient.

For the second stage, we load the atoms into tweezers. As the frequency of the sisyphus cooling feature depends on the trap depth, it can be used to homogenize the array by centering the feature across the array, as was done in Figure 4.3. We scan the frequency of the Sisyphus cooling beam, then record the cooling feature. A Super-Gaussian can fit it, as it is approximately a rising flank for lower frequencies, then a plateau, and a falling flank for higher frequencies. The center point of the fit is then the detuning δ_{ij}, which is used to find the mask M_{ij} by dividing them with their median:

$$M_{\mathrm{ij}} = \frac{\delta_{\mathrm{ij}}}{\mathrm{median}\left(\delta_{\mathrm{ij}}\right)} \tag{4.21}$$

The exact process as in the first stage is repeated on the auxiliary camera, using the new coefficients M_{ij}.

The optimized Sisyphus cooling parameters are then used during all imaging stages and in a separate cooling stage after the first image, before applying sideband cooling. To enhance the Sisyphus cooling feature, we take multiple (up to ten) images in a row.

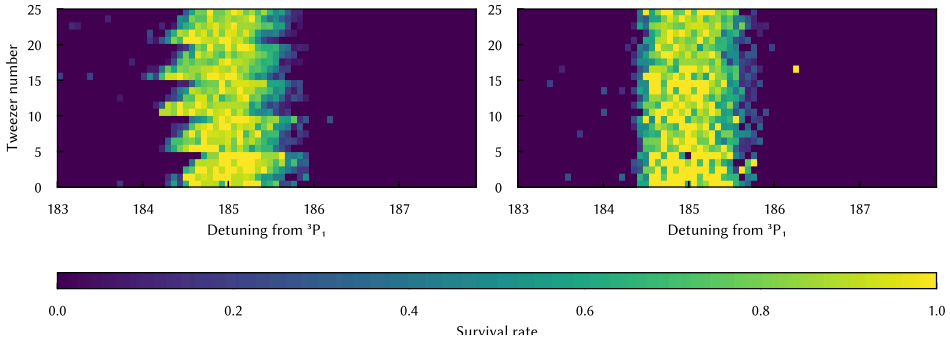


Figure 4.3: **Tweezer homogenization using $^1S_0 \rightarrow {}^3P_1$ transition.** To homogenize the tweezer amplitudes, we use the fact that the $^1S_0 \rightarrow {}^3P_1$ transition sees a light shift. When the tweezer amplitudes are homogenized, then the light shift is the same, as is the case in the second plot. For this measurement, the array was five columns by five rows. The tweezers were counted first from left to right, then top to bottom, which explains the feature in the left figure, showing a column-dependent light shift. The color scale shows the mean survival rate of the atoms in the tweezers. The standard deviation in the relative detuning between tweezers here is $\sigma_{\mathrm{rel,raw}} = 3.94\%$ before optimization and $\sigma_{\mathrm{rel,opt}} = 1.07\%$ after optimization.

4.3. Assuring perfect filling

Atoms trapped in optical tweezers can be moved with high fidelity [38] and are used to achieve a filled tweezer array, which, in turn, determines the available qubits on a quantum computer. Commonly, a static pattern is generated using an SLM or a DMD, and atoms are transferred to dynamic traps, which can be changed in real time, often generated by AODs.

To obtain initial results on the machine's performance, it is not necessary to implement the static tweezer stage immediately. Instead, we opted for using a pair of AODs to generate a 2D tweezer array. It has been shown before [26], that using an AOD for generating a large chain of filled, evenly spaced atoms is possible. In that case, after the single atoms are loaded into the tweezers, an image is taken to establish the occupation of the chain. Empty traps are turned off, and all other traps are moved towards the center until an even spacing is reached.

For the 2D-array generated by AODs, individual control of single sites is not possible. Each trap generated by the first AOD is a seed for more traps in the next dimension

when entering the second AOD. As such, only rows and columns can be controlled individually.

Sorting chains of atoms

The same idea for achieving 100% filling, as in the case of the 1D chain, can still be used. The idea of the algorithm is again to turn off empty tweezers and move them together. However, since only rows and columns can be controlled individually, turning off a tweezer means turning off a full row or column. For the following mathematical notation, N_i is the number of tweezer rows and M_i the number of tweezer columns. We consider an initial configuration of $N_0 \times M_0$ and a target configuration of $N_1 \times M_1$, where $N_1 \gg M_1$. In the case where the initial number of columns and the target number of columns are the same ($M_0 = M_1$), rows that are not fully filled are turned off, and atoms that occupied a trap are lost. Only fully filled rows remain, and they can be pushed together into the target geometry. In cases where there are more initial columns than target columns, the algorithm begins by selecting only the columns that result in the highest number of filled rows. A visual representation of the algorithm is given in Figure 4.4.

Probability of successful sorting

To understand how many tweezers are necessary in the system, the success rate of a sorting event can be calculated. For this, we are interested in finding a target of n rows

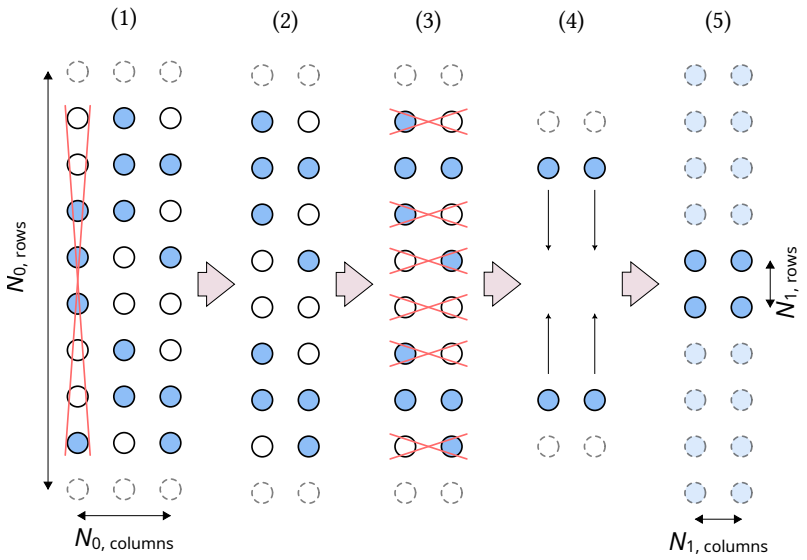


Figure 4.4: **Sorting scheme.** The available degrees of freedom in a pattern generated using an AOD can be used to create a defect-free array. In the first step, the best of two columns is chosen, which contains the most pairs of two atoms in a row. Rows which have defects are eliminated and the resulting traps are moved together.

that are fully filled from an initial distribution of N_0 available rows. This is given by the binomial distribution:

$$B(n \mid p, N_0) = \binom{N_0}{n} p^n (1-p)^{N_0 - n} \tag{4.23}$$

The actual probability is slightly higher because we are interested in *at least* N_1 target rows; hence, the total probability is the cumulative sum over all distributions from N_1 to N_0 rows.

$$B_{\text{cum}}(p, N_0, N_1) = \sum_{k=N_0}^{N_1} B(k \mid p, N_1) \tag{4.25}$$

For a two-column setup, the probability of finding exactly two atoms in a row is $p = \frac{1}{4}$. With the above parameters, this can also be labeled as "the probability of two columns having at least N_1 atoms at the same positions", written as: $P_2 = B_{\text{cum}}(1/4, N_0, N_1)$. With the option of multiple columns, the probability is found by matching two columns out of the initial M_0. The number of possible combinations for two columns out of M_0 is the binomial coefficient $M_{\text{combinations}} = \binom{M_0}{2}$. The resulting problem is finding two columns that have at least N_1 rows of exactly two atoms out of all possible combinations of columns - the disjunctive probability ($P(A \cup B) = P(A) + P(B) - P(A) * P(B)$) of combining pairs of columns:

$$P_{\text{final}} = \bigcup^{M_{\text{combinations}}-1} P_2 \tag{4.27}$$

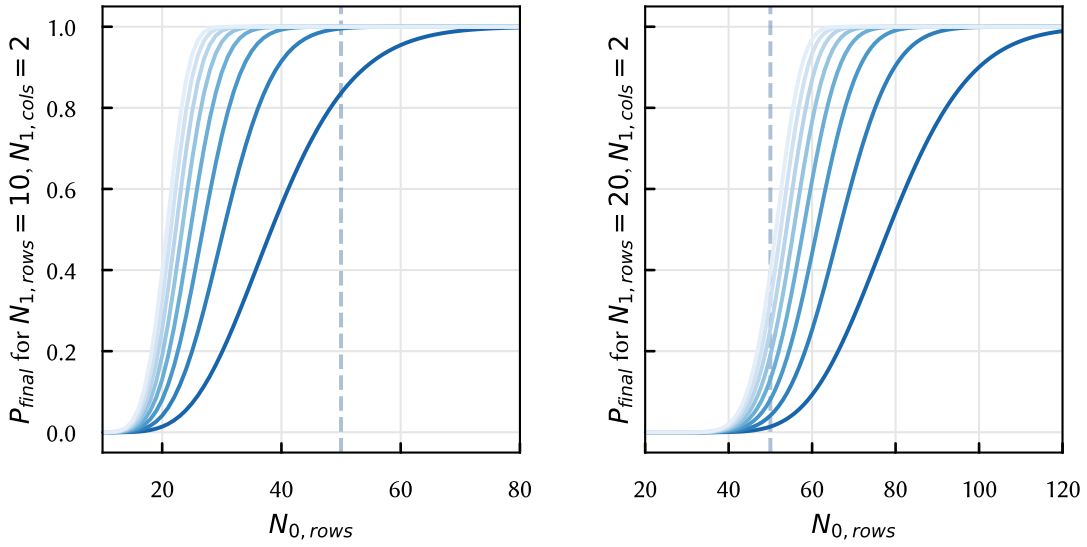
These results are used to understand how many initial tweezers are necessary to reach a target, which in this instance was 2×10 atoms. Examples of the results are plotted in Figure 4.5 and the Python code is given in Appendix C..

Mathematical description of moving atoms via constant jerk

Experimentally, moving atoms means changing the frequency applied to the AOD. A good frequency ramp is found by constraining the acceleration of the atom to a linear profile with the jerk factor j [38]:

$$a(t) = j_0 t + a_0 \tag{4.29}$$

we note that the position of the atom is proportional to the frequency applied: $x(t) \propto f(t)$, therefore, the frequency for the waveform can be found by integrating the acceleration twice.



(a) Initially 10 rows available

(b) Initially 20 rows available

Figure 4.5: **Probability of loading exactly two atoms in $N_1 = 10$ and $N_1 = 20$ rows.** The x-axis shows how many rows are initially available, resulting in the probability on the y-axis. Colors from dark to bright are $M_0 = 2$ to $M_0 = 8$ columns, the vertical line at $N_0 = 50$ is added as a guide to compare the two examples.

$$\begin{aligned} v(t) &= \int_t \mathrm{d}t' a(t') \\ &= \frac{1}{2} j_0 t^2 + a_0 t + v_0 \end{aligned} \tag{4.31}$$

$$\begin{aligned} f(t) &= \int_t \mathrm{d}t' v(t') \\ &= \frac{1}{6} j_0 t^3 + \frac{1}{2} a_0 t^2 + v_0 t + f_0 \end{aligned} \tag{4.33}$$

The frequency ramp will run over a duration T symmetrically, such that $t \in [-T/2, T/2]$. The ramp will start and end with zero velocity of the atom; the starting and ending frequencies are inferred from the camera image. Therefore, the starting conditions are:

$$\begin{aligned} v(-T/2) &= 0 \\ v(T/2) &= 0 \\ f(-T/2) &= f_{\text{start}} \\ f(T/2) &= f_{\text{end}} \end{aligned} \tag{4.35}$$

from which follows, that

$$\begin{aligned} a_0 &= 0 \\ j_0 &= \frac{-12\Delta f}{T^3} \\ v_0 &= -\frac{1}{2} j_0 \left(\frac{T}{2}\right)^2 \\ f_0 &= \frac{f_{\text{start}} + f_{\text{end}}}{2} \end{aligned} \tag{4.37}$$

Finally, the waveform $\Phi(t)$ is calculated by inferring its phase from integrating the frequency:

$$\begin{aligned} \Phi(t) &= \sin(2\pi\varphi(t)) \\ \varphi(t) &= \int_t \mathrm{dt}' f(t') \\ &= \frac{1}{24} j_0 t^4 + \frac{1}{6} a_0 t^3 + \frac{1}{2} v_0 t^2 + f_0 t + \varphi_0 \end{aligned} \tag{4.39}$$

The full waveform consists of a set of three partial waveforms: The static, initial pattern, the sorting ramps, and the static, final pattern. To conserve continuity between these segments, the initial phase is chosen to match both the initial and final phases and is interpolated in between. In our case, the interpolation is cubic, but a linear interpolation likely works as well.

The Python code used to generate the ramps is given in Appendix D.

4.4. Using the Spectrum M4i6631-x8 to generate tweezer patterns

Tweezer patterns can be generated using any RF generator; however, to generate large, dynamic patterns, it is helpful to make use of arbitrary waveform generators (AWGs). The Spectrum PCIe card series has been used in similar experiments before [36,87,105,106], and is a good candidate. It has a high sampling rate at 1.25 GHz, such that high frequency tones can be generated, a large memory for larger sequences of RF signals and since it is a PCIe card, data transfer from the computer to the card is fast, which is necessary when doing real-time feedback from the experiment to the tweezer array (e.g. when sorting atoms or for doing conditional gates). When even faster

Feature	Property
Number of channels	2
Max. sampling rate	1.25GHz
Data resolution	16bit
Form factor	PCIe x8 Gen2
Streaming speed PC to card	Up to 2.8GB/s

Table 4.1: **Properties of the Spectrum M4i6631-x8 card.** The Spectrum card is used to generate RF to drive the AODs.

computation is necessary, the card can be connected to a GPU to enhance real-time performance.

This chapter discusses how to work with the Spectrum card to generate static and dynamic tweezer patterns. Only the M4i6631-x8 card (see Table 4.1 for properties) is discussed; other cards probably work the same way, although data formats and available features may differ.

4.4.1. Data formats

Programming the card always works by transferring data points to the card's memory. Data points are 16-bit signed integers, such that a cos wave is in the range $[-(2^{15}-1), 2^{15}-1]$. Data points that overflow the range wrap to the other end and will result in incorrect waveforms. It is recommended to include a function in the code to dump the waveform sent to the card memory into a file for examination. Waveforms that are dumped can, for example, be Fourier-transformed to analyze the spectrum, which maps directly to tweezer positions.

The card reads out the memory based on the sampling rate. Effectively, there is a pointer that starts at a defined position, reading one 16-bit value per timestep and converting it to an analog value in the DAC. Data can be written into arbitrary positions of the memory. For the replay, a start position and length can be set. For example, a memory with size $n = 2^{12}$ samples at a sampling rate of $f_{\mathrm{SR}} = 1.25\,\mathrm{GS\,s^{-1}}$ is read out in $t = \frac{n}{f_{\mathrm{SR}}} = 3.28\,\mu\mathrm{s}$.

The experiment uses both channels of the card, one for each axis of the AOD. In a multi-channel setup, the channels are multiplexed, so that the memory alternates between data values of channel 0 and channel 1, see Figure 4.6. This poses a challenge when fast transfer rates are required, as the data points for the channels are generally stored

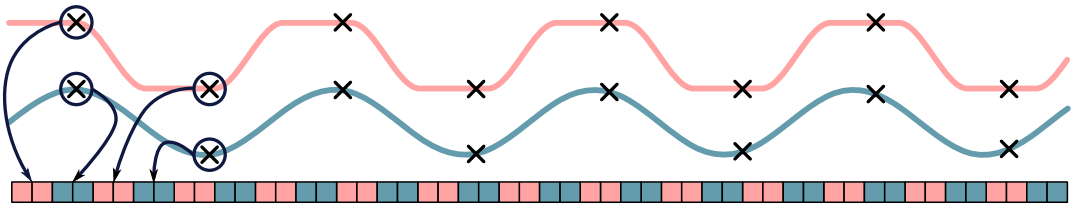


Figure 4.6: **Memory layout of the Spectrum card.** In the 2-channel configuration, the one-dimensional memory of the spectrum card is set up to alternate between data of the first and second channels. The figure shows two example waveforms with datapoints relating to the respective memory segments.

separately. Therefore, the memory should be filled before the experiment starts. The card has some replay modes that can help make efficient use of memory. However, features with a gap in replay are not discussed because, to keep atoms in tweezers, the replay must be fully continuous.

4.4.2. Replay modes

The Spectrum card offers five replay modes, all of which are suitable for generating tweezer arrays. In fact, while writing this thesis, a new replay mode was released: the direct digital synthesis (DDS) mode, which is especially useful for generating tweezer patterns. Even though it has now become the most important tool, in the scope of the thesis, we were able to sufficiently test and use all other modes and thus do not discuss the DDS mode. The following replay modes may still be helpful in applications that do not necessarily need sine waveforms.

Standard replay mode
In standard replay mode, the memory is filled once and then replayed. The memory size can be defined, as well as the number of times the defined memory is replayed. In general, the number of loops should be set to 0, such that the memory is replayed indefinitely.

FIFO single replay mode
In this mode, the memory is replayed as in the standard replay mode; however, the on-board memory acts as a FIFO buffer, meaning that new data points have to be streamed continuously. Effectively, when the memory pointer has read a memory value, that memory block is invalidated until written again. The computer has to run a loop, where data points are continuously transferred to the card, while it is still playing back. This

data mode may seem helpful in dynamic tweezer arrays (e.g., for sorting); however, there are some caveats.

The writing speed must be faster than the reading speed; otherwise, the card will encounter an underflow because it tries to read invalidated data. This process, however, also means that if data points are calculated live, the writing speed will decrease because both the transfer and calculation occur on the computer. It is not possible to use the full sampling rate because the PCIe x8 transfer speed for two channels with 16-bit data points is insufficiently fast.

Moreover, especially in this mode, the card needs extra cooling because during continuous data transfer, it can easily exceed 90 °C, at which point it will stop. These temperatures were generally reached after ~5 minutes of transferring data. As before, the length of the memory and the number of loops can be set.

Sequence replay mode
This mode was eventually used for sorting in the experiment. Here, the memory is split into segments of equal size. Each segment contains additional configuration about the replay. The options are:

- Replay length (even though the segment memory is fixed, the amount that is replayed can be less)
- How many times the segment is played back
- Which segment to run next
- End condition: The segment will always be played back `n_loop` times; however, the card can be configured to check for a trigger or not. If it doesn't, the segment moves on after playing `n_loop`. If it is configured to check for a trigger, the card plays back the segment `n_loop` times, checks for a trigger, and if none is received, plays the segment again `n_loop` times.

The sequence replay mode is handy for dynamic tweezers because the memory and segment configuration can be written live. However, data is not invalidated in the FIFO replay mode, allowing the card to continue playback even if nothing is written. This means that one segment can be run while another is written, thus allowing the full sampling rate to be used.

The sequence replay mode is most efficient if no data has to be written during runtime; however, that only works as long as the on-board memory is not filled. Writing a new segment configuration, however, can happen in real time. In the case of sorting atoms, the starting pattern, the ending pattern, and all possible paths are pre-calculated and kept in the computer memory. Each of these blocks will fill one segment, and if all blocks fit into the memory, they are transferred before the experiment starts. The number of

loops for all segments is set to 1, and loops are achieved by configuring the segment so that the next segment to run points to itself. At the beginning of the experiment, the starting pattern loops until it is known which path to play back for the sorting. At that point, the segment playing the sorting path is configured to point to the ending pattern. The ending pattern points to itself, and finally, the starting pattern is configured to point to the sorting segment.

The downside to using this mode, however, is that all segments must have the same size, even though sorting could be done faster. Otherwise, continuity between waveforms cannot be guaranteed; this point is further discussed in Section 4.5.

Replay using GPU

The sequence replay mode is limiting when it is not possible to pre-calculate all paths, or when different segment sizes would be required. In this case, replay using the GPU is possible. It is the most versatile of the modes; however, it requires much more programming effort.

In this mode, the memory is continuously read out as in the FIFO single replay mode; however, calculation and transfer happen on the GPU. On the CPU, the user configures a block size called `notifysize`, which is the amount of memory allocated before new data needs to be written. A loop is running, requesting the current memory position to be written, and that info is relayed to the GPU. The GPU can run in parallel for each data point, such that for a sufficient `notifysize`, the time required to calculate one block is the same as the time for calculating a single data point.

However, it is important to note that there is an overhead when running functions on the GPU, which depends on the amount of data that needs to be passed. This can quickly accumulate when complicated waveforms need to be played (e.g., multitone waveforms with ramps). Moreover, even though the calculation is almost instantaneously, there is a delay of at least one `notifysize` / `sampling_rate`. The `notifysize` cannot be arbitrarily small, since then underflows can happen again, when the GPU cannot catch up to the reading speed. As in the case of the FIFO replay mode, the sampling rate has to be reduced.

4.5. Debugging tools for RF-based optical tweezers

When we began using waveforms generated on the Spectrum card to create optical tweezer arrays for atoms, we encountered some pitfalls that will be noted here.

The atoms are quite sensitive to any issues in the frequency domain, making the spectrum analyzer the best debugging tool. If something looks off there, then it is unlikely that the trap works, even though the auxiliary camera might indicate otherwise. In this regard, the most significant issues were with continuity in the waveform. This is especially important when the waveform is digitally generated, e.g., in the case of the Spectrum card, the played waveform will always repeat (unless it is generated on the fly, as in the case of the FIFO and GPU mode).

Waveform continuity
To ensure the waveform remains continuous when repeated, it must fit within the time window in which it is played back. To achieve that, the frequency is slightly adjusted until the start and end nodes of the waveform match. A word of caution: The last datapoint is one timestep ahead of the first datapoint. This is solved easily analytically in the following way, where `N` is the number of datapoints in the waveform:

The time window is $t_{\text{window}} = \frac{N}{f_{\text{sr}}}$, where f_{sr} is the sampling rate. Then any frequency $f_n = \frac{n}{t_{\text{window}}}$ fits into the time window for all $n > 1$. Therefore, any desired frequency has to be rounded to the closest f_n, or in other words, $f_n * t_{\text{window}}$ needs to be an integer. In Python, this is computed as follows:

```python
def round_frequency(sampling_rate, num_samples, f, mul=1):
    t_window = num_samples / sampling_rate
    num_tooth = mul * round(f * t_window / mul)
    f_new = num_tooth / t_window

    return f_new
```

In the case where the tweezer array is evenly spaced with a center frequency f_c and spacing Δf, then it is sufficient to round f_c and Δf, which also guarantees an even spacing after rounding. However, there is a special case to consider: when the number of tones is even, no tone is present at the mean of all frequencies. Then `num_tooth` is uneven, and the tones are not continuous. In that case, `mul=2` guarantees rounding to even teeth only.

A waveform composed of multiple tones $\Phi_i(t)$ is written as $\Phi(t) = \sum_i \Phi_i(t)$. When each tone is continuous, such that

$$\Phi_i(t_{\text{start}}) = \Phi_i(t_{\text{end}}) \tag{4.41}$$

then the full waveform is also continuous, since

$$\Phi(t_{\text{start}}) = \sum_i \Phi_i(t_{\text{start}}) \overset{(40)}{=} \sum_i \Phi_i(t_{\text{end}}) = \Phi(t_{\text{end}}) \tag{4.43}$$

Therefore, to make the full waveform continuous, it is sufficient to round each tone.

Verifying the waveform

To ensure the correct waveform is applied to the AOD, it is crucial to inspect it beforehand. Most importantly, if the RF power is too high, the crystal can be destroyed. While all other sources of errors do not cause critical failure, they can cause delays when trying to debug the problems.

Thus, the waveform should be inspected before it is sent to the AOD. The first two steps involve checking the raw data played on the waveform generator. After that, the RF output is plugged into a spectrum analyzer. Here, there are a few key points to look for.

First, the total output power of the signal should not exceed the AOD damage threshold. The total output power is the sum of the individual tones in a linear scale. If the powers are measured in dBm, then the total output power is calculated as: $P_{\text{total}} = 10\log_{10}\left(\sum_i 10^{P_{\text{dBm,i}}/10}\right)$ and the spectrum should be band-pass filtered, such that only the relevant tones contribute. As the signal will be amplified, it is important to characterize the amplifier not only for single-tone signals but also for multitone signals, as was done in Figure 4.7.

Seeing discontinuity directly on the spectrum analyzer is not possible; however, if the signal is not clean (e.g., a weird noise floor), it strongly indicates that something is wrong, such as a large noise floor. Lastly, the tones should not drift and should be referenced to a global clock in the lab, such as a 10MHz reference.

For better discontinuity analysis, it is recommended to view the output on an oscilloscope. Here, it is necessary to trigger on the start of the signal and watch for breaks. To improve on this solution, the oscilloscope trace can be converted into a spectrogram, where discontinuities can be seen directly. A Python script can acquire the oscilloscope trace and directly convert it into a spectrogram. As shown in Appendix E. Figure 4.8, this example illustrates a spectrogram trace with and without discontinuity errors.

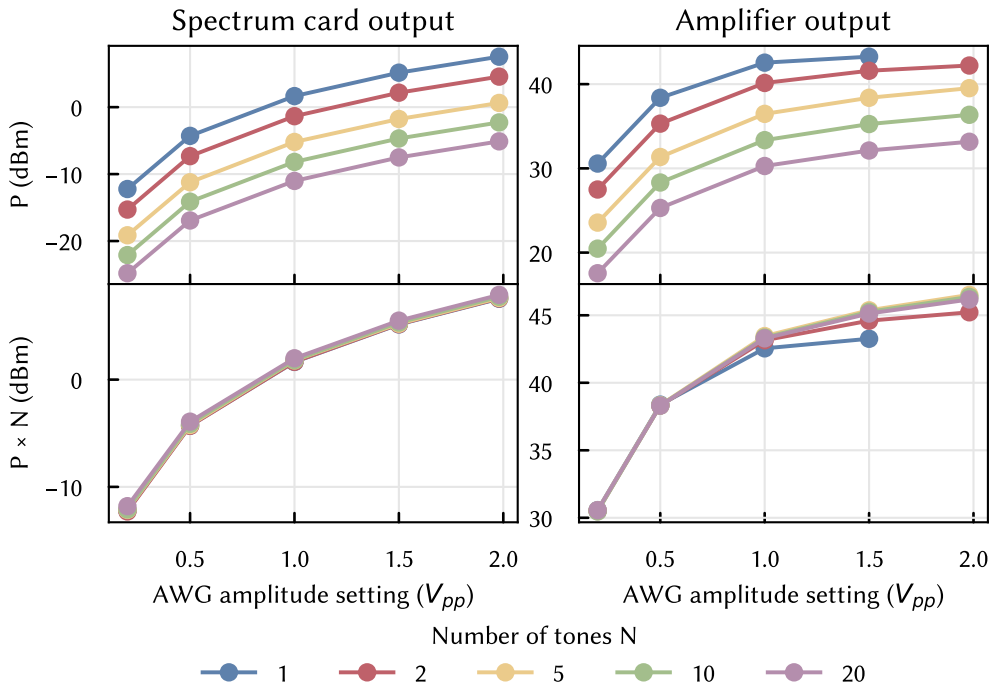


Figure 4.7: **Output power of a multitone signal.** The amplifier for the RF signal to the AOD is characterized by measuring the power after setting a multitone signal on the spectrum card, then measuring the output power after the amplifier. The output was attenuated due to the limited power bandwidth of the spectrum analyzer. The lower plots illustrate the power when the signal is multiplied by the number of tones applied, indicating that the output power per tone remains unchanged in the multitone setup. Deviations are only seen on the output after the amplifier; however, that is likely a result of the nonlinear properties of the attenuator.

4.6. Aligning the objective

The objective is one of the most critical opto-mechanical parts in the system. It focuses the tweezer light to create dipole traps for the atoms, it collects fluorescence photons of the atoms, and will later be used for addressing individual atoms. It defines the connectivity and geometry of the system, and thus, its alignment needs to be perfect. The procedure is two-fold: The objective needs to be centered and aligned perfectly parallel to the glass cell, and its focus must be at the position of the atoms, which is only precisely known when the MOT has been created. This means the alignment has to involve the atoms and cannot be done just by eye. Secondly, as the objective is used to image single atoms, the imaging system needs to be aligned with it. This primarily involves aligning

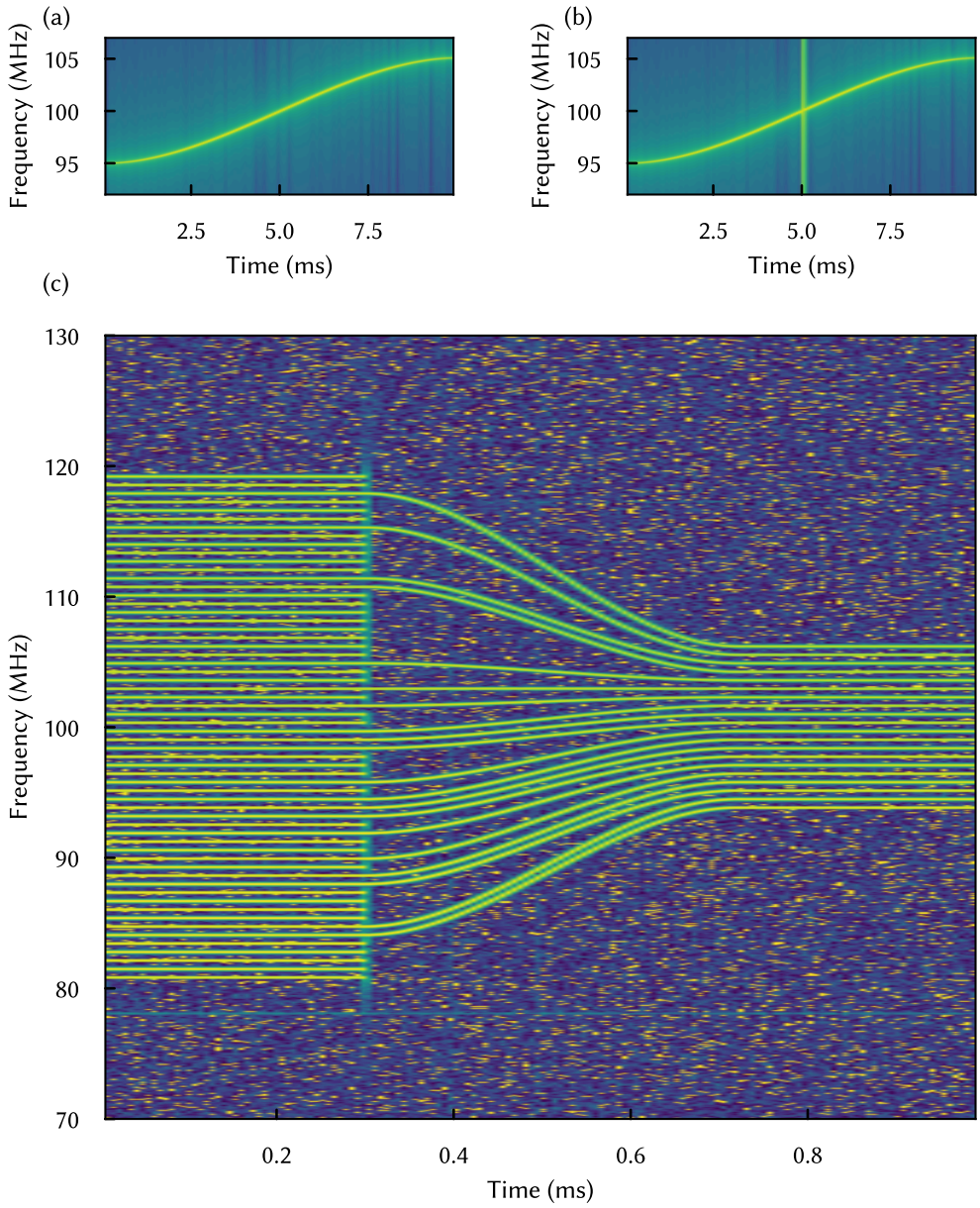


Figure 4.8: **Spectrogram of a constant jerk trajectory.** Plots (a) and (b) show a jerk ramp that is used to move a tweezer from one position (95 MHz) to another position (105 MHz). In (b), the waveform performs a phase jump halfway through, indicating that the spectrogram is a good tool for identifying discontinuities. Plot (c) shows a real oscilloscope trace of a full sorting sequence, collapsing selected rows. The only discontinuities visible are at trap sites, which are turned off.

the imaging lens before the camera and adjusting the camera's position, so that the 4f-condition is fulfilled and the image of the atoms is projected onto the camera chip. A procedure was devised, which is explained below.

We use a reference beam for the alignment procedure, which is initially collimated using an infinity corrected camera. There are two requirements to the alignment beam: first, it needs to pass the glass perpendicular; second, it needs to be aligned to the MOT, as the reference beam is used for the tweezer position, which is inside the MOT in the best-case scenario. The angle to the glass cell is aligned by looking for the reflection from the first surface; therefore, we have a very long beam path (5 m). The reflected beam is then coupled back into the collimator that outputs the reference beam. The MOT is then loaded, and as the reference beam gets closer to the atoms, they are heated out and the fluorescence decreases. Both procedures are repeated iteratively until convergence.

To align the angle of the objective, we will once again look for a reflection. However, this will be much harder because the reference beam passes through two glass surfaces. Moreover, the imaging system can only be aligned if there is an image to project. The vacuum system is on precision rails, allowing us to retract the glass cell and place a reference glass plate instead. The angle of the glass plate is aligned with respect to the reference beam. Now we notice that a beam that is reflected from the first surface of the objective interferes with the reflected beam from the glass surface, creating a shearing interferometer. The fringes are distorted for a misaligned objective; correcting the distortion aligns the objective to the glass plate.

With the objective in place, there is a well-defined focal point where atoms will later be trapped. To align the imaging system, we insert a target TC-RT01 on a 5-axis stage, position it, and then image it onto the camera. We can move the target to select which pattern or pinhole we want to image. The reference beam traverses a pinhole, and we use an iris before the camera, along with the camera chip itself, to ensure the reference beam enters perpendicular to the camera and is centered on the chip. As the reference beam is likely not perfectly centered on the pinhole, aberrations are visible as a result. The pinhole is moved in the xy-plane, and the mirrors are walked again to centralize the beam. This process is repeated until the aberrations have vanished.

In the final step, the tweezer beam is overlapped with the reference beam. When the target is imaged, the tweezer beam should also be visible on the camera when it reflects off the target. The beam is moved until it enters the pinhole Figure 4.9, indicating that it is aligned on the designated position. As the tweezer beam does not yet enter the pinhole perfectly, aberrations will be visible. After optimizing those out, we have a good starting position to find atoms in tweezers. The vacuum system is pushed back into its

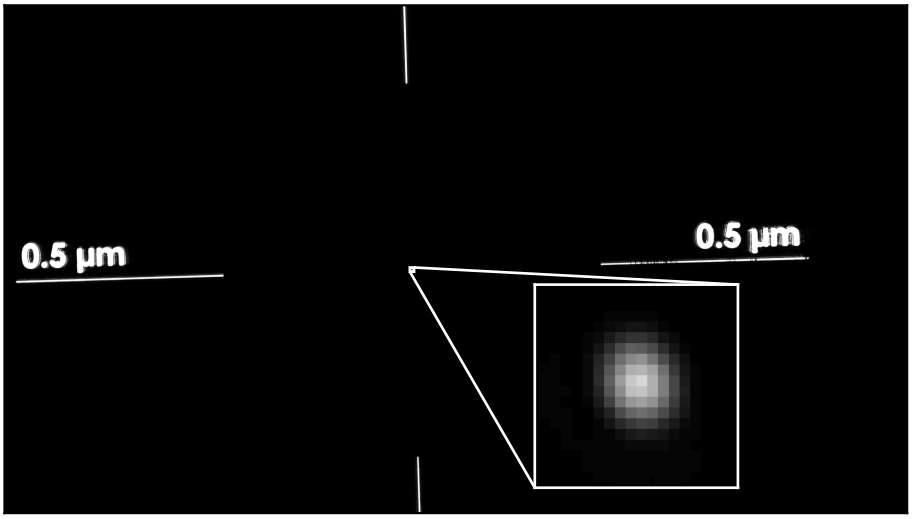


Figure 4.9: **Image of a 500 nm pinhole during alignment.** First, the text and lines are brought into focus by moving the pattern up or down. Then, the point spread function (PSF) is the feedback in the alignment of the beam, where the goal is to reduce aberrations.

original position, and atoms are loaded and imaged. Further optimizations then follow using the atomic feedback.

5. Trapping, cooling, and measuring single atoms

The area of quantum many-body systems excels in its control over atoms, as they can be cooled to their electronic and motional ground state with high fidelity [65]. Atoms can be individually accessed and read out in lattice or tweezer experiments, where they are trapped individually in potential minima. This amount of control is beneficial in quantum computing applications, which require high-fidelity qubit interactions on these manifolds. The process of cooling and loading single atoms is described in the following chapter, including our experimental results.

5.1. Towards single ultracold atoms

Atoms are loaded into single tweezers and cooled to their electronic and motional ground state in a seven-part sequence:

- Transporting atoms from the oven to the glass cell during blue 2D-MOT
- Capturing and trapping atoms as a cloud in a blue MOT
- Further cooling and compressing the cloud in a red MOT
- Loading the atoms into optical tweezers, then heating out pairs of atoms during parity projection, keeping only single atoms
- Reading out the occupation in the tweezers by taking fluorescent atoms in an imaging sequence
- Cooling single atoms using a Sisyphus cooling protocol
- Cooling motional states of single atoms to the ground state during sideband cooling

Each stage underwent a set of optimizations to determine the correct laser powers, frequencies, timings, and magnetic fields, as well as to improve alignment. The whole timing sequence is given in Figure 5.1 and Figure 5.2. The overall sequence time can still be improved; however, this thesis focuses on bringing the system into a functional state.

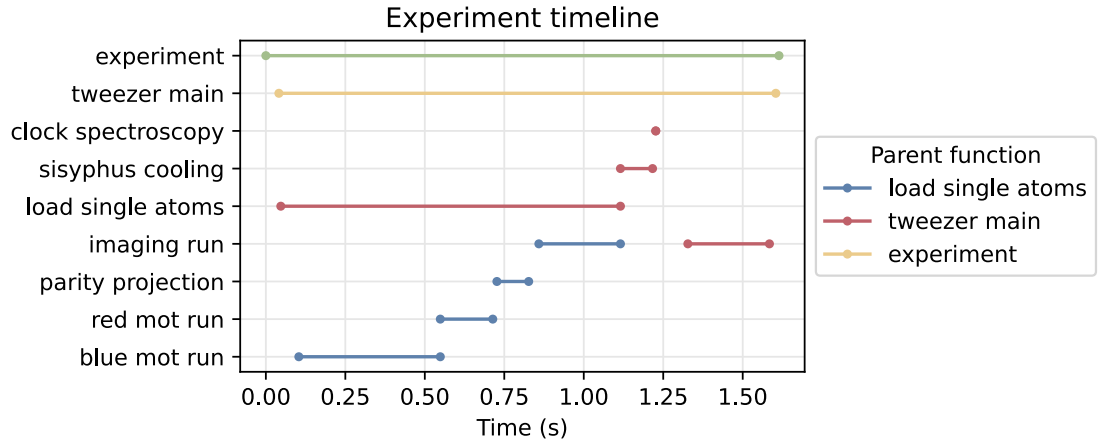


Figure 5.1: **Experimental timeline.** The timeline of the experiment is shown, where each line corresponds to a function in the Python code, with its runtime according to the x-axis. Colors represent the parent function, from which they were called.

5.2. Cooling in a magneto-optical trap

Atoms are trapped and cooled first in a blue MOT, which is directly followed by a red MOT. In our experiment, we chose to use six fibers instead of counter-propagating beams. They are aligned as shown in Figure 3.14: Two beams enter the glass cell on the horizontal plane along the y-axis (transverse to the atom beam direction). In the x-direction, there is only access from the front surface of the glass cell; the back surface is connected to the rest of the vacuum system. Due to this optical obstruction, and the additional restriction along the z-axis from the objective, the other two pairs of beams enter the glass cell diagonally in the xz-plane at an angle of $\pm 25°$ to their horizontal plane. Both the red and blue beams are combined on dichroic mirrors before entering the glass cell, with individual control over their respective polarization via $\lambda/2$ and $\lambda/4$ waveplates. The beam sizes are 7.5 mm for all beams. As the glass cell is not coated, refraction will shift the laser beam and cause power loss, as shown in Figure 3.5.

Various sources are discussing the theory behind the blue and red MOT in ^{88}Sr [55,107,108]. As a general overview, the level scheme in Figure 2.1 shows the transitions relevant for the experiment. For the blue MOT Section 2.8, the y-bias coils in anti-Helmholtz configuration create a magnetic field of 0.67 G m^{-1} at 230 A applied current. Due to the magnetic field gradient and the uniform emission of spontaneous photons, the atoms experience a position-dependent force, eventually trapping them in the field center.

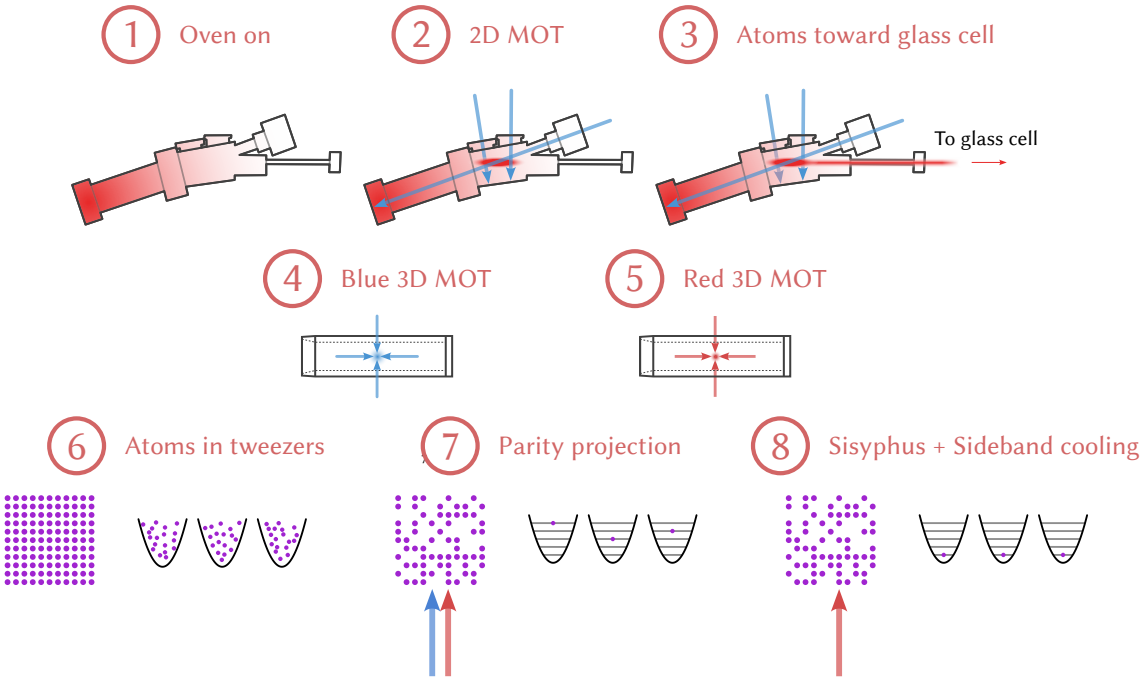


Figure 5.2: **Pictographic timeline of the experiment.** The figure shows how atoms are prepared in the experiment. They are heated in the oven section of the AOSense module, where they evaporate uniformly. They are cooled transversally in the 2D-MOT and longitudinally using the Zeeman slower beam, after which they are on a trajectory towards the glass cell. Here, atoms are cooled and trapped in a dual-stage MOT, after which they are transferred to the tweezer array. We perform parity projection to heat out pairs of atoms, until only single atoms remain. Atoms are then cooled using Sisyphus and sideband cooling, until they end up in the motional ground state.

Atoms that end up in the $5p^3P_1$ state quickly decay back to the electronic ground state $5s^1S_0$. However, the decay channel $5p^1P_1 \rightarrow 4d^1D_2$, although unlikely, is still populated due to the large amount of scattering events. This state can only decay into the $5p^3P_J$ manifold and is then problematic for two reasons:

- The decay into the triplet manifold is slow, taking $\tau = 330\,\mu s$
- The $5p^3P_2$ state does not naturally decay; therefore, atoms stuck there are transparent to the cooling light

To counteract this, a repumper on the $5p^3P_2 \rightarrow 6s^3S_1$ line at 679.29 nm causes atoms to be pumped into the upper state. From there, they can decay into any of the three $5p^3P_J$ states. As the $5p^3P_0$ state is stable, another repumper on the $5p^3P_0 \rightarrow 6s^3S_1$ at 707.20 nm pumps atoms back, such that eventually they all end up in the unstable $5p^3P_1$.

Alternatively, atoms that decay to the $4d^1D_2$ state can already be intercepted before they decay into the triplet manifold. It is possible to repump them into the np^1P_1 states, where

$n \in [6, 7, 8]$. This process is vital for future experiments, where atoms pushed out using the 461 nm light can survive in the $4d^1D_2$ and are then detected as false positives.

We follow the red MOT implementation detailed in [109]. There, the red laser frequency is adiabatically swept across three magnetic sublevels of the $^3P_1$ state, resulting in a so-called sawtooth-wave adiabatic passage (SWAP). Experimentally, this is easily achieved by maximizing the red MOT absorption signal by varying the sweeping range and sweeping frequency. During the transition from the red to the blue MOT, the red lasers are turned on, and the blue laser power is ramped down. The linewidth of the red MOT transition compared to the blue MOT is much narrower, resulting in a denser cloud. To increase the fraction of captured atoms, the magnetic field gradient is ramped down to $0.059\ \mathrm{G\ m^{-1}}$ in 30 ms.

We found that to increase the absorption signal even further, the SWAP MOT stage can be followed up by another single-frequency MOT stage, during which the frequency of the 689 nm laser remains constant. Example absorption signals are given in Figure 5.3. Following this, atoms are transferred into optical tweezers and, in a parity projection sequence [110], pairs of atoms are excited out of the trap such that only single atoms remain.

5.3. Loading atoms into optical tweezers

The lasers used to generate the tweezers has a wavelength of 813.4 nm, which is magic for the ground state $^1S_0$ and the clock state $^3P_0$, i.e. the differential light shift between the states is zero [111,112].

In the following stage, pairs of atoms are heated out in a process called parity projection [29,96], after which tweezers contain either zero atoms or one atom. Molecular resonances are used to excite the pair transition and two of these molecular transitions [110] are available close to lasers that are already used in the experiment - the imaging light at 461 nm and the 689 nm laser for the $^1S_0 \rightarrow {}^3P_1$ transition, which is used for the red MOT, Sisyphus and sideband cooling.

During the parity projection sequence, both imaging and red cooling lasers, along with the 707 nm and 679 nm repumpers, are switched on for 40 ms. The parameters that can be iterated over for the parity projection pulse are the powers and frequencies of the lasers, as well as the duration of the parity projection pulse. To optimize the sequence, histograms are taken (see Section 5.4) and optimized so that only zero-atom and one-

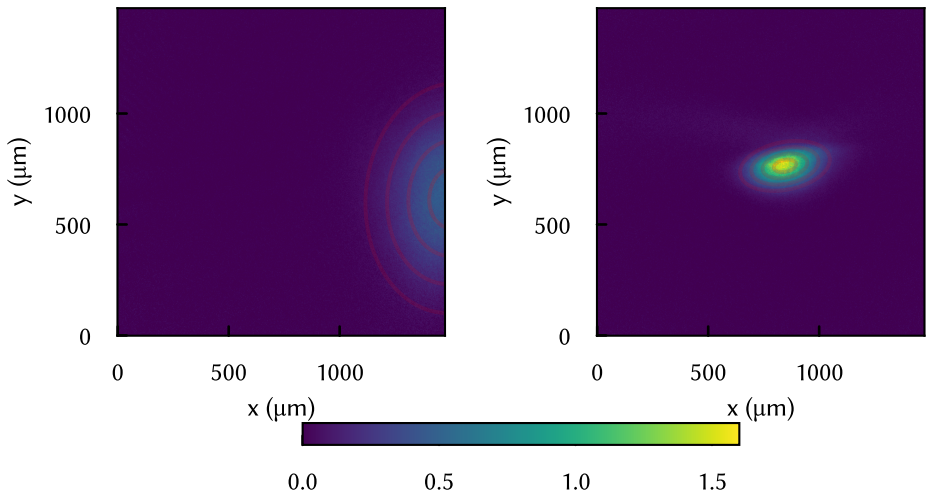


Figure 5.3: **Absorption image of blue (left) and red (right) MOTs.** The cloud is magnified via a 150/100 telescope, the pixel size is $3.69\ \mu m^2$. Fitting a 2D Gaussian to the image results in 1.5×10^6 atoms in the blue MOT and 0.48×10^6 atoms in the red MOT. Each line in the image corresponds to the $n\sigma$ for $n \in \mathrm{N}$ radius. For the blue MOT, we have $\sigma_{\mathrm{major}} = 256\ \mu m$, $\sigma_{\mathrm{minor}} = 191\ \mu m$. For the red MOT, we have $\sigma_{\mathrm{major}} = 101\ \mu m$, $53\ \mu m$. The loading times were 500 ms for the blue MOT, 150 ms and 25 ms for the red SWAP and single-frequency MOT.

atom peaks remain. Once the histograms looked good enough (as in Figure 5.4), we moved on to the cooling sequences.

5.4. Imaging single atoms

Atoms in tweezers are imaged by illuminating them with light, resulting in a high scattering rate and thus emitting many photons. At high driving field, the spontaneous emission rate converges to $\gamma/2$, for $^1P_1$, it is $\gamma = 30.29$ MHz [70]. The calculated fraction of collected photons can be calculated in the following way: Circularly polarized photons are emitted as $I(\theta) \propto \frac{1}{2}(1 + \cos^2(\theta))$ [113]. The objective has a NA: $\mathrm{NA} = n \sin(\theta_{\mathrm{max}})$, where we take $n = 1$ for air and vacuum. Then, the collected photons are in the range $\theta \in [90\,° - \theta_{\mathrm{max}}, 90\,° + \theta_{\mathrm{max}}]$, so in 3D, the fraction becomes:

$$
\begin{aligned}
f_{\text{collected}} &= \frac{\int_0^{2\pi}\int_{\pi-\theta_{\max}}^{\pi} I(\theta)\sin(\theta)\ \mathrm{d}\ \theta\ \mathrm{d}\ \varphi}{\int_0^{2\pi}\int_0^{\pi} I(\theta)\sin(\theta)\ \mathrm{d}\ \theta\ \mathrm{d}\ \varphi} \\
&= \frac{\int_{\pi-\theta_{\max}}^{\pi} (1+\cos^2(\theta))\sin(\theta)\ \mathrm{d}\ \theta}{\int_0^{\pi} (1+\cos^2(\theta))\sin(\theta)\ \mathrm{d}\ \theta} \\
&= \frac{3}{8}\int_{\pi-\theta_{\max}}^{\pi} (1+\cos^2(\theta))\sin(\theta)\ \mathrm{d}\ \theta \\
&= \frac{1}{8}(4-\cos^3(\theta_{\max})-3\cos(\theta_{\max}))
\end{aligned}
\tag{5.1}
$$

The objective here has a $\mathrm{NA} = 0.65$, so that $f_{\text{collected}} = 0.160$, thus photons are collected at a frequency of $\gamma_{\text{collected}} = \frac{\gamma}{2} f_{\text{collected}} = 2.42\,\mathrm{MHz}$.

In order to read out the occupation of the tweezers, atoms are illuminated with off-resonant 461 nm light. At a high scattering rate of $\Delta_{\mathrm{FWHM}} = 2\pi \times 30.29\,\mathrm{MHz}$, photons are spontaneously emitted. They are collected on a camera, as they pass through an objective and a telescope to increase the magnification.

The first signal was found by generating a single tweezer, which we purposely clipped with an iris in the beam before the objective. Doing this increases the waist in the imaging plane close to the atoms. This method allows one to get close to a position where atoms can be caught. With this signal, the objective position is carefully adjusted while increasing the iris radius, until it can eventually be removed from the setup.

The imaging pulse consists of an imaging duration, where the imaging light is on, as well as the 679 nm and 707 nm repumpers (for the same reason as the blue MOT). As this results in heating of the atoms, we do Sisyphus cooling at the same time (see Section 5.5). The result is a monochrome image of atoms given by the geometry of the tweezer pattern. The image is analyzed by selecting a rectangular region of interest with a constant size around the tweezers. In the first step, all pixel counts in the regions of interest are summed and visualized in a histogram.

The signal from the camera has three main error contributions. For one, the readout electronics create a constant offset because the charges are not amplified perfectly. Secondly, general electrical noise can add false positive counts, and finally, the background light level around the glass cell or in the imaging path adds another constant offset. Thankfully, the camera has a low-noise mode suitable for single photon counting, thus random electrical noise is not an issue. After subtracting the offset, the pixel count in a region of interest around the tweezers is proportional to the number of atoms in the tweezers. Therefore, the histogram will show several Gaussian distributions, spaced by

a constant value, where the first peak refers to an empty tweezer, the second peak to one atom in the tweezer, etc. The optimal imaging sequence is found when the zero-atom peak and the one-atom peak are clearly separable. The peaks can be more distinctly separated by emitting more photons, i.e., when the laser power is higher. If there is a bridge connecting the two peaks, it is most likely due to atom loss during imaging, where the heating rate exceeds the cooling rate until atoms escape the tweezer potential. Examples of imaging histograms are given in Figure 5.4.

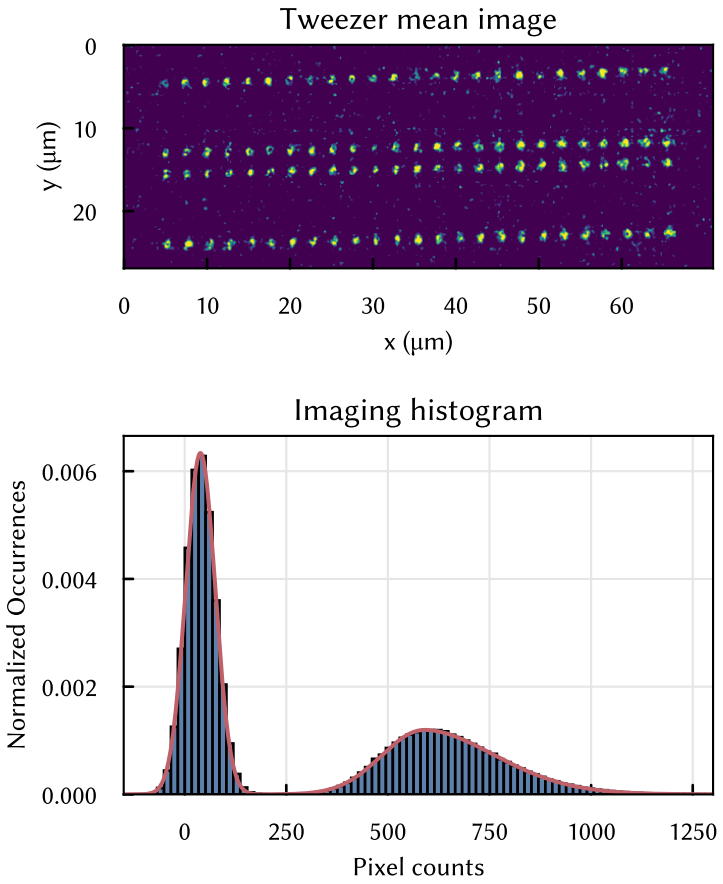


Figure 5.4: **Imaging histograms.** The upper image is an average of 5000 shots of atoms loaded in optical tweezers. The histograms show pixel counts in regions of interest around every tweezer. Empty tweezers generally have pixel counts close to zero, which is seen as the higher of the two peaks. The right peak corresponds to pixel counts when an atom is loaded. The fit consists of a Gaussian and an asymmetric super Gaussian, as heating effects during imaging introduce a bias towards lower counts. This image was taken at a later stage of the experiment, with the SLM already in place.

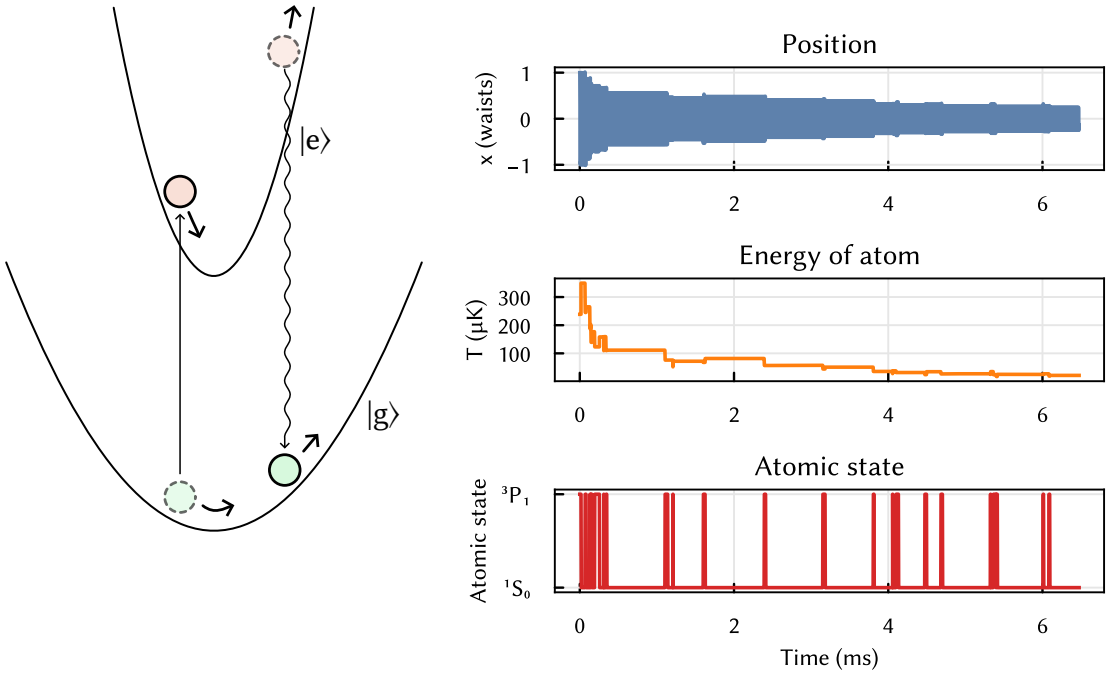
5.5. Sisyphus cooling

After only single atoms remain in the trap and an image has confirmed them, they need to be cooled to the motional ground state. The first stage uses the so-called Sisyphus cooling effect, which is then followed by sideband cooling afterwards. The Sisyphus cooling mechanism used here relies on the fact that there is a trap-induced light shift difference between the ground and excited states. This is theoretically explained in [87], and more in-depth in [114] and [115]. A pictorial explanation is given below.

In the semiclassical picture, the atom oscillates back and forth in the harmonic trap, conserving the total energy $E = E_{\text{pot}} + E_{\text{kin}}$. The gradient of the trap determines its velocity. However, when the atom changes state, it encounters a new potential determined by the polarizability of the state, and thus, the gradient also changes. This can lead to cooling if, for example, the atom starts in a tight trap and then changes states into a loose trap at the turnaround point where its velocity is zero. It has used up all its kinetic energy to climb the steep hill of the tight trap, but now gains less kinetic energy as it falls down the loose trap.

With this, the atom can be effectively cooled if we can create a circumstance where it gains less velocity relative to the previous state, i.e., moving only upward in the tight trap and only downward in the loose trap. The process of stimulated excitation is stochastic and symmetric; therefore, the probabilities of gaining more and less velocity on a trap change are both the same. However, spontaneous decay breaks this symmetry, as it is an effect only present in the excited state. Having the excited state trap be tight leads to a situation where the atom will always gain less velocity as it changes from the excited state to the ground state. It effectively gains kinetic energy only when it decays near the center (where the velocity is maximal) and is excited near the top (where the velocity is minimal). However, due to its motion, the atom spends the least amount of time in the center of the trap. Since it is now more likely to decay than to be excited, the atom effectively loses kinetic energy.

The process is schematically represented in Figure 5.5 with a classical Monte-Carlo simulation of a cooling process. It shows the motion of the atom, as well as its total energy, which changes every time the atomic state changes. The simulation uses literature values [87] for the polarizabilities for 813 nm of the $^1S_0$ and $^3P_1$ states, which are in atomic units $\alpha_{1S0} = 286$ au and $\alpha_{3P1} = 353.9$ au respectively. The trap is assumed to have a waist $\omega_{\text{trap}} = 750$ nm and depth $U_{\text{trap}} = 700$ µK. The driving light has a power of $P_{\text{drive}} = 1$ mW, waist $\omega_{\text{drive}} = 0.5$ mm and detuning of $\Delta = -2\pi \times 3$ MHz. The simulation assumes a harmonic trap; the number of points is based on the trap frequency in



a) Pumping and spontaneous emission during sisyphus cooling

b) Monte-Carlo simulation of sisyphus cooling process

Figure 5.5: **Pictorial representation of Sisyphus cooling process and Monte-Carlo simulation of effective cooling.** a) shows the ground and excited state potential wells, in which we can imagine the atom oscillating. A laser continuously drives the atom between the states. Changing from a shallow to a tight trap, the atom effectively gains energy on a downhill and loses energy on an uphill. Since the atom spends more time the higher it moves, spontaneous emission effectively removes kinetic energy, due to the steep uphill in the excited state. b) The Monte-Carlo simulation only assumes classical movement with stochastically changing potentials and the added probability of spontaneous emission. The energy of the atom is calculated based on its current velocity and position in the trap. More details on the simulation are in the main text.

the ground state, such that without the driving beam, it would do 500 oscillations. Two hundred points per oscillation are evaluated for numerical stability. In classical Monte Carlo simulations, the position is updated based on the force applied at each time step. Here, the Verlet integration [116] is used, where

$$
\begin{aligned}
a_i &= \frac{F(q; x_i)}{m_{\mathrm{Sr88}}} \\
x_i &= x_{i-1} + v_{i-1}\Delta t + \frac{1}{2} a_{i-1}\Delta t^2 \\
v_i &= v_{i-1} + \frac{1}{2}(a_{i-1} + a_i)\Delta t
\end{aligned}
\tag{5.2}
$$

where a_i is the acceleration, v_i the velocity and x_i the position of the atom in the trap. The force $F(q; x_i)$ depends on the current quantum state $q \in [g, e]$ and is then the derivative of the trap potential of that state. Although the atom is simulated with classical motion, the excitation and decay are quantum mechanical. At each time step, a random number $r \in [0, 1]$ is evaluated against the scattering rate of the state Γ to test for a decay (if the atom is in the excited state) or against the stimulated emission scattering rate Γ_{sc}:

$$
\Gamma_{\mathrm{sc}} = \frac{\Gamma}{2} \frac{I_{\mathrm{drive}}/I_{\mathrm{sat}}}{1 + I_{\mathrm{drive}}/I_{\mathrm{sat}} + 4(\Delta_{\mathrm{eff}}/\Gamma)^2}
\tag{5.3}
$$

where $I_{\mathrm{drive}} = \frac{2P_0}{\pi\omega_0^2}$ is the peak intensity of the driving light (uniform across the array). Then, if the condition $r < \Gamma\Delta t$ is true for the ground state, or $r < \Gamma\Delta t$ or $r < \Gamma_{\mathrm{sc}}\Delta t$ for the excited state, then the atom's state is flipped. This process works because spontaneous emission is a symmetry-breaking process that applies only to the excited state. An example of a result is presented in Figure 5.5. This model only provides a slight idea of the working principle of the Sisyphus cooling process; for a complete understanding, the waveform of the atom needs to be taken into account.

To perform sisyphus cooling in the experiment, the 689 nm laser is red detuned with respect to the $^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}_1$ transition. As Sisyphus cooling is used during imaging, the cooling feature can be found by scanning the laser frequency around the transition frequency and measuring the survival after the imaging stage. This procedure will produce the plot in Figure 5.6. Due to varying trap depths in the tweezers, the resonance will not have the same frequency across the array. Therefore, the cooling feature itself can be used in a feedback loop to homogenize the tone amplitudes of the tweezer array.

5.6. Calibrating magnetic fields

In the next cooling stage, the magnetic offset fields are used to shift a magnetic sublevel of $^3\mathrm{P}_1$ into a magic condition with the ground state trap. To achieve good precision, it is necessary to calibrate the magnetic coils correctly. Since the $^1\mathrm{S}_0 \rightarrow {}^3\mathrm{P}_1$ resonance

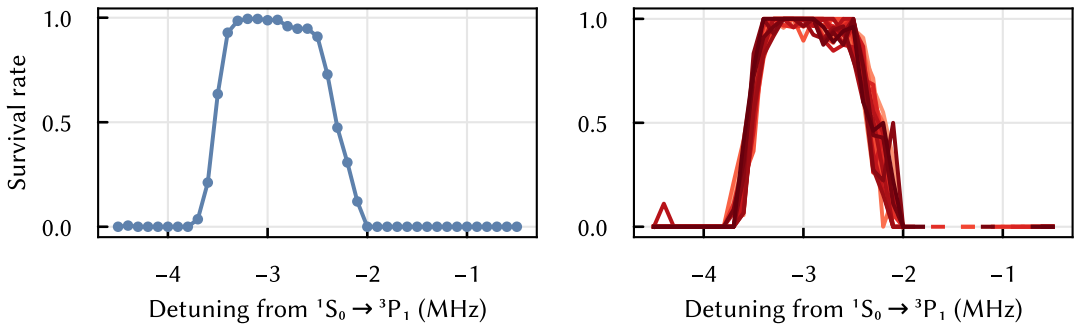


Figure 5.6: **Spectroscopic measurement of the Sisyphus cooling pulse.** We can see a relatively broad region around −3 MHz detuning, where the cooling works well and atoms are retained. Outside that region, atoms are heated out. The tweezer light is off during this measurement, as that would introduce a light shift.

is known, it is straightforward to apply different fields and test at which detuning the magnetic sublevels are found.

Example measurements are shown in Figure 5.7. The m_J splitting is recorded for varying magnetic fields, and the result is extrapolated to find the true zero field position. To avoid introducing light shifts, the tweezers are off during the measurement.

5.7. Sideband cooling

Another process to further cool the atom's motional state is sideband cooling [117]. It is purely quantum, in the sense that the quantization of the vibrational levels in the atomic trap is exploited.

5.7.1. Theory and geometric considerations

In this process, a laser is detuned to drive a transition from a high vibrational state n_g in the ground state trap to a lower vibrational state $n_e < n_g$ in the excited state trap, which in the following is $\Delta n = n_e - n_g = -1$. In the Lamb-Dicke regime, spontaneous decay for $|n_e - n_g| > 1$ is strongly suppressed, so that from $n_e = n_g - 1$, the atom is most likely to decay back into one of $[n_g, n_g - 1, n_g - 2]$, which will eventually lead to the atom in $n_g = 1$, where no more excitations of the driving light are possible.

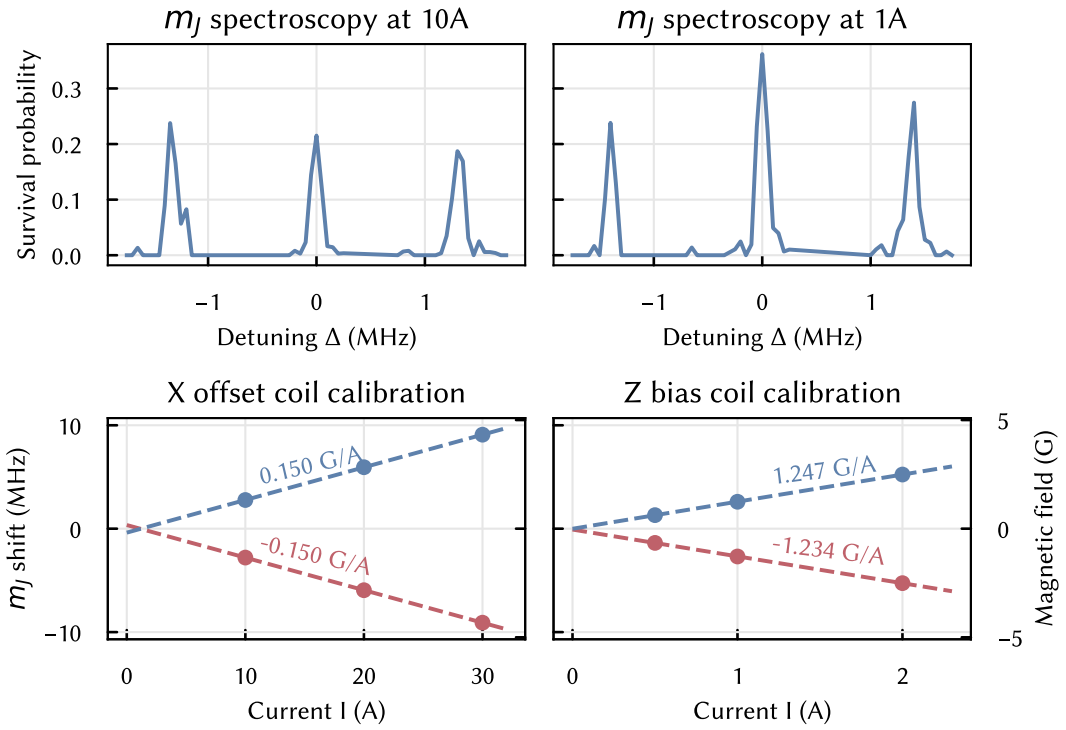


Figure 5.7: **Magnetic coil calibration.** For the x offset (left column) and z bias (right column, used as an offset coil here) on the $^1S_0 \rightarrow {}^3P_1$ transition, the magnetic coil calibration was performed. The top row shows example measurements of m_J splittings for one magnetic field. The peak positions are then plotted in the linear plots in the bottom row. Extrapolating to the zero position gives the current at which the magnetic field is truly zero. The magnetic field per current compares very well to theoretical and experimental values (measured using a magnetic field sensor) in Table 2.4.

However, as the motional state separation depends on the trap depth, this process is susceptible to differential light shifts between ground and excited states. Due to this, it is necessary to have a magic condition to cancel the differential light shift between the ground and excited states, unlike Sisyphus cooling, where different trap depths are required [118].

To achieve a magic condition, the Zeeman shift can be used to align the excited state magnetic sublevel with the same polarizability as the ground state. The quantization axis is well defined for high fields, and the effective projection depends on the angle of the tweezer polarization with respect to the magnetic field. For high fields, the polarizability is plotted in Figure 5.8.

We noticed that the process did not work well when the field was applied in one plane, but worked well in another. This discrepancy arises because our tweezers are generated

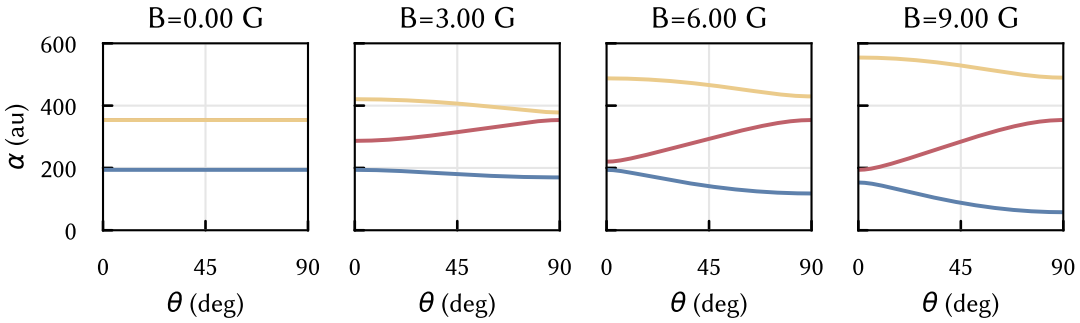


Figure 5.8: **$^1S_0$ and $^3P_1$ state polarizabilities.** The diagram shows the polarizabilities of the ground state $^1S_0$ and the excited state $^3P_1$ magnetic sublevels, m_F=-1,0,+1 (blue, red, green). As the magnetic field is increased, the splitting of the sublevels increases and the m_F=0 state forms a crossing for a magic condition close to 45°.

in different planes. The AODs are 25 mm apart, and considering that the tweezer beams pass through a 2f telescope with magnification $M = 7.5$ yields an axial magnification of $M_{\text{axial}} = M^2 \approx 56$. Therefore in the back-focal plane of the objective, the 25 mm separation is magnified to 1.4 m.

In the back-focal plane of the objective, the tweezer rays have an angle with respect to the objective plane, which, when focused down, results in a position in the atomic plane. The objective performs an effective Fourier transform from angles to positions. However, when the image of the AOD is not in the back-focal plane of the objective, the transformation is not perfect anymore, and the tweezers have an additional angular component in the atomic plane. This process is illustrated in Figure 5.9. From that diagram, due to an effective displacement of the AOD, we can derive an angular deviation $\delta\theta \approx \frac{\alpha\Delta z}{f_{\text{obj}}} = \frac{x\Delta z}{f_{\text{obj}}^2}$. For x the size of the atomic array (50 um), Δz the effective displacement of the AOD in the back-focal plane of the objective (1.4 m) and f_{obj} the effective focal length of the objective (24 mm), this results in an angular deviation of $\delta\theta \approx 6.9$ deg. This changes the magnitude of the electric field of the light along the quantization axis: $E_{\text{rot}} = E_0 \cos(\delta\theta)$ and therefore the intensity to $I_{\text{rot}} = I_0 \cos^2(\delta\theta)$, so that $\frac{I_{\text{rot}}}{I_0} = 98.5\%$, which is the change of the Stark shift on the $^1S_0 \rightarrow {}^3P_1$ due to the different polarization.

The best way to solve this is to overlap the AODs in the back-focal plane of the objective, which is achieved by a 1:1 telescope between the two AODs. Figure 5.9(b) shows the propagation direction of the tweezers, and we note that our pattern of choice is a 2x20 array, so an array that is elongated along one direction. The tweezers always point mostly towards z, so the long axis can be chosen to be along x or y, as

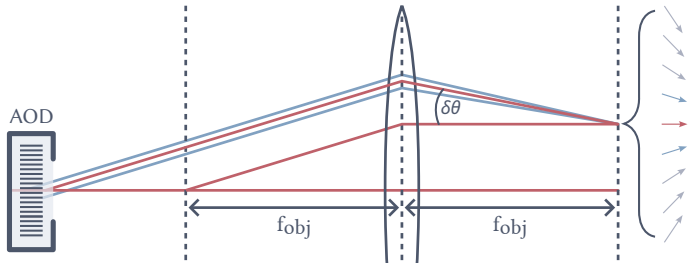


Figure 5.9: **Tweezer polarization of a 2D-AOD.** The schematic shows the effect of a beam that is not in the focus of the objective (shown as a single lens). It has an effective angle in the tweezer plane relative to a beam that is in focus. Transverse magnification of the telescope and the separation of the two AODs results in an effective displacement of the AODs before the objective. The arrows on the right show the resulting polarization of the optical traps in the tweezer plane.

well as the polarization (as it is perpendicular to the propagation direction). To reduce the error in polarization, the long axis should be perpendicular to the polarization axis. As an example, let us consider a 1D line of tweezers aligned along the x-axis, with its propagation direction $\vec{x}$ mostly along z, and the polarization $\vec{p}$ also along the x-axis. The angular deviation $\delta\theta$ increases with distance; therefore, the outermost tweezer n has the most significant deviation, and for that, the equations are as follows:

$$\begin{aligned} \vec{x} &= (\sin(n\,\delta\theta), 0, \cos(n\,\delta\theta)) \\ \vec{p} &= \vec{x} \times (0,1,0) = (-\cos(n\,\delta\theta), 0, \sin(n\,\delta\theta)) \end{aligned} \tag{5.5}$$

However, in the case where the polarization is along the y-axis, then

$$\begin{aligned} \vec{x} &= (\sin(n\,\delta\theta), 0, \cos(n\,\delta\theta)) \\ \vec{p} &= (0,1,0) \end{aligned} \tag{5.7}$$

Choosing the polarization correctly is crucial in reducing the error in polarization, arising from the geometric displacement of the two AODs. Of course, this only works since we are using an elongated array; there will still be an error, since there are two rows. However, it is smaller by a factor of n compared to the case where the polarization is chosen along the long axis.

5.7.2. Measurement

Before measuring the sideband cooling efficiency and the occupation of vibrational states, the magic angle must be calibrated, which eliminates the differential light shift

between the $^1S_0$ and $^3P_1$ states. The coils used to drive this field are the two offset field coil pairs along the y and z directions. We decide on a high field, in this case $B = 30$ G, and the magic angle θ_{magic} defined via $B_y = B\cos(\theta_{\text{magic}})$, $B_z = B\sin(\theta_{\text{magic}})$. From the calculations in Figure 5.8, we know the angle is around the 45° mark. It can be precisely determined by measuring the light shift of the transition at varying angles, as the goal is to find an angle where the light shift is zero. From Figure 5.10, we take the magic angle as 47.8°.

To verify that atoms are cooled to the ground state, we use a laser that couples to the vibrational sublevels on the $^1S_0 \rightarrow {}^3P_1$ transition. As we detune the beam across the

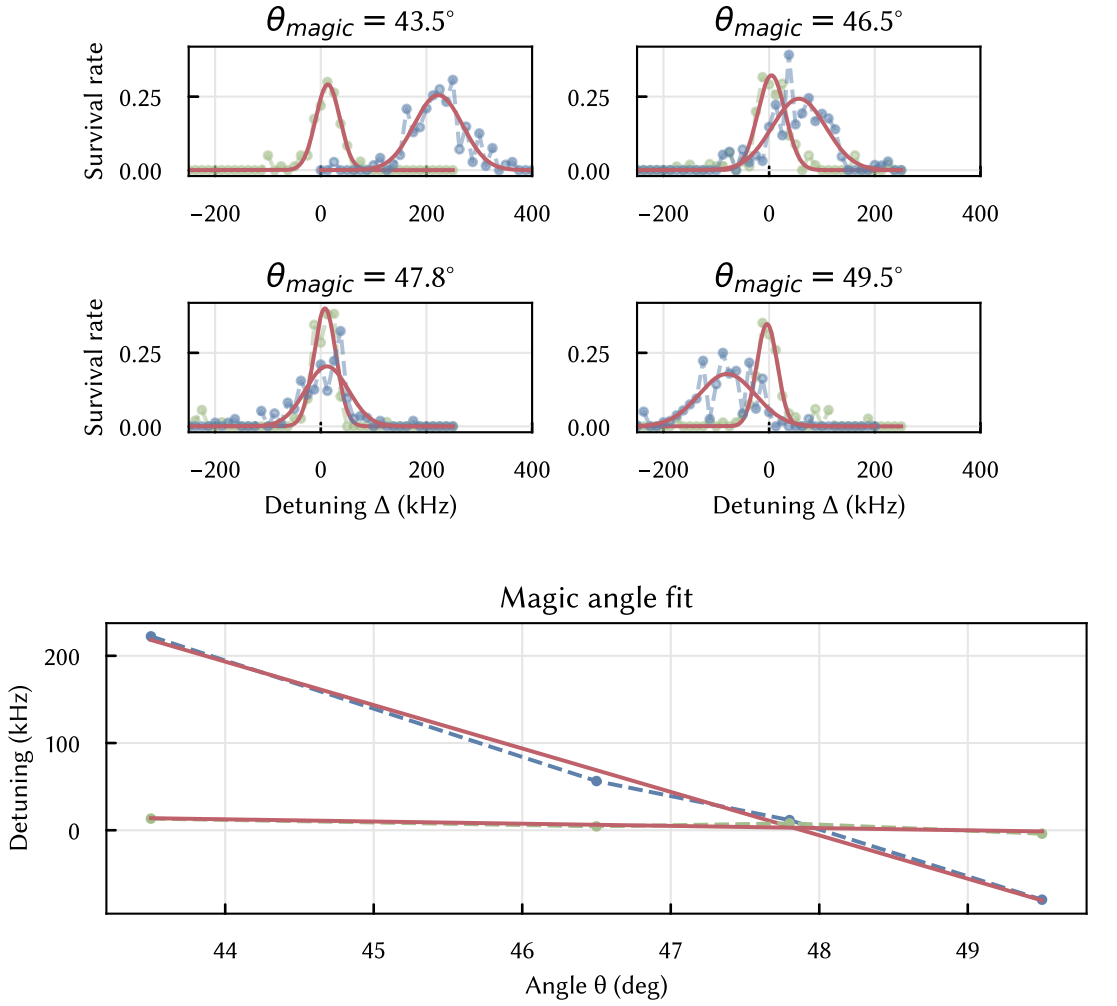


Figure 5.10: **Measurement of the magic angle.** The light shift of the $^3P_1$ $m_F = 1$ sublevel is measured for different angles of the magnetic field and two different trap depths. The magic angle is where there is no differential light shift, which is where the resonance frequency in the low and high tweezers match.

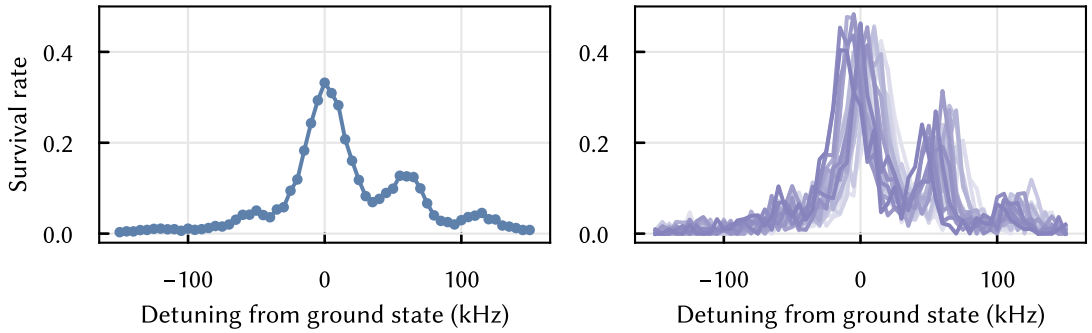


Figure 5.11: **Sideband spectrum on $^3P_1$.** The sideband spectrum is measured by shelving atoms into the $^3P_1$ state for different detunings. When atoms are resonant with a change in the vibrational sublevel of n, n+1, or n+2, they appear as three distinct peaks in the data. The left image shows a mean over all tweezers and the right shows each tweezer separately, clearly marking a non-perfect magic angle.

red sideband, we come across resonances whenever the frequency matches the energy difference for integer Δn. Most atoms will be susceptible to excitations of $\Delta n = 1$, but then only atoms not in the ground state are susceptible to excitations of $\Delta n = 2$ and so on. After atoms are shelved to the $^3P_1$ state, a push-out pulse (e.g., the imaging beam) removes residual atoms in the $^1S_0$ state. Letting the atoms decay back to the ground state and acquiring an image results in the outcome shown in Figure 5.11, both averaged over all tweezers and tweezer-selective. On the latter plot, it is clear that there is still a differential light shift between the tweezers. As the shelving pulse length is much larger than the lifetime of the $^3P_1$ state, at most 50% of atoms can be detected. The $\Delta n = 1$ peak is close, however, due to the decay, the steady state is less than 50%. Another factor is that vibrational levels are not evenly spaced, so for high n, the frequency no longer matches.

Using the plot, the separation of the peaks yields the axial trap frequency $f_{\mathrm{trap}} = 58.9$ kHz. Moreover, analyzing the area difference between the red and blue sideband results in the average occupation $\overline{n} = \left(A_{\mathrm{red}}\, A_{\mathrm{blue}}^{-1} - 1\right)^{-1} = 0.21$ resulting in a ground state occupation probability of $p_0 = (1 + \overline{n})^{-1} = 83\%$.

5.8. Verifying the trap frequency

We found the axial trap frequency using the sideband spectrum in the previous chapter. To verify that this frequency is correct, we do two additional measurements: a release and recapture measurement and parametric modulation spectroscopy.

5.8.1. Parametric modulation spectroscopy

In this measurement, the tweezers oscillate between a high and low power with a given frequency. If the frequency matches twice the trap frequency [119,120], then atoms are lost. This procedure yields both the axial and longitudinal trap frequency. The trap frequency is modulated, with the tweezer power quickly jumping between high and low values n_{periods} times. There is then a short pause of 250 us, after which the sequence is repeated. For us, the values are $n_{\text{periods}} = 30$ and $n_{\text{repeats}} = 30$. The frequency is scanned over many experimental runs, and the result is shown in Figure 5.12. It shows three features, at ~40 kHz, ~120 kHz and ~220 kHz, which refers to trap frequencies ~20 kHz, ~ 60 kHz and ~ 110 kHz. We can identify the feature at 60 kHz with the result from the sideband cooling spectrum, which was $f_{\text{trap}} = 58.9$ kHz.

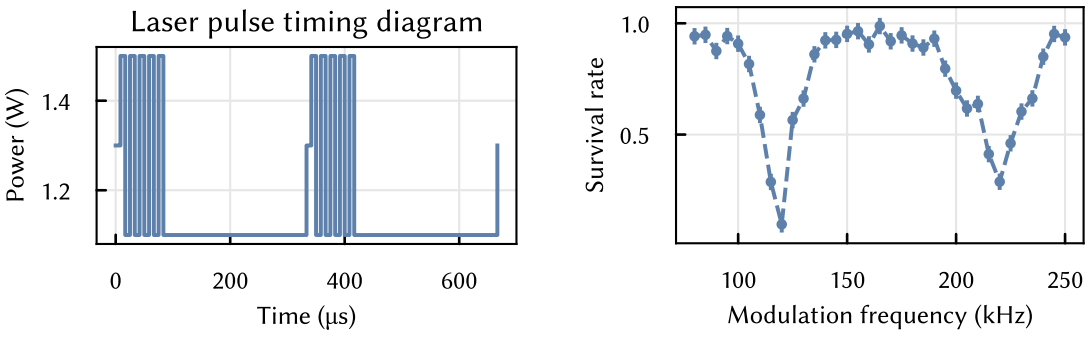


Figure 5.12: **Parametric modulation spectroscopy.** The trap oscillates at the given modulation frequency. If the trap oscillates at two times the trap frequency, atoms are heated out, resulting in measured trap frequencies of 60 kHz and 110 kHz. The left plot shows schematically how the tweezer power is modulated. Here $n_{\text{periods}} = 5, n_{\text{repeats}} = 2$ according to the definition in the main text.

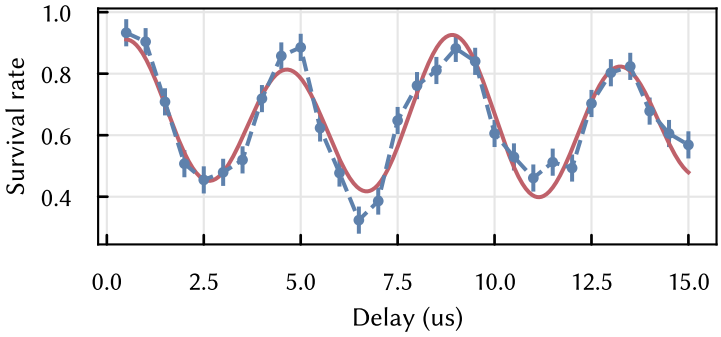


Figure 5.13: **Release and recapture for trap frequency.** Measuring the trap frequency by releasing atoms in tweezers and recapturing them. The fit in red is a sum of two sines and gives frequencies of 235 kHz and 135 kHz with amplitudes 0.22 and 0.06, respectively, resulting in trap frequencies of 117 kHz and 67 kHz, respectively.

5.8.2. Release and recapture

The second method for verifying the trap frequency is a release and recapture measurement. Here, the traps are turned off and then turned back on after a delay. Once again, the trap frequency is half of the measured frequency. The resulting survival rate is shown in Figure 5.13, and the fit agrees with the measurement from the parametric modulation spectroscopy.

5.9. Characterizing the trap depth

The trap depth is a fundamental parameter in the experiment affecting non-magic states and the lifetime of the atoms. Moreover, characterizing the actual trap depth is a tool to understand losses of the tweezer light before the vacuum cell. To measure the trap depth, we use the non-magic state $^3P_1$, which conveniently is used to cool atoms during the Sisyphus cooling measurement and sideband cooling. The power of the tweezer light before the AOD is measured, and a spectrum of the cooling feature is taken before the initial and final image. The resonance frequency in the absence of an electromagnetic field for $^1S_0 \rightarrow {}^3P_1$ is determined when the trap depth is zero. This can be extrapolated by measuring the cooling feature for various tweezer powers. We assume a tweezer waist of 750 nm (which was the designed waist) and taking theoretical values for polarizabilities gives the result in Figure 5.14.

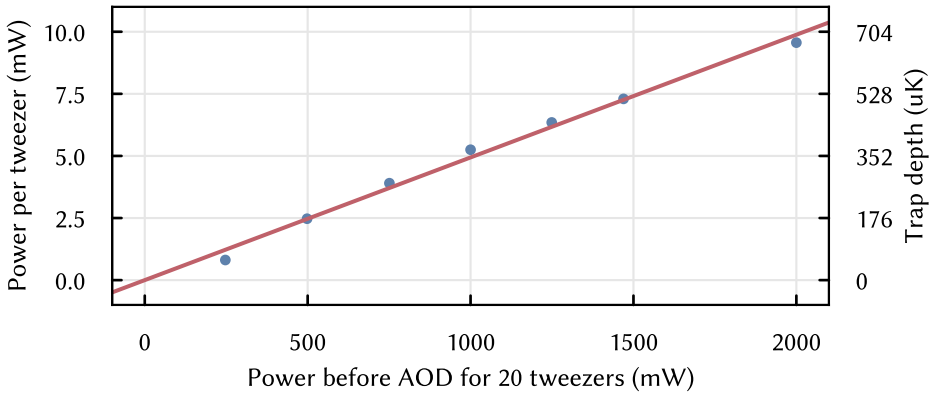


Figure 5.14: **Tweezer power calibration.** We calibrate the tweezer power to understand the resulting trap depth and losses at the atoms. The curve was measured by taking a spectrum of the $^1S_0 \rightarrow {}^3P_1$ cooling transition for different input powers. The resulting frequency is mapped to a light shift calculated from a waist of 750 nm and polarizabilities of 276.6 au and 355.0 au for the ground and excited state, respectively.

Examining the result, we can see that only 10% of the power before the AOD arrives at the atoms. Although we did not note down the numbers back then, the beam into the AODs was aligned based on diffraction efficiency. Measurements from a later point revealed that diffraction efficiencies were around 90% for the first and 60% for the second AOD, totalling a 46% power loss. This means there had to be another factor to explain the power loss. We later discovered that the collimated beam before the objective was not well-centered, which also affected cross-talk between tweezer sites, as the clipping on the objective creates a strongly diffracted beam in the focal plane.

5.10. Lifetime of atoms in optical tweezers

There was much effort involved in designing the vacuum system, where new insights were developed, such as in the choice of material and modeling the system with MoFlow to find an optimal solution. We want to use the atoms to verify that the vacuum is at a very low pressure in the glass cell itself, and for this, we measure the lifetime of the atoms in the optical tweezers at a trap depth of approx. 700 µK per tweezer (see Section 5.9). The measurement involves performing Sisyphus and sideband cooling, then allowing the experiment to idle for a set amount of time, which was not intensively optimized. We record a 1/e lifetime of $\tau = 24.6$ min across all tweezers and $\tau_{\text{row}} = 27.4$ min when

considering only one of the two rows. The results are shown in Figure 5.15. This result already surpasses the current record for non-cryogenic experiments [34]. There are plans to improve on this result, as we currently expect some heating from non-optimal tweezers. Also, more effective cooling during idle times can be implemented, such as pulsed cooling, which results in less heating due to phase noise in the laser compared to continuous cooling.

The result can be used to estimate the background pressure in the vacuum, which mainly consists of H_2 and He. At room temperature, these stray atoms and molecules have enough momentum to knock atoms out of a 700 µK deep trap; therefore, we can assume that every collision results in loss. Freely moving atoms collide with anything in their path, given the atom's radius, which a collisional cylinder can describe. The velocity and a time window give the height of the cylinder:

$$V = \pi r^2 v \Delta t \tag{5.9}$$

The collisional frequency of these background atoms is then calculated using the number density $\rho = \frac{N}{V} = \frac{p}{k_B T}$ within the time window:

$$Z = \frac{V\rho}{\Delta t} = \pi r^2 v \rho = \sigma v \rho = \sigma v \frac{p}{k_B T} \tag{5.11}$$

Since we can safely assume the Strontium atom as a stationary target, we can use literature values for the collision cross section between cesium and the background gas, $\sigma_{\mathrm{Cs}} = 2 \times 10^{-14}\,\mathrm{cm}^2$ [121]. As the collision cross section is proportional to the radius squared, we can infer an approximate value for Strontium:

$$\sigma_{\mathrm{Sr}} = \left(\frac{r_{\mathrm{Sr}}}{r_{\mathrm{Cs}}}\right)^2 \sigma_{\mathrm{Cs}} \approx 0.59 \sigma_{\mathrm{Cs}} \tag{5.13}$$

Then the collision frequency is related to the lifetime: $Z_i = \frac{1}{\tau}$. The mean velocity is calculated from the Maxwell-Boltzmann distribution:

$$v = \sqrt{\frac{2 k_B T}{m}} \tag{5.15}$$

and finally, the background pressure is calculated as:

$$p = \frac{k_B T}{\sigma_{\mathrm{Sr}} v \tau} \tag{5.17}$$

Using values from the experiment yields pressures $p \approx 3.6 \times 10^{-12}\,\mathrm{mbar}$, when accounting for both H_2 and He, which is consistent with our estimates from Section 2.5.

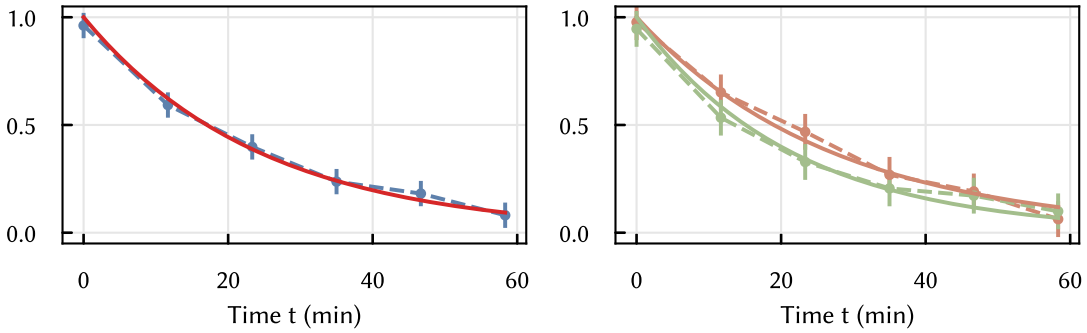


Figure 5.15: **Atomic vacuum lifetime.** The lifetime of the atoms in the tweezers was measured for initial Sisyphus and sideband cooling, then letting the experiment idle. The left plot shows averages over all tweezers, the right plot shows averages over the two rows separately. The $1/e$ lifetime for all averages is $\tau = 24.6$ min, the best row gives a lifetime of $\tau_{\text{row}} = 27.4$ min.

6. Effect of motion on qubit coherence

The cornerstone of this experiment is its ability to drive quantum gates used in a universal quantum computer, which contains both single-qubit and multi-qubit gates [15,36]. This platform encodes the qubits in the ground state $|0\rangle = {}^1S_0$ and the clock state $|1\rangle = {}^3P_0$. In this chapter, the focus is on describing the single-qubit interaction, creating superposition states, and highlighting the challenge of coherently transporting qubits. In a quantum computing algorithm, transport is an essential operation, directly enabling methods such as SWAP gates and designated quantum zones (such as for readout, storage, and experiment) [34,122]. However, this operation is challenging, as any non-adiabatic movement induces heating on the motional quantum states of the atom.

In the Lamb-Dicke regime, motional spread $\overline{n}$ is small to suppress coupling between motional and electronic states [57]. Therefore, even if transport does not directly affect the electronic state, operating outside the Lamb-Dicke regime reduces the fidelity of single-qubit operations, e.g., the fidelity of Rabi oscillations on the transition.

In the Lamb-Dicke regime, where the motional spread $\overline{n}$ is small, the coupling between motional and electronic states is suppressed. This ensures high-fidelity gate operations, as the Rabi frequency for the carrier transition remains close to its maximum value. However, if transport induces heating, the Rabi frequency is reduced, leading to lower gate fidelity and increased sensitivity to noise.

This chapter presents two physically motivated trajectories: the constant jerk and adiabatic sine. We theoretically and experimentally characterize their impact on motional excitation, focusing on how transport affects single-qubit operations. We present the fidelity of the single-qubit gate in a Rabi oscillation measurement, and its result is analyzed in a quantum computing context, where a randomized benchmarking (RB) measurement quantifies SPAM errors and gate errors separately.

While this chapter focuses on physically motivated trajectories, other approaches, such as optimal control techniques, are also possible. These methods are further discussed in the conclusion as a potential avenue for future work.

6.1. Motion basics for different trajectories

Moving atoms in optical tweezers is generally straightforward, as the atom will try to follow the trap's movement. The atom can be excited into higher motional states, up to a regime where it is ejected from the trap. Moreover, an open question remains: it is unclear if movement affects quantum algorithms by changing the internal atomic state.

Two commonly used trajectories discussed in the literature [34,38,123] are a constant jerk, where the derivative of the acceleration (the jerk) is kept constant, and an adiabatic sine trajectory, which forces the velocity and acceleration at the beginning and end of the trajectory to be zero.

The two waveforms are compared by their velocity, acceleration, and jerk, as well as the heating quantified by the change in motional state quantum number n, derived from [38,124]:

$$\Delta N = \frac{|\tilde{a}(\omega)|^2}{2\ \hbar\ \omega/m} \tag{6.2}$$

where $\tilde{a}(\omega_0)$ is the Fourier-transformed acceleration, ω the trap frequency and m the mass of the atom.

6.1.1. Constant jerk trajectory

The constant jerk ramp is motivated by classical motion, where the initial and final velocities shall be zero. This gives four initial conditions:

- $x(0) = x_0$
- $x(T) = x_0 + \Delta x$
- $v(0) = 0$
- $v(T) = 0$

With this, the equation for the position requires four free parameters; thus, the acceleration itself cannot be constant. The jerk (the first derivative of the acceleration) is kept constant, and position is then found by integrating the jerk four times:

$$\begin{aligned} j(t) &= \dddot{x}(t) = j_0 \\ a(t) &= \ddot{x}(t) = a_0 + j_0 t \\ v(t) &= \dot{x}(t) = v_0 + a_0 t + \frac{1}{2} j_0 t^2 \\ x(t) &= x_0 + v_0 t + \frac{1}{2} a_0 t^2 + \frac{1}{6} j_0 t^3 \end{aligned} \tag{6.4}$$

Applying the initial conditions results in the following values for the constants:

$$\begin{aligned} v_0 &= 0 \\ a_0 &= \frac{6\Delta x}{T^2} \\ j_0 &= -\frac{12\Delta x}{T^3} \end{aligned} \tag{6.6}$$

Therefore, using $\tau = t/T \in [0, 1]$ results in the following trajectory:

$$\begin{aligned} x(t) &= x_0 + 3\Delta x \frac{t^2}{T^2} - 2\Delta x \frac{t^3}{T^3} \\ x(\tau) &= x_0 + \Delta x[\tau^2(3 - 2\tau)] \end{aligned} \tag{6.8}$$

With the following equations for velocity, acceleration, and jerk:

$$\begin{aligned} \dot{x}(\tau) &= \frac{6\Delta x}{T}[\tau(1 - \tau)] \\ \ddot{x}(\tau) &= \frac{6\Delta x}{T^2}[1 - 2\tau] \\ \dddot{x}(\tau) &= -\frac{12\Delta x}{T^2} \end{aligned} \tag{6.10}$$

We can calculate initial and final, as well as average and maximum values for the velocity and acceleration (jerk is constant):

$$
\begin{aligned}
v_0 &= 0 \\
v_1 &= 0 \\
v_{\text{avg}} &= \langle \dot{x}(t) \rangle \\
&= \frac{6\Delta x}{T} \langle \tau(1-\tau) \rangle \\
&= \frac{6\Delta x}{T} \int_0^1 \mathrm{d}\ \tau\ \tau(1-\tau) \\
&= \frac{6\Delta x}{T} \left(\frac{1}{2} - \frac{1}{3} \right) \\
&= \frac{\Delta x}{T} \\
v_{\text{max}} &= \max\{\dot{x}(t)\} = \frac{3}{2}\frac{\Delta x}{T}
\end{aligned}
\tag{6.12}
$$

$$
\begin{aligned}
a_0 &= 6\frac{\Delta x}{T^2} \\
a_1 &= -6\frac{\Delta x}{T^2} \\
a_{\text{avg}} &= \langle \ddot{x}(t) \rangle = 0 \\
a_{\text{max}} &= \max\{\ddot{x}(t)\} = 6\frac{\Delta x}{T^3}
\end{aligned}
\tag{6.14}
$$

These results are tabulated in Table 6.1 for comparison with the adiabatic sine trajectory. To calculate the heating according to Equation (6.2), the Fourier transform of the acceleration is required:

$$
\begin{aligned}
\tilde{a}(\omega) &= \int_0^1 T\ \mathrm{d}\ \tau\ e^{i\omega T\tau} a_{\text{max}}(1-2\tau) \\
&= a_{\text{max}} T \int_0^1 \mathrm{d}\ \tau\ e^{i\omega T\tau}(1-2\tau) \\
&= a_{\text{max}} T \left[\left(\frac{1}{i\omega T} + \frac{1}{i\omega T}\tau + \frac{1}{(\omega T)^2} \right) e^{i\omega T\tau} \right]_0^1 \\
&= a_{\text{max}} T \left[\left(2\frac{1}{i\omega T} + \frac{1}{(\omega T)^2} \right) e^{i\omega T} - \left(\frac{1}{i\omega T} + \frac{1}{(\omega T)^2} \right) \right]
\end{aligned}
\tag{6.16}
$$

Since no further simplification is possible, the result is evaluated and plotted in Figure 6.2. The oscillatory term is visible in the result, suggesting that fast movements are possible in the correct parameter space. Classically, the atom oscillates in the trap

with the trap frequency, so moving the atom is an additional force, resulting in a driven harmonic oscillator. Then, the motion of the trap and the atom can oscillate constructively, severely heating the atom.

The regime $\omega T \gg 1$ with the trap frequency on the order of $\omega \approx 100\,\mathrm{kHz}$, such that $T \gg 10\,\mu\mathrm{s}$ simplifies the expression. Here, only the $i\omega T$ terms remain, then:

$$\tilde{a}(\omega)_{\omega T \gg 1} = \frac{a_{\max}}{i\omega}(2e^{i\omega T} - 1) \tag{6.18}$$

Here, the oscillatory term still has a significant contribution, which is also visible in Figure 6.2, unless very conservative movement durations are considered. Long movement times must be carefully weighed against the decay of coherence or atomic lifetime, since longer durations for quantum algorithms can lead to more uncertain results. Averaging out the oscillatory term by assuming a spread of trap frequencies [38] is not viable, as quantum computing applications require similar performance for all qubits, and hence it is undesirable to have a large spread in trap frequencies in the first place, which also has implications for cooling performance.

6.1.2. Adiabatic sine trajectory

The constant jerk trajectory starts with zero velocity, then maximally accelerates the atom and constantly decelerates until the atom is at rest. This trajectory may seem counterintuitive, as it means a huge force spike at the beginning. An alternative trajectory is the adiabatic sine, which has a ramp both in velocity and acceleration [34]:

$$x(\tau) = x_0 + \Delta x\left[\tau - \frac{1}{2\pi}\sin(2\pi\tau)\right] \tag{6.20}$$

where $\tau = t/T \in [0, 1]$. This gives the following equations for velocity, acceleration, and jerk:

$$\begin{aligned}
\dot{x}(\tau) &= \frac{\Delta x}{T}[1 - \cos(2\pi\tau)] \\
\ddot{x}(\tau) &= 2\pi\frac{\Delta x}{T^2}\sin(2\pi\tau) \\
\dddot{x}(\tau) &= 4\pi^2\frac{\Delta x}{T^3}\cos(2\pi\tau)
\end{aligned} \tag{6.22}$$

Contrary to the constant jerk, all initial and final values for velocity and acceleration are zero:

$$v_0 = v_1 = a_0 = a_1 = 0 \tag{6.23}$$

However, as the jerk is not constant this time, its initial and final values are:

$$j_0 = j_1 = 4\pi^2 \frac{\Delta x}{T^3} \tag{6.24}$$

Velocity is the only component with an average value:

$$\begin{aligned} v_{\text{avg}} &= \frac{\Delta x}{T} \\ a_{\text{avg}} &= 0 \\ j_{\text{avg}} &= 0 \end{aligned} \tag{6.25}$$

and the maximum values are:

$$\begin{aligned} v_{\max} &= 2\frac{\Delta x}{T} \\ a_{\max} &= 2\pi \frac{\Delta x}{T^2} \\ j_{\max} &= 4\pi^2 \frac{\Delta x}{T^3} \end{aligned} \tag{6.26}$$

To calculate the heating rate, the Fourier transformation of the acceleration needs to be calculated Equation (6.2):

$$\begin{aligned} \tilde{a}(\omega) &= \int_0^1 T \ \mathrm{d}\ \tau \ e^{i\omega T\tau} 2\pi \frac{\Delta x}{T^2} \sin(2\pi\tau) \\ &= 2\pi \frac{\Delta x}{T} \int_0^1 \mathrm{d}\ \tau \ e^{i\omega T\tau} \sin(2\pi\tau) \\ &= 2\pi \frac{\Delta x}{T} \left[2\pi \frac{1 - e^{i\omega T}}{4\pi^2 - \omega^2 T^2} \right] \\ &= 4\pi^2 \frac{\Delta x}{T} \frac{1 - e^{i\omega T}}{4\pi^2 - \omega^2 T^2} \end{aligned} \tag{6.28}$$

Note, that even when $\omega \to \frac{2\pi}{T}$, $\tilde{a}(\omega)$ remains finite:

$$\begin{aligned}\lim_{\omega\to 2\pi/T}\tilde{a}(\omega) &= \lim_{\omega\to 2\pi/T} 4\pi^2\frac{\Delta x}{T}\frac{1-e^{i\omega T}}{4\pi^2-\omega^2T^2}\\ &\overset{(*)}{=} \lim_{\omega\to 2\pi/T} 4\pi^2\frac{\Delta x}{T}\frac{iTe^{i\omega T}}{2\omega T^2}\\ &= 4\pi^2\frac{\Delta x}{T}\frac{i}{4\pi}\\ &= \frac{i\pi\Delta x}{T}\end{aligned} \tag{6.30}$$

where at $(*)$, we took the derivative of ω in the numerator and denominator according to L'Hôpital's rule.

6.1.3. Comparison between trajectory methods

For transporting atoms, two trajectories were discussed: The constant jerk and the adiabatic sine, displayed in Figure 6.1. Both start from zero velocity, accelerate, and then decelerate the atoms. Tabular data for velocity, acceleration, and jerk are summarized in Table 6.1. The significant difference is that the adiabatic sine trajectory has no initial and final acceleration, whereas the constant jerk trajectory starts applying a force on the atom immediately. The jerk in the adiabatic sine is about three times higher, which is to be expected, since the acceleration needs to be ramped up at the start. Also, even though the jerk in the constant jerk trajectory is constant, there is a delta-spike at $t = 0$ and $t = T$, where the acceleration jumps from 0 to the jerk value.

The theoretical picture evaluated for both trajectories is depicted in Figure 6.2, which looks at the increase of harmonic oscillator levels in the motional states of the trap. It averages over a normally distributed sample of trap frequencies in $[97.5\ \text{kHz}, 102.5\ \text{kHz}]$. Minima in heating for transport parameters are visible, due to the oscillatory behaviour, which is connected to the duration of the move and the trap frequency. This effect can be understood in a semiclassical picture, where the atom oscillates in the trap potential. The acceleration of the transport induces a force on the atom, such that when the Fourier component of the time-dependent force overlaps with the trap frequency, heating occurs as in a driven harmonic oscillator. The minima between the different harmonics of the trap frequency are then optimal parameters for movement.

Moreover, a clear difference between parametrizing with the maximum acceleration $a_{\max}$ and the duration T is visible. For long durations ($> 100\ \mu\text{s}$) or low accelerations ($< 0.02\ \mu\text{m}\ \mu\text{s}^{-2}$), both trajectories are comparable. However, while the constant jerk excels for short distances and short durations, the adiabatic sine appears to perform better at

Constant jerk

Parameter	Initial value	Final value	Mean	Max
Velocity	0	0	$\Delta x/T$	3/2 $\Delta x/T$
Acceleration	$6\Delta x/T^2$	$-6\Delta x/T^2$	0	$6\Delta x/T^2$
Jerk	$-12\Delta x/T^3$	$-12\Delta x/T^3$	$-12\Delta x/T^3$	$-12\Delta x/T^3$

Adiabatic sine

Parameter	Initial value	Final value	Mean	Max
Velocity	0	0	$\Delta x/T$	2 $\Delta x/T$
Acceleration	0	0	0	$2\pi\Delta x/T^2$
Jerk	$4\pi^2\Delta x/T^3$	$4\pi^2\Delta x/T^3$	0	$4\pi^2\Delta x/T^3$

Table 6.1: **Summary of motion parameters for two different trajectory methods.** The most prominent difference is the initial and final acceleration, which is zero for the adiabatic sine, but not for the constant jerk.

high accelerations and high distances. A comparison between the two waveforms was verified in an experiment Figure 6.2, where the full atomic array was moved back and forth with different trajectories. Here, the array is generated with an SLM, which is preceded by a 2d-AOD. The atoms are moved 10 µm in one direction and then back the

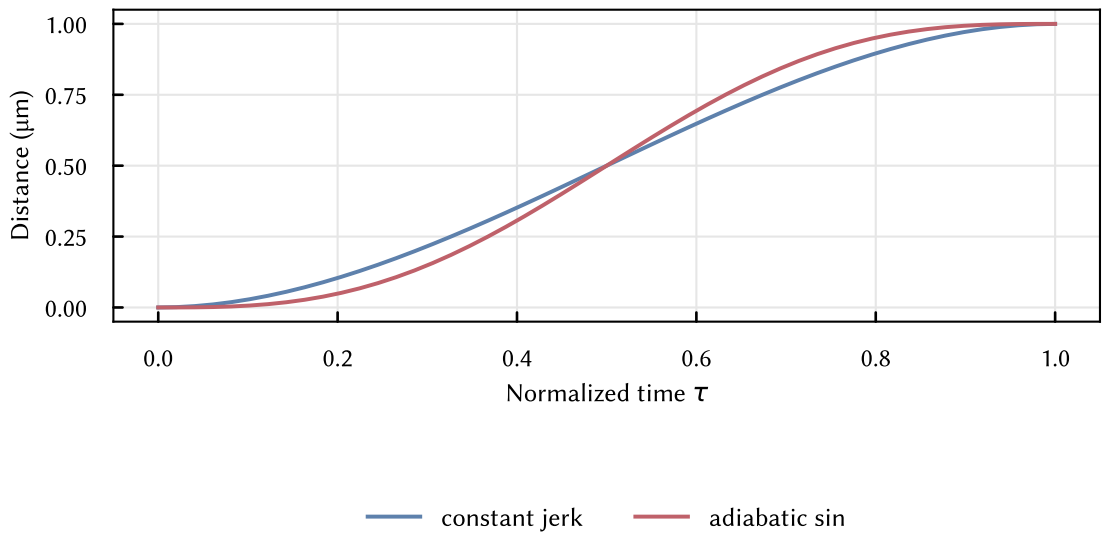


Figure 6.1: **Shape of tweezer transport trajectories.** The shape of the two waveforms, constant jerk and adiabatic sine, as a function of $\tau = t/T \in [0, 1]$.

same way to the origin. The data appears to fit the theory; unfortunately, comparison data where the movement is parametrized against $a_{\max}$ is not available. In another experiment, the data was extracted tweezer-resolved, parametrized with $a_{\max}$. Five tweezers were selected from lowest to highest total survival, and the features where the Fourier component of the transport matches the harmonics of the trap frequency are clearly visible.

With a trap frequency of 60 kHz and a trap depth of 700 μK, we expect 460 excitations to lose the atom (from $\frac{1}{2}k_B T = \delta n\ \hbar\ \omega$, where δn is the number of excitations). This reasoning agrees with the comparison to Figure 6.2 a), where the edge is right around five excitations, so 100 repetitions yield the same order of magnitude.

When the transport is used in a quantum algorithm as an operation, its fidelity impacts the result of the program. The heating effects explored here can be a leading cause of fidelity loss. Even if only motional states are excited and the electronic state remains unaffected, the wider spread in motional occupation $\overline{n}$ reduces the fidelity of single-qubit operations. The Rabi frequency depends on the Lamb-Dicke parameter $\eta = \frac{2\pi}{\lambda}\sqrt{\frac{\hbar}{2m\omega}}$ [125]:

$$\Omega_{n,n} = \Omega_0 e^{-\eta^2/2} L_n(\eta^2) \tag{6.31}$$

where λ is the frequency of the driving laser, m the mass of the atom, ω the trap frequency and $L_n(\eta^2)$ the Laguerre polynomial.

For quantum computing, the most crucial quantity is the fidelity of the single-qubit gate, which here is simply the excited state fraction after a π-pulse. After the atoms are initialized, they mostly occupy the motional ground state (in Figure 5.11, we measured $p_0 = 83\%$), so we can assume that the laser is tuned to the Rabi frequency of $n = 0$. The detuning of the higher states is, then

$$\Delta_n = \sqrt{\Omega_{0,0}^2 - \Omega_{n,n}^2} = \Omega_{0,0}\sqrt{1 - L_{n(\eta^2)}} \tag{6.32}$$

Our measured linewidth is approximately $\Gamma = 1$ kHz (see results of next chapter), so that the state $n = 3$ can already only be driven with a fidelity of 50%. We approximate the line shape of the $^1S_0 \rightarrow {}^3P_0$ transition as a gaussian $G(f)$ with linewidth $\Gamma = 1$ kHz centered at the transition frequency. This approximation is valid here, as the approximation only breaks for large detunings, where contributions are small. The fidelity $\mathcal{F}(\overline{n})$ is then calculated by summing over the contributions of each motional state n, weighed by the binomial occupation distribution $P_n(\overline{n}) = \overline{n}^n/(1+\overline{n})^{n+1}$ and the detuning Δ_n:

$$\mathcal{F}(\overline{n}) = \frac{1}{G(0)} \sum_n P_n(\overline{n}) G(\Delta_n) \tag{6.33}$$

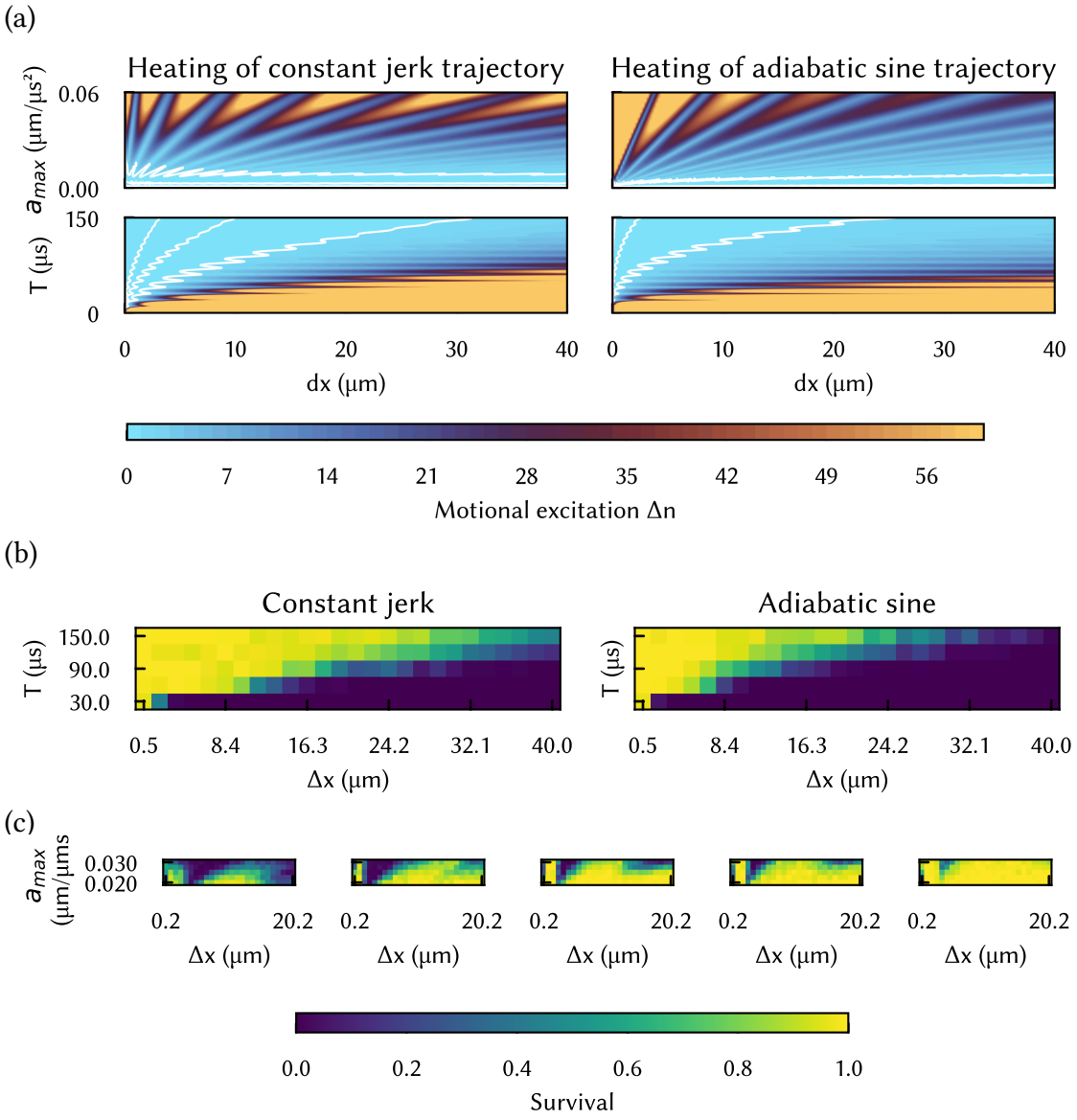


Figure 6.2: **Summary of heating of atoms during motion for different trajectory methods.** a) Theoretical calculations based on Equation (6.16) and Equation (6.28) with normally distributed trap frequencies $\in [97.5\,\text{kHz}, 102.5\,\text{kHz}]$. The trajectories are parametrised by $a_{\max}$, and the duration T. A diverging color scale was chosen to increase visibility where atoms may be lost. White contour lines are added for the $n = 1$, $n = 0.1$, and $n = 0.01$ levels. The threshold was placed to match the data in b). b) Experimental data for ground state movement for different trajectories. The movement was repeated 100 times, showing atom loss comparable to a). c) Tweezer resolved measurement similar to b), however, parametrized by $a_{\max}$. Here, five tweezers were selected from the lowest to the highest total survival.

Figure 6.3 shows the result of these calculations. We can see that the fidelity remains constant up to $\overline{n} \approx 1.5$, but then starts to decay. In Figure 6.2, comparing the experimental data with the theory, a change in $\Delta n = 0.01$ can be observed for long movement times. As such, even moving ~ 100 times during a quantum algorithm increases $\overline{n}$ to ~ 1.0, where the fidelity decay is predicted to be $< 2 \times 10^{-6}$ and likely dominated by the imaging fidelity or other loss contributions.

This result demonstrates that coherent transport can be integrated into quantum algorithms without significant fidelity loss, even for large-scale systems requiring hundreds of qubit movements. It is a critical step toward scalable, reconfigurable quantum computing architectures.

6.2. Coherent Clock Rabi oscillations

After gaining intuition on how transporting atoms affects their motional quantum state, we can investigate experimentally how transport affects qubit superposition states. As the atom is encoded in $|0\rangle = 1S0$ and $|1\rangle = 3P0$, the single-qubit gate is a Rabi transition between the two states. The transition is first quantified by measuring Rabi oscillations and spectroscopy, and the gate fidelity is extracted in a RB measurement. The qubit motion is then investigated using two different methods: First, by executing a move in the wait time during a Ramsey measurement and then in an IRB experiment, where the movement error can be extracted similarly to the single-qubit operation.

As described in Section 5.7, atoms are prepared in the electronic and vibrational ground state by cooling them in a sisyphus and sideband cooling sequence. Once there, it is possible to do high fidelity clock spectroscopy on the single-qubit transition $^1S_0 \rightarrow {}^3P_0$. The clock beam enters via the front surface of the glass cell and is a circular beam of ~ 15 mW. Details of the laser setup and the measurement will be part of a future thesis [126], therefore, the following is only a presentation of the results. We were able to identify the clock resonance by measuring the atomic survival rate after shelving the atoms to the $^3P_0$ state and pushing out the ground state atoms. During the imaging sequence, atoms are repumped from the $^3P_0$ to the $^3P_1$ state, where atoms decay back to the ground state, which fluoresce and therefore indicate atoms that were excited on the $^1S_0 \rightarrow {}^3P_0$ transition.

We measure a Rabi frequency of $\Omega/2\pi = 2.7\,\mathrm{kHz} \pm 3\,\mathrm{Hz}$ (fit error) and a raw π-pulse fidelity of $98.9\% \pm 0.5\%$ in March 2024. The fidelity is inferred from the spectroscopy in Figure 6.4, which uses the pi-time $\tau_\pi = 1/2\Omega$. The error bars are based on fits, taking

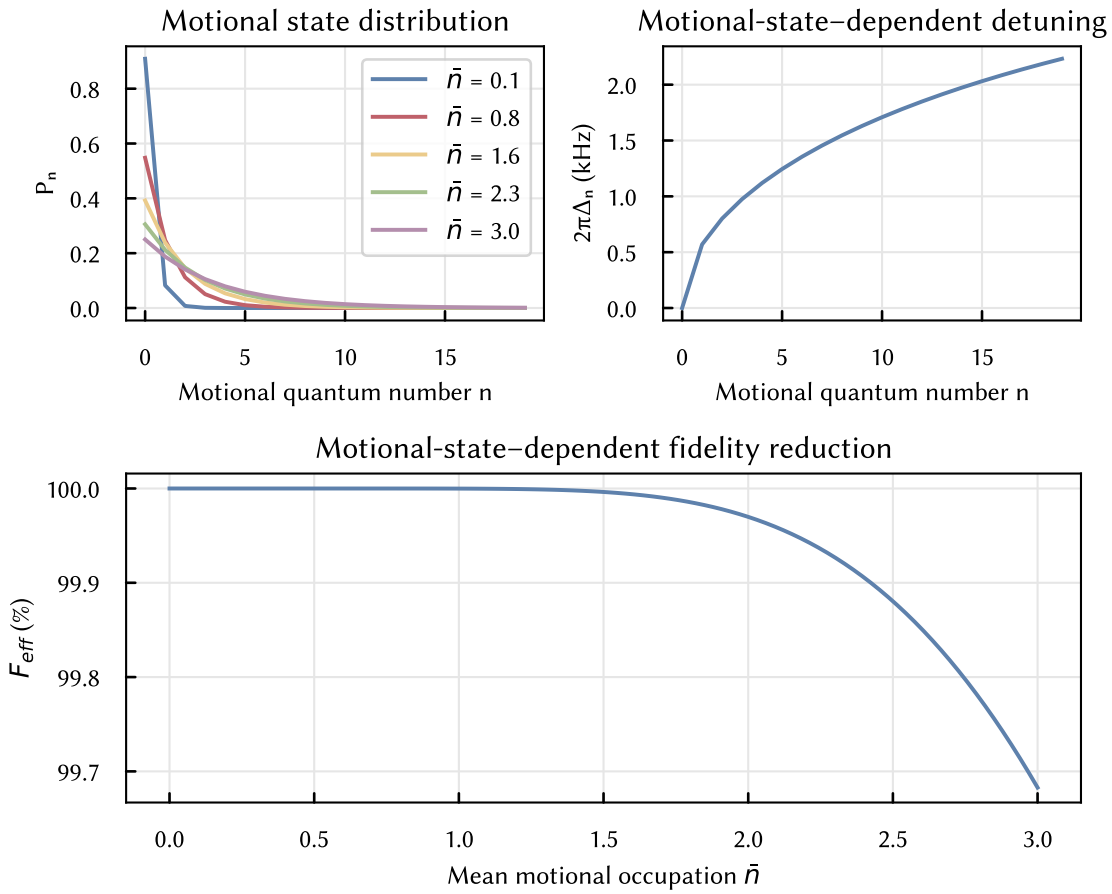


Figure 6.3: **Motional state dependence in quantum computing** During transport, atoms can be excited into higher motional states. Higher motional states require lower Rabi frequencies and thus higher detunings. Shown is the binomial state distribution P_n for $\overline{n} \in [0.1, 3.]$. The detuning is plotted for motional states up to 20, showing that states $n > 4$ already exceed the linewidth of the clock transition measured in this thesis. The reduction in π-pulse fidelity is a direct consequence of driving detuned Rabi oscillations on an ensemble of motional states P_n.

into account the quantum projection noise at $N_\Omega = 75$ shots per point for the Rabi oscillation and $N_\nu = 22$ shots per point for the spectroscopy.

The fidelity is lower than the current state-of-the-art for neutral atom quantum computing [15,127]. It is currently limited by beam pointing fluctuations, which the atoms experience as varying light intensity, resulting in an average of different Rabi frequencies. In general, the full potential of the transition has yet to be exhausted, and further analysis is conducted in the PhD thesis [126].

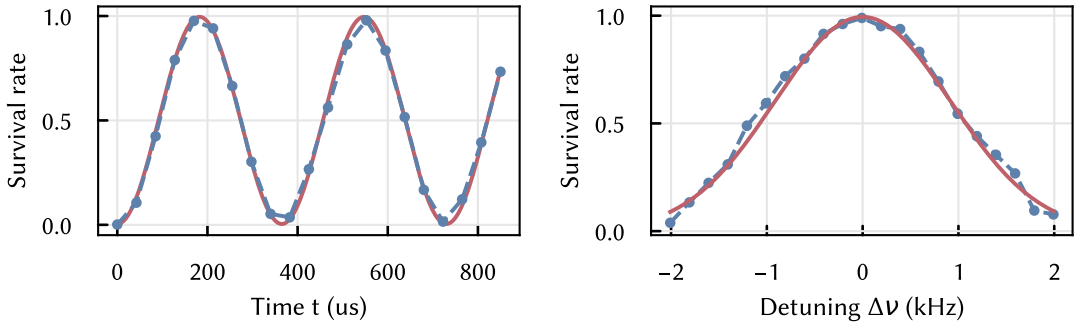


Figure 6.4: **Rabi oscillations of the clock state and spectroscopy on the $^1S_0 \rightarrow {}^3P_0$.** The Rabi frequency of the fit gives $\Omega = 2\pi \times 2.7\,\mathrm{kHz} \pm 3\,\mathrm{Hz}$ with a $\pi/2$ fidelity of $98.9\% \pm 0.5\%$. Error bars in the plots are smaller than the size of the markers.

These measurements were taken in the absence of motion, so atoms remain primarily in the motional ground state with a measured $\overline{n} = 0.21$. They provide a baseline for the following measurements, where superposition states are transported.

6.3. Coherent qubit transport

The effect of atom transport on the qubit manifold is best understood in two measurements. A Ramsey measurement quantifies the error on the superposition state $|s\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, where the loss of coherence, as well as phase errors, can be extracted. On the other hand, examining errors on a gate level can be accomplished in an IRB measurement. In any RB measurement, gate and SPAM errors are separated, where in the regular RB, the number of single-qubit gates is modified, and in the IRB case, the number of single-qubit gates is fixed and movement is interleaved instead. The movement error then comes up as the gate error.

6.3.1. Characterization in a Ramsey measurement

The first measurement is a Ramsey measurement with a movement operation during the dark time. The atoms are prepared in a superposition with a $\pi/2$ pulse, then they idle for a time T, and then the second $\pi/2$ pulse is executed with a varying phase $\varphi \in [0, 2\pi]$ to measure a Ramsey fringe.

The result is one revolution of a sine with an offset, an amplitude, and a phase φ:

$$f(x) = (1 - \mathcal{L}) * \frac{1 + C * \cos(x + \varphi)}{2} \tag{6.35}$$

$\mathcal{L}$ is the loss of atoms, e.g., due to heating, C is the contrast, which quantifies the coherence of the state, and φ is the phase shift. The atomic loss $\mathcal{L}$ should be the same as the ground state loss from above. However, if the contrast C changes when increasing the velocity or the acceleration of the move, then this relates directly to a gate error, which can not be compensated. A change in phase φ can be calibrated and compensated with a phase gate. The result of the measurement is shown in Figure 6.5. The dark time T was always set to fully contain the move, plus a small buffer to ensure there was no overlap. For this reason, there is a change in coherence and phase between different T; however, both values are constant when the distance Δx is varied, indicating no direct gate error, even though there is a change in the motional state number. The loss is comparable to Figure 6.2.

As we see in the measurement, the contrast remains constant, and as such, no significant gate error can be extracted. More importantly, the contrast remains constant even though there is substantial heating. In Figure 6.3, it was explored that motional heating must be significant to impact single-qubit gate fidelity, which agrees with the coherence results in the Ramsey measurement.

6.3.2. Interleaved randomized benchmarking

Randomized benchmarking has become the fundamental tool for evaluating and comparing quantum gate errors. At the current state of quantum computers, many areas are still being refined, and one issue is SPAM errors, which is a two-fold problem. For one, atoms may not be initialized in the desired state, which is often a more significant issue when the ground state has some hyperfine splitting, requiring atoms to be pumped into one of the sub-states. As the bosonic ^{88}Sr does not have this hyperfine splitting, it is possible to negate this error entirely. On the other hand, measurement errors occur when the qubit state is read out. In neutral atom quantum computers, this usually means imaging the atoms and measuring the photons. The resulting image needs to be analyzed for occupied lattice or tweezer sites by applying a threshold to a region of interest based on histograms, as shown in Figure 5.4.

In our detection scheme, atoms are imaged on the $^1S_0 \rightarrow {}^1P_1$ transition, where atoms have a low probability to decay into the $^1D_2$ state (see Figure 2.1). This decay hinders the detection, as we currently have no way of quickly repumping atoms from $^1D_2$ back to $^1P_1$ (where they would then decay back to the ground state).

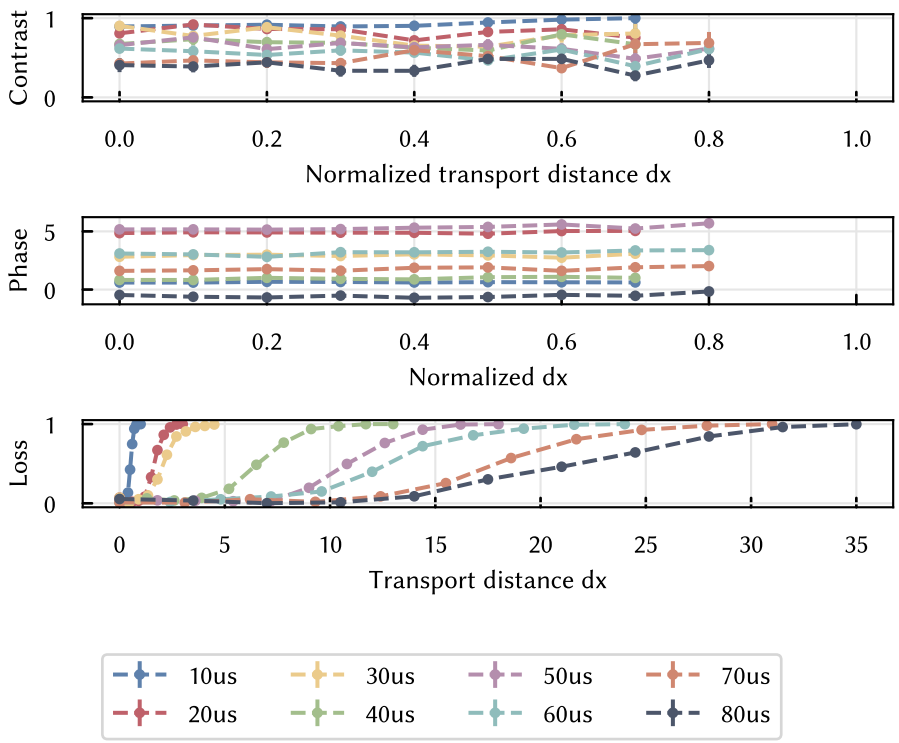


Figure 6.5: **Clock state coherence during transport in a Ramsey measurement.** Coherence of the clock state was investigated when the qubit was moved in a Ramsey sequence. Atoms were transported on the constant jerk trajectory during the free evolution time between $\frac{\pi}{2}$ pulses. The phase of the second pulse was used as the iteration parameter, allowing a Ramsey fringe to be taken. The contrast C, loss $\mathcal{L}$, and phase shift φ of the fringe were then extracted according to Equation (6.35). The duration of the dark time varied for different movement durations, causing the initial phase and contrast to differ. However, since it remains constant for different distances, we can conclude that transport has no marginal effect on the accumulated phase. Data points where loss is > 5% were ignored in the contrast and phase.

In this regard, being able to measure quantum gate error despite the presence of SPAM errors allows working on both issues simultaneously. Randomized benchmarking was first suggested in [128]. The algorithm works by defining a set of gates, which are applied randomly to the qubit. If the gates are in the Clifford group, this allows a full quantification of the quantum computing performance, since Clifford gates can scramble the state entirely on the Bloch sphere [129]. The benchmarking algorithm is defined with a length m, limiting the number of randomly applied gates. The final state of the atom needs to be read out to extract the error (which includes SPAM error) as a function of m.

In [128] a, the gate set are uniform operators from the Pauli group of the form $e^{\pm\sigma_b \pi/2}$ for π pulses and $e^{\pm\sigma_u \pi/4}$ for $\pi/2$ pulses, where $b = 0, x, y, z$ and $u = x, y$ randomly chosen and σ_0 is the identity operator.

Here, the qubit is encoded in the ground state $|0\rangle = 1S0$ and the clock state $|1\rangle = 3P0$, so single-qubit operations mean driving the transition with a single laser and its respective phase (φ). It is sufficient to implement $X(\theta)$ and $Y(\theta)$ rotations for arbitrary rotations on the Bloch sphere with the angle θ [130], since Z rotations are compositions of $X(\theta) + Y(\theta)$ gates. The difference between $X(\theta)$ and $Y(\theta)$ is a shift in laser frequency $\varphi \in [0, \pi]$, respectively. Thus, the computational gate set is $\{X(\pm\pi/2), Y(\pm\pi/2)\}$. For reading out the final state, the atom needs to be in an eigenstate of σ_z. For simplicity, we want the qubit to end up in $|1\rangle$, so the sequence ends with either zero, one, or two final gates selected from the gate set. This procedure is possible because the final state can be easily calculated based on the random selection of gates.

The survival in the RB is fitted to

$$\frac{1}{2} + \frac{1}{2}(1 - 2\varepsilon_{\text{spam}})(1 - 2\varepsilon_{\text{gate}})^N \tag{6.37}$$

according to [106], such that the fidelity is $\mathcal{F}_{\text{gate}} = 1 - \varepsilon_{\text{gate}}$. Figure 6.6 displays the result of our measurement, resulting in a spam error $\varepsilon_{\text{spam}} = 3.77\% \pm 0.14\%$, gate error $\varepsilon_{\text{gate}} = 0.126\% \pm 0.006\%$, and thus a fidelity $\mathcal{F}_{\text{gate}} = 99.874\% \pm 0.006\%$, where errors are based on the uncertainty of the fit. The gate error is dominated by imaging fidelity, which was measured as $\mathcal{F}_{\text{imaging}} = 98.97\%$ and is caused by atoms decaying into $^1D_2$ as explained above.

Randomised benchmarking uses a random single-qubit operation to explore gate errors on the whole Bloch sphere. Similarly, motional gate errors can be extracted in a IRB measurement. The single-qubit gates are randomised, but their total number remains fixed. A waiting time τ is introduced in between single-qubit gates, where atoms can be transported back and forth, effectively acting as an identity gate. The total number of transports is then a parameter that is scanned in the experiment, so that Equation (6.37) is used, where the offset then includes both SPAM and single-qubit gate errors.

Figure 6.7 shows the result for the experiment. We were interested in the reduction of the gate fidelity based on the distance Δx. However, to obtain meaningful results for randomized benchmarking, the moves themselves could not cause loss. Thus, the survival of atoms was measured without single-qubit gates, and the sequence only had the maximum number of moves. The survival was measured against movement duration, and the duration was picked where survival converged to a maximum (which is effectively only imaging survival at that point). This results in varying velocities

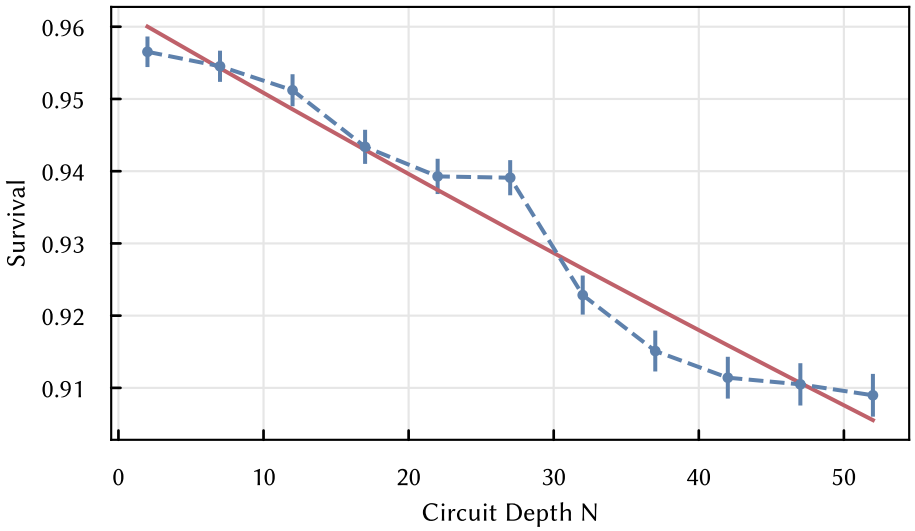


Figure 6.6: **Randomized benchmarking on clock transition.** A randomized benchmarking measurement was taken for the natural gate set $\{X(\pm\pi/2), Y(\pm\pi/2)\}$ according to the methods in the main text. The resulting gate fidelity is $\mathcal{F}_{\text{gate}} = 99.874 \pm 0.006$ with a spam error of $\varepsilon_{\text{spam}} = 3.77\% \pm 0.14\%$.

across the IRB measurement, which clearly correlate to the resulting gate fidelity. The results paint a different picture compared to the Ramsey measurements. Even if the measurement is taken in a regime where no coherence or heating loss is registered, a significant gate error is still present. Of the values recorded, a 1 µm move at 40 µs duration still results in a $\varepsilon_{\text{movement}} = 0.17\% \pm 0.06\%$ movement gate error. It is important to understand the exact loss mechanisms in the measurement to interpret the results. For quantum algorithms, it should be possible to move atoms coherently, preserving their quantum state. In this measurement, however, atom loss can still be part of the result, and the dependence on acceleration suggests this as well. The atom's state is measured by pushing out ground-state atoms. Therefore, taking the same measurement with and without a push-out pulse will result in a value for loss due to movement.

The movement gate error $\varepsilon_{\text{movement}} = 0.17\% \pm 0.06\%$ is comparable to the single-qubit gate error $\varepsilon_{\text{clock}} = 0.13\% \pm 0.006\%$, suggesting that transport is a viable operation in a quantum algorithm. Even though Figure 6.3 shows that significant heating is necessary to reduce the single-qubit gate fidelity, a strong dependence of the IRB measurement on the instantaneous acceleration is visible. Moreover, if motional heating affects single-qubit gate fidelity, then the movement gate error from Figure 6.7 can not be separated sufficiently from single-qubit gate errors, as they depend on the number of preceding moves.

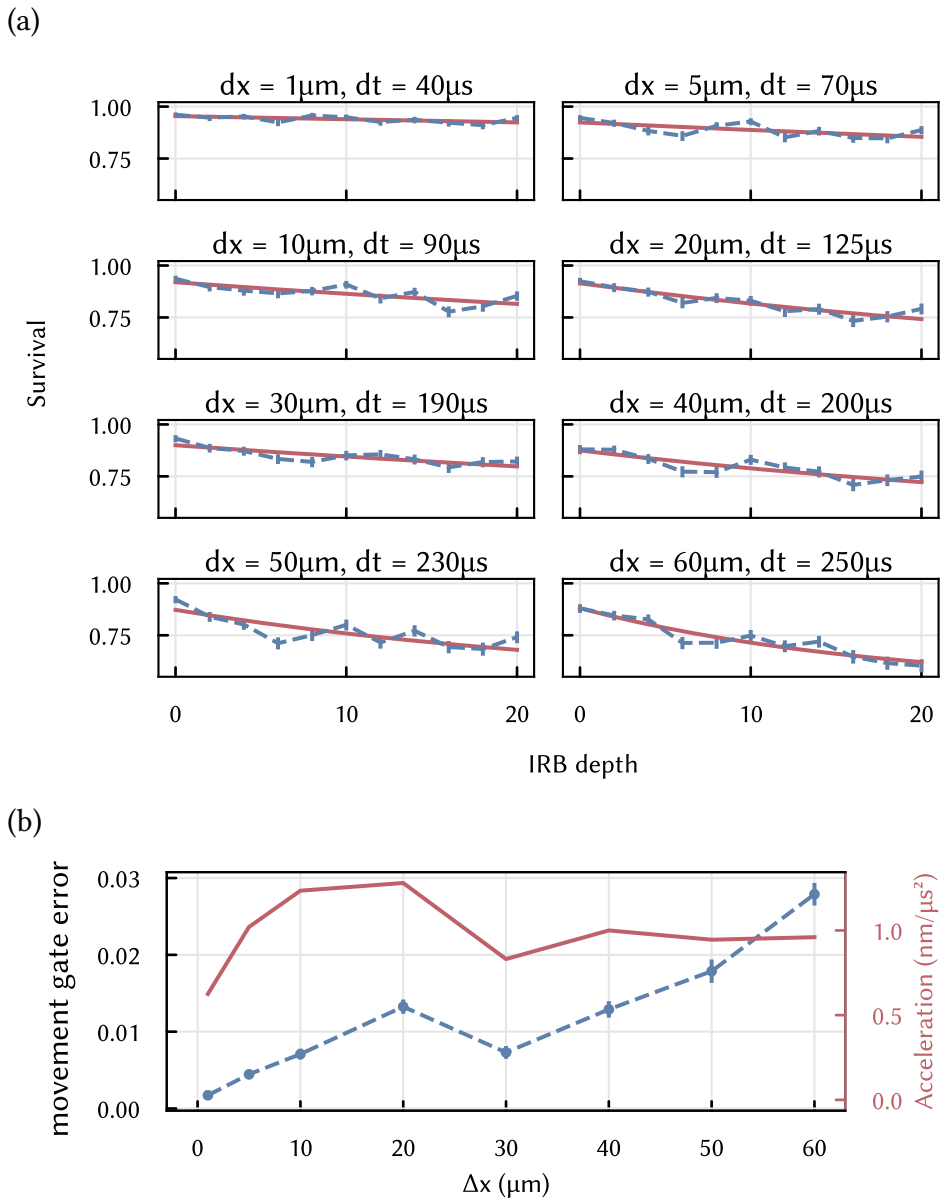


Figure 6.7: **Interleaved randomized benchmarking for varying transport distances.** To quantify the effect of movement on atoms during a quantum algorithm, an interleaved randomized benchmarking measurement was performed. The natural gate set was fixed to 25 gates in total; the number of interleaved moves varied across experiments and is represented on the x-axis of the plots on the left side. The movement and spam errors are extracted. The gate errors of the single-qubit gates are part of the spam error, and this value is instead extracted from Figure 6.6. The movement errors are plotted for different distances, which also shows a clear correlation to the instantaneous acceleration of the move. The distance and duration of a move are given for one direction; however, the full move is a back-and-forth movement.

The constant jerk and adiabatic sine trajectories presented here serve as a fundamental starting point for designing trajectories that minimize motional heating during atom transport. While these trajectories already demonstrate low heating and low gate errors, further improvements can be achieved through optimal control techniques. Optimal control methods, as explored in [49,123], use pre-defined metrics to tailor the transport fidelity for specific goals, such as reducing motional heating or maximizing gate fidelity.

In these approaches, the trajectory contains free parameters, such as higher-order polynomials, and is dynamically optimized based on experimental results. For example, a metric here can be the gate fidelity extracted by the IRB measurement, which helps refine the trajectory to minimize gate errors and maintain high transport times. Ultimately, combining the insights from the trajectories presented here with optimal control techniques offers a powerful pathway to minimize movement errors and unlock the full potential of reconfigurable quantum computing architectures.

7. Software for neutral atom experiments

Even though quantum computers excel at specific calculations, their naming suggests they are a functional alternative to classical computers. That, however, is not true, and in fact, quantum computers have to rely on classical machines to interact with the qubits. For a neutral atom quantum computer, many devices are used to control various parts of the experiment, including laser control software, PID loops for stabilizing lasers or magnetic fields, precise timing hardware, and individual pulse control. Some of these problems can be solved using dedicated electrical hardware; however, the following chapters will highlight where it is important to have classical computing with good software.

7.1. Hardware for precise timing

In this experiment, we opted to use the ARTIQ software stack together with the Sinara hardware [131]. Other popular timing systems include, for example, the ADWIN real-time system [132] or National Instruments PXIe cards. Both the Sinara hardware and the ARTIQ software are open source and are already used in ion quantum computing experiments.

The main timing controller of the Sinara ecosystem is the Kasli [133]. It can be used in a primary/secondary architecture, where the primary device handles the timing and delegates commands down to the secondary device. We make use of digital input / output (DI/O) cards, analog output cards, and DDS cards. The crates are then configured as follows:

1. 1x Kasli 2.0 + 24x SMA-TTL
2. 1x Kasli 2.0 + 2x Fastino
3. 1x Kasli 2.0 + 1x Clocker + 6x Urukul

There are even more options available, for example AWGs or cards for PDHs, so it is conceivable to run most of the experiment from this system; however, we opted to use specialized or custom hardware for other tasks that are not directly linked to the timing system.

7.2. Driving the hardware

The ARTIQ software stack consists of a Python frontend, where the experiment control structure is written, and a Rust backend, which runs on the real-time FPGA of the Kasli. We adapt the ARTIQ infrastructure by inheriting from `artiq.experiment.EnvExperiment`, which serves as the base class for starting an experiment. The scheduler has access to the instance's `build` and `run` functions, where `build` runs on the host machine and can perform all software-side initializations (such as creating new class instances and allocating memory). On the other hand, `run` is compiled and then executed on the FPGA. Functions running on the FPGA are called "kernel" functions and are thus decorated using `@artiq.experiment.kernel`.

Even though `build` is called first, `run` will be the main entrance point of the experiment. ARTIQ uses remote procedure calls (RPCs) to communicate between the kernel and the host; however, only kernel functions can call host functions (but they can return values). By default, the kernel will wait for the host function to finish. If that is not desired, they can be marked to run asynchronously by using the decorator `@artiq.experiment.rpc(flags={"async"})`.

7.3. The software backend

The control system includes functionality for managing data, servers for communicating with the experiment, definitions of hardware-specific interfaces, and a main loop to run the experiment. The control system is organized according to Listing 7.1, and the flow is shown in Figure 7.1. The main entrance point is `server.py`, which starts various servers used to interface with the control system. The `experiment.py` file defines the base class `MQVAExperiment`, which the user uses to write their experiment. The `user_channels` directory contains definitions for various interfaces into hardware, and finally, a `parameter_manager` is used to handle all iterable parameters in a physics experiment.

```
├── mqva_control
│   ├── config.py
│   ├── experiment.py
│   ├── parameter_manager
│   │   ├── __init__.py
│   │   ├── nested_dict.py
│   │   ├── parameter_manager.py
│   │   └── parameter_watchdog.py
│   ├── scheduler.py
│   ├── server.py
│   ├── submit.py
│   ├── tools.py
│   └── user_channels
│       ├── artiq_modules
│       │   ├── analog_calibrated.py
│       │   ├── fastino_wrapper.py
│       │   ├── ttl_wrapper.py
│       │   └── urukul_wrapper.py
│       ├── cameras
│       │   ├── manta_camera.py
│       │   ├── orca_camera.py
│       │   └── ORCA_lib
│       │       ├── dcamapi4.py
│       │       ├── dcam.py
│       │       └── orca_client.py
│       ├── coils
│       │   ├── caenels_power_supply.py
│       │   └── highfield_coil.py
│       ├── dummy_wrapper.py
│       ├── __init__.py
│       ├── test.py
│       ├── user_channel.py
│       ├── visa_devices
│       │   ├── rigol.py
│       │   └── srs_fg382.py
│       ├── waveform.py
│       ├── wieserlabs_wrapper.py
│       └── xilinx_iis_board.py
```

Listing 7.1: **mqva_control folder structure.** The software stack has folders for the parameter manager and custom user channels. All other utility classes are defined in the root.

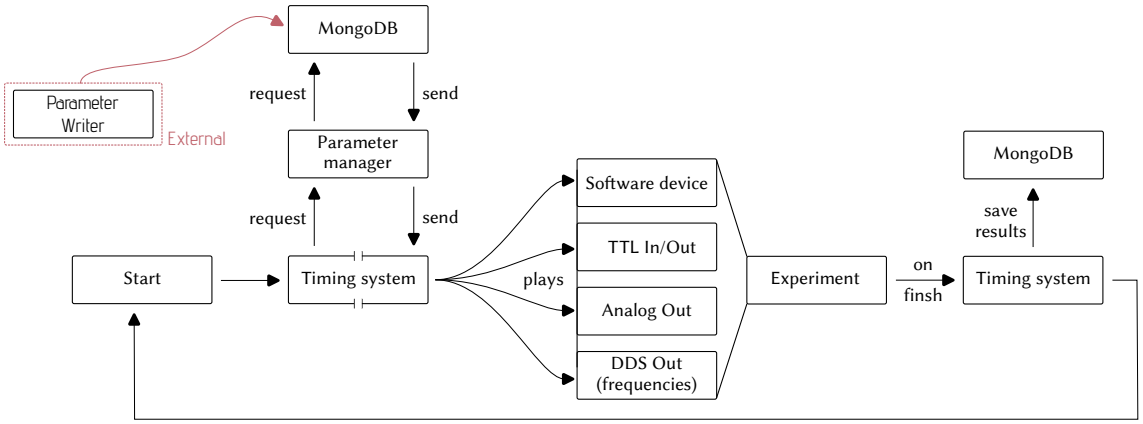


Figure 7.1: **Experimental flow chart.** The experimental control software acts according to this flowchart. The timing system requests parameters from the parameter manager, which are used in the sequence. The timing script runs through, and parameters are inserted at the desired locations, which are used to set values on the different hardware devices. When the experiment has finished, the data is written to a Database and is ready to be evaluated. Since the experiment requests parameters from the parameter manager, which reads from the database, an external process can asynchronously write into the database when a new experiment is configured. This means the parameter writer can also be automatic, when experiments need to be scheduled.

The software was structured to allow high cycle times in the experiment and ensure good usability for the user. They can primarily focus on writing the experiment code without having to do too much management on the side. The software architecture is set up according to Figure 7.2. The user will start the `mqva_server` process, which spawns the three sub-servers.

Scheduler: The Artiq scheduler waits for inputs from the user to start and stop the experiment. As the compilation of a script is often a bottleneck, a watchdog runs in the background and, in the event of a change in the experiment file, compiles the experiment such that it is ready to be run.

Devices control: Whenever the experiment has to be recompiled, it also gets reinitialized, losing all its instances. Oftentimes, we do not want to lose a connection to hardware or reinitialize it due to timing overhead. For this reason, a device control server runs in the background to keep the connection to the hardware alive. It is then possible to connect to the device's control server and manage the hardware from there.

Websocket publisher: This sub-server accepts subscribers, and when the experiment is finished, it sends the results to all its clients. Mostly, this includes the images generated by the camera to get feedback on the atoms. This is extremely useful, for example,

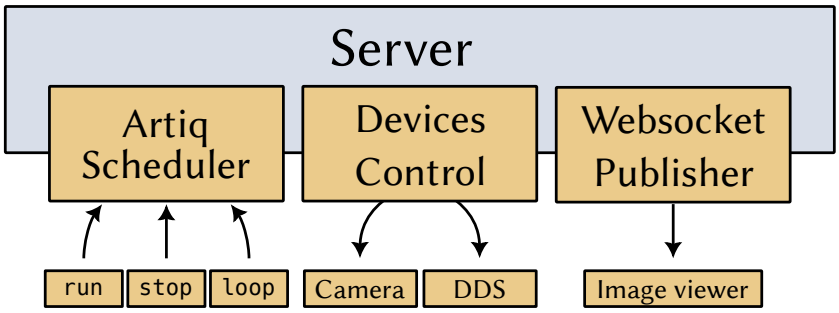


Figure 7.2: **Software architecture of `mqva_control`.** The `mqva_server` is made up of three sub-servers. The artiq scheduler accepts commands to either run the experiment once, in a loop, or stop it. A device control server runs to manage hardware in a separate process, and a WebSocket Publisher sends the results of an experiment to Clients that are subscribed to it.

when optimizing an MOT signal or when doing anything related to tweezers. Having fast feedback allows for debugging before running a lengthy experiment, which in turn means cutting down on development time.

7.4. Parameter manager

Ultracold atom experiments are made up of many parameters, such as frequencies, amplitudes of AOMs, shutter timings, pulse lengths, loading times for MOTs, etc. Most of these parameters will need to be optimized, typically by scanning the variable over a set number of experiments. A good code structure can help keep track of these parameters. In a first version of the software, functions were written to have the parameters as keyword arguments in the function definition, such as in Listing 7.2.

```python
@kernel
def load_mot(blue_mot_frequency = 100*MHz,
             blue_mot_power     = 0.6,
             loading_time       = 200*ms):
    ...
```

Listing 7.2: **Parameter definition in the control script.** Keeping track of parameters by listing them as keyword arguments in the function definitions.

In this approach, functions called without arguments would yield the "optimal" outcome (for which parameters were already optimized). This also meant that, for debugging purposes, it was possible to call the functions with arguments, overriding the default value, or setting an iteration parameter that would be changed on every iteration of the experiment to scan a parameter.

However, this approach is also susceptible to errors, such as when function calls are not cleaned up after a debugging session, which can cause a temporary solution to become permanent. This made it more difficult to find where parameters were set in the first place. Moreover, finding a single parameter in the source code was not straightforward, and sometimes parameters were not defined in the function head at all.

Finally, as the code grew larger, compile times also increased. In our codebase, changing a single parameter means waiting for more than 10 s. Manually scanning a single parameter, for example, to find good starting conditions, would take minutes, with most of the time spent waiting for the compiler. Moreover, it is always good practice to keep the experiment in a running state, so that thermalization effects can be minimized.

These issues gave rise to a new idea, where experiment logic and its parameters were fundamentally separated. For this reason, the experiment now includes a parameter manager, which keeps track of parameters and their values. As it is a separate instance, parameters can be changed asynchronously. The experiment queries the parameter manager for its parameters at the beginning of a run and then runs with the new set.

For experimentalists, this is a much better way of running the experiment because setting parameters and writing logic are usually two distinct tasks. Also, the experiment can now be written sustainably because changing the logic still involves the bottleneck of recompiling the code; however, changing parameters comes with no overhead and is instantaneous. In this approach, the code is written to include flags to switch between different states of the experiment, such as incorporating specific cooling techniques or switching between a MOT and tweezer experiment.

Parameters are stored in a database, which also allows for retrieving all parameters of a specific experimental run. For convenience, the parameter manager includes functionality to manage parameters from `toml` files and automatically writes them into the database. This architecture gives even more freedom to handle the experiment. Parameters can now be programmatically changed, making it easy to run iterative experiments or switch between experiments, because a complete set of parameters can define each new experiment. For example, running a single-qubit gate would have an experiment with a specific pulse length of the clock light, while a two-qubit gate would set this pulse length to π and another parameter to enable the UV light.

Thus, parameters are managed in specific files, and experiments are written by defining the order in which these files are loaded, known as "load orders". Files that are loaded later can overwrite previously loaded parameters. In this way, it is possible to define a default experiment that contains all optimized parameters, with all switching flags set to false. Experiments will then have to enable parts of the experiment by setting specific flags to true and, if necessary, setting new values of parameters.

The load orders also define iteration parameters and should be seen as one specific experiment (e.g., clock spectroscopy, clock Rabi, etc.). An example load order looks like this:

```toml
# Start date of writing this config:
#  2024-03-11
# Author:
#  Robin, Kevin

[load_order]
parameter_groups = [
  "Experiment/2024_03_14_optimized",
  "CoilStage/2024_02_06_optimized",
  "BlueMotStage/2024_02_06_optimized",
  "ImagingStage/2024_02_06_optimized",
  "RedMotStage/2024_02_06_optimized",
  "SisyphusCoolingStage/2024_03_20_optimized",
  "SidebandCoolingStage/2024_03_21_optimized",
  "SortingStage/2024_03_28_optimized",
  "TweezerStage/2024_02_06_optimized",
  "ClockStage/2024_02_06_optimized",

  "UVStage/2024_02_06_optimized",
  "TweezerMain/2024_02_06_optimized",
  "ClockMain/2024_03_14_optimized",
]

run_name = "clock ramsey"
run_description = """
Measuring clock Ramsey fringes
"""

iteration_order = [
```

```
    { parameter = "ClockStage.do_clock_spectroscopy.ramsey_time",
    start = "0.0ms", stop = "60.0ms", num = 21},
]

run_mode = "iterate"
```

Using this approach also allows for easy implementation of a scheduler, which switches between experiments.

7.5. MQVA Scheduler

We established that the experiment runs continuously, and at the beginning of each run, it checks the database for the current set of parameters, which define the exact procedure. This method makes it easy to update the parameters externally and automatically, for example, in scheduled experiments. It can mean experiments that run regularly, at specific times and intervals, or chained experiments. The current implementation of the MQVA Scheduler is subscribed to the websocket server of the main experiment loop, which contains the current shot number. In this way, the scheduler can be configured to run for a certain number of shots before continuing to the next experiment, or returning to the default (or previous) state.

A scheduled experiment is defined in a toml file in the following format:

```toml
[[Measurements]]
    load_order = "clock/clock_spectroscopy"
    run_mode = "iterate"
    n_shots = 20

    iteration_order = [
        { parameter = "ClockStage.prepare.pulse_duration", start =
        "1us", stop = "2us", num = 31, delta = false }
    ]

    [Measurements.parameter_overrides]
        "ClockStage.prepare.delay_between_pulses" = "10ms"
        "ClockStage.prepare_clock_spectrum_card.spin_echo_ampl" =
        "0.5"
```

The measurement takes the `load_order`, as defined above, the `run_mode`, which can be either loop or iterate, and the number of shots we want to take. In iteration mode, the iteration variables need to be passed in the same way as in the `load_order`, where multiple iteration variables can be passed. Finally, parameters can be overridden.

Currently, measurements cannot be scheduled for specific times or intervals. When the main experiment loop runs, the measurement can be executed by calling the command `mqva_scheduler` with the path to the above toml file.

7.6. MQVAExperiment base class

In addition to allowing for high cycle times, another focus in writing the control software was on the system's ease of use. A user of the control system will want to focus solely on writing its logic, without having to worry about mundane tasks like handling scopes, moving instances, or memory management.

A custom class `MQVAExperiment`, inheriting from `artiq.experiment.EnvExperiment`, was written to handle these tasks. It also considers that the highest priority is system performance, allowing for high cycle times. When the class is instantiated, its `build` function is called. This function should handle most of the overhead and is only run once every time the experiment is recompiled. Most importantly, it creates instances of so-called user channels, which are interfaces to different parts of the experiment.

User channels are Python classes and can be anything, ranging from simple DI/O or analog channels to more complex interfaces, such as those for cameras. All user channels have a `toml` configuration, which is parsed as a dictionary and passed as `**kwargs` to the constructor of the user channel. An instance of the user channel is passed directly to every function in the experiment, so that it is accessible at every step. A simple example of a user channel is the `FastinoWrapper` in Listing 7.3. It has a kernel function that allows the user to set the voltage of an analog channel at runtime. Other user channels may also incorporate `rpc` functions, e.g., cameras that need to have an acquisition loop running in the background. The experiment directory contains a directory structure as listed in Listing 7.4.

The `build` function here reads the configuration files in the `configs` folder, which contains either experimental parameters or user channels. The user channels are then instantiated based on their definitions (see Figure 7.2) and stored in the `MQVAExperiment.user_channels` variable.

```
from artiq.experiment import ns
from artiq.experiment import kernel

class FastinoWrapper:
    def __init__(self, get_device, channel_name, channel_type,
    channel):
        channel -= 1

        # One fastino has 32 channels
        fastino_num = channel // 32
        fastino = get_device(f"fastino{fastino_num}")

        self._fastino = fastino
        self._channel_num = channel - 32*fastino_num

    @kernel
    def set_voltage(self, v):
        self._fastino.set_dac(self._channel_num, v)
        delay(10*ns)
```

Listing 7.3: **User channel example.** An example of the `FastinoWrapper` user channel. It takes the `get_device` function of the current experiment instance, as well as parameters from the configuration file. We have two Fastinos in the experiment, each with 32 channels, such that the second Fastino starts at channel number 33.

```
EXPERIMENT_NAME
├── configs
│   ├── 0-CONFIG_NAME.toml
│   └── 1-CONFIG_NAME.toml
├── main.py
└── parameters
    ├── load_orders
    │   └── ...
    ├── parameter_groups
    │   └── ...
    └── run_config.toml
```

Listing 7.4: **Experiment script folder structure.** The folder structure of an experiment is defined in the mqva_control software stack.

When the user sends a command to run the experiment either via `mqva_submit run` or `mqva_submit loop`, the `MQVAExperiment.run` function is called, which initially runs on the FPGA. This function starts the main loop of the experiment and is expected to run until the user stops it. It contains calls to the following functions in this order:

1. **`ParameterManager.reload_parameters()`:** Check the code for parameter definitions and replace function calls to include the loaded parameters.
2. **`MQVAExperiment.prepare()`:** The initialization function of the actual experiment. Runs on the host and is implemented by the user.
3. **`MQVAExperiment.run()`:** The main logic of the experiment. Runs on the kernel and is implemented by the user.
4. **`MQVAExperiment.save_all()`:** Saves the generated data (like images) in a database and as HDF5 files and publishes the data using the Websocket Publisher.

The run time of `ParameterManager.reload_parameters()` is very short (<1ms) and `MQVAExperiment.save_all()` runs asynchronously, therefore taking no time at all. Thus, unless the `MQVAExperiment.prepare()` function does not contain any significant overhead, the experimental time is given by the timings defined in the main function.

8. Conclusion

In this thesis, we present a new Strontium tweezers platform for neutral atom quantum computing. The apparatus was built from scratch, based on previous experiments and experiences in the Bloch group. We managed to create a more compact vacuum system, which rests on rails, opening up debugging options of the objective systems. Although at the time of publication of this thesis, it has been almost four years since the kickoff of this experiment, the first single-qubit and Rydberg results were already achieved in January 2024, proving that this platform allows building up experiments on a short time scale. In fact, with this blueprint in place, we expect this process to go even faster now.

A large portion of the experimental setup was building cooling laser systems in a standard and compact way. We have shown that it is possible to fit the cooling beams into a standard server rack, with fiber outputs toward the experiment to preserve space around the chamber. Furthermore, we use ILAs in the rack, which has become incredibly useful in generating the necessary power. Thus, it was possible to quickly find a MOT signal and to load atoms into optical tweezers generated by AODs. The vacuum system has proven to be exceptional, with atomic lifetimes up to 27 min on selected tweezers.

We show how to generate optical tweezer arrays using crossed AODs driven by a Spectrum M4i6631-x8 PCIe card. It has many options for replaying custom waveforms, either from memory or generated on-the-fly by the CPU/GPU. We demonstrate how to create a perfectly filled array in a generalized 2D-AOD array by moving and removing lines and columns, and we have used this method for parts of the experiment.

The experimental sequence was classified, beginning with the blue and red MOT atom numbers and temperatures. The sequence continues by loading the atoms into tweezers, where parity projection takes care of only loading single atoms into tweezers. A sisyphus cooling stage pre-cools the atoms, such that sideband cooling allows cooling into the motional ground state $|(n) = 0.21$, resulting in a ground state occupation of $p_0 = 83\%$. The trap frequencies as part of the sideband cooling optimization series are measured and characterized.

With the atoms fully initialized (loaded and cooled), we show that the system is functional and perform various experiments on the transport of atoms. Here we compare two different transport modes, the constant jerk and the adiabatic sine, and find that

both trajectories have regimes where they excel. Quantum computing applications may want to use the transport of atoms as a method to distinguish the neutral atom quantum computing platform from others. For this, it is essential to understand the effect on the coherence of an atomic state during its transport. Naturally, we test the loss and coherence in a Ramsey measurement, where the atom is in a superposition state of the ground and excited states. This does not show any large-scale reduction in coherence; however, an IRB measurement does reveal that the effect is not negligible. For a move on the order of one tweezer spacing, the lowest recorded error is $\varepsilon_{\text{movement}} = 0.17\%$. This is on par with the current record for single-qubit gate errors. Although optimal regimes for configurations of distance and acceleration or duration were theoretically explored, they have yet to be reviewed in the IRB measurement. Moreover, this measurement was only performed with the constant jerk trajectory. Thus, by performing more tests, the fidelity can improve further.

8.1. Outlook

Ultimately, the goal of this experiment is to run quantum algorithms. As was shown, single-qubit operations on the clock transition are already possible. There have been some preliminary results on Rabi oscillations on the Rydberg transition, which is the prerequisite for interaction gates and the obvious next step for the experiment. Once single and multi-qubit gates work with reasonable fidelities, simple quantum algorithms [134,135] can be tested. Rydberg-based multi-qubit interactions can be enhanced when atoms are closer together, and such a lattice will be implemented in the future.

Many new ideas will enable the system to become more competitive. Single-qubit addressing will become very important, as currently, all gates only work globally. The cycle time is currently around 0.5 Hz. With lossless detection and reinitialization of qubits, this could decrease to 10 Hz [136]. To increase the detection fidelity, erasure as shown in [41] can be implemented, requiring a fast imaging setup.

However, the future of quantum computing relies on high-fidelity gates. Long algorithms can only be performed if accumulated errors are low; therefore, error correction schemes must be implemented. Schemes already exist for using composite pulses to mitigate specific error sources [137], however, more generally, surface codes have been demonstrated before [36,138], and new proposals are available [139–141]. On a similar note, optimal control trajectories can be implemented relatively easily, as well as other fault-tolerant schemes, such as brick-wall circuits [142]. The long lifetimes presented here allow for running long quantum algorithms. In the future, these will be implemented via remote access, with the software stack in this thesis serving as the foundation. An exciting prospect is seeing how the motion of the qubit is integrated into the algorithms, either via error correction or as a phase gate.

Appendix

A. Magnetic field of rectangular and circular coils

```python
def B_wire(x, y, z, L, I):
    """
    Calculates the magnetic field of a straight, finite wire of length L
    """
    L /= 2
    r = np.sqrt(x**2 + y**2)

    f = mu_0 * I / (4*np.pi * r)
    f1 = (z+L)/np.sqrt(r**2 + (z+L)**2)
    f2 = (z-L)/np.sqrt(r**2 + (z-L)**2)
    return f * (f1-f2) *1e4 # convert to gauss

def B_rectangle(z, width, height, I):
    """
    Calculates the magnetic field of a rectangular coil with width w and height h
    by assembling four straight wires
    """
    wire1 = B_wire(width/2, z, 0, height, I)
    wire2 = B_wire(height/2, z, 0, width, I)

    return 2*wire1 + 2*wire2

def B_circle(z, R, I):
    """
    Calculates the magnetic field of a circular coil with radius R
    """
    return (mu_0 * I * R**2) / (2 * (R**2 + z**2)**(3/2)) *1e4 # convert to gauss

def stack_rectangular_field(z, n_axially, n_radially, wire_thickness, width, height, I):
    """
    Coils are generally wound with a specified number of turns. This function
    assembles a rectangular coil with a number of turns in the axial and radial
    direction of the coil given a wire thickness.
    """
    field = np.zeros_like(xfine)
    for n_ax in range(n_axially):
```

```
37          for n_r in range(n_radially):
38              field += B_rectangle(z + (2*n_ax+1)*wire_thickness/2, width +
                (2*n_r+1)*wire_thickness/2, height + (2*n_r+1)*wire_thickness/2, I)
39      return field
40
41  def stack_circular_field(z, n_axially, n_radially, wire_thickness, radius, I):
42      """
43      Coils are generally wound with a specified number of turns. This function
44      assembles a circular coil with a number of turns in the axial and radial
45      direction of the coil given a wire thickness.
46      """
47      field = np.zeros_like(xfine)
48      for n_ax in range(n_axially):
49          for n_r in range(n_radially):
50              field += B_circle(z + (2*n_ax+1)*wire_thickness/2, radius +
                (2*n_r+1)*wire_thickness/2, I)
51      return field
```

B. Automatic relocking of ILAs via Arduino

```c
/***********
According to https://epjtechniquesandinstrumentation.springeropen.com/articles/10.1140/epjti/s40485-024-00113-z, the process is as follows:
 - Scan the current from top to bottom ~15 times (to thermalize the jump point)
 - Scan around the jump point:
   - Sweep up
   - Sweep down
   - Take trace during sweep down and identify jump point (moving stddev)
   - Move up
   - Move down to jump point with some offset
***********/

/***********
HOW TO USE

1. The SMA cables from left to right are:
  - Actuator channel 0
  - Photodiode channel 0
  - Actuator channel 1
  - Photodiode channel 1
  - Actuator channel 2
  - Photodiode channel 2
  - Actuator channel 3
  - Photodiode channel 3

2. Plug in the actuator and the photodiode of the desired channel
3. Connect the arduino to the computer and load this program
4. Connect to the serial monitor (top-right in the arduino IDE)

You have a number of commands available, these will be listed later, for now do the following
5. type 'set currentChannel x' where x refers to the channel you are using
6. type 'voltage 0.1'. This sets the voltage output of this channel to 0.1, which is half way between 0.0 and 0.2 (the scanning range)
7. Lock the ILA manually and note down the current. This way we can make sure that the lock point is centered in the sweep range
8. Type 'lock' a few times, make sure that in the output window, Lockpoint x remains around the same number every time.
   If it locks to the correct lock point, you are done. Otherwise:

We will need to calibrate the delay between setting the current and seeing the signal on the photodiode.
9. Look at the current. If it is higher than the value you noted down earlier, we will decrease the delay, otherwise it will be increased.
10. Type 'set lockOffsetStep x' where x is proportional to the delay (the value is in voltage steps), the default is 50, you can go quite high, since we do 1000 steps, but I'd
    say it is quite unlikely we have more than 200 delay.
11. Repeat steps 8 to 10 until the ILA remains locked.
***********/

// === CONFIGURATION ===
```

```
44  float vStart = 0.2;                          // sweep start voltage
45  float vEnd = 0.0;                            // sweep end voltage
46  int numSweeps = 30;                          // number of sweeps to average over
47  int numSteps = 1000;                         // number of steps for each sweep
48  int lockOffsetSteps[4] = { 25, -100, 100, 50 };  // there is some delay from when the value is set to when it is read. This offset you must change, anything > 0
49  int windowSize = 10;                         // To find the lock point, we take a moving std over this window
50  float vDeltaPct = 0.5;                       // When checking if the laser is unlocked, it looks for the difference between the voltage in the locked and unlocked state times this percent value
51  int currentChannel = 0;
52
53  int settleTime_us = 10;
54
55  // Don't change anything below this line
56  uint8_t DAC_MAP[4] = { 0, 2, 1, 3 };
57  uint8_t ADC_PINS[4] = { A2, A3, A4, A5 };
58
59  const float vRef = 1.25;  // Voltage reference
60  const int adcMax = 4095;  // 12-bit ADC
61  int samples[1000];        // Store ADC readings
62
63  bool startRelocking = false; // this will be set to true once the on-board button is pressed
64                               // you can also manually set it to true to always start the locking procedure on boot
65
66  float unlockedPDVoltagesAtLockPoint[4] = { 0.0, 0.0, 0.0, 0.0 };
67  float finalPDVoltagesAtLockPoint[4] = { 0.0, 0.0, 0.0, 0.0 };
68  // ========================
69
70  float dacToVoltage(int dacValue) {
71    // Ensure the DAC value is within the valid range
72    if (dacValue < 0) dacValue = 0;
73    if (dacValue > adcMax) dacValue = adcMax;
74
75    // Convert DAC value to voltage
76    float voltage = (float(dacValue) / adcMax) * vRef;
77    return voltage;
78  }
79
80  int voltageToDac(float voltage) {
81    // Clamp voltage to the valid range
82    if (voltage < 0) voltage = 0;
83    if (voltage > vRef) voltage = vRef;
84
85    // Convert voltage to DAC value
86    int dacValue = int((voltage / vRef) * adcMax);
87    return dacValue;
88  }
89
90  void splitString(String input, char delimiter, String parts[], int maxParts) {
91    int partIndex = 0;
92    int startIndex = 0;
93    int delimIndex = 0;
```

```
94
95    while (partIndex < maxParts && (delimIndex = input.indexOf(delimiter, startIndex)) != -1) {
96      parts[partIndex++] = input.substring(startIndex, delimIndex);
97      startIndex = delimIndex + 1;
98    }
99
100   // Add the final part (or the whole string if no delimiter was found)
101   if (partIndex < maxParts) {
102     parts[partIndex++] = input.substring(startIndex);
103   }
104
105   // Optional: clear remaining elements
106   while (partIndex < maxParts) {
107     parts[partIndex++] = "";
108   }
109 }
110
111 inline float getVoltageFromSweep(float vStart, float vEnd, int lockPointStep, int numSteps) {
112   return vStart + lockPointStep * (vEnd - vStart) / (numSteps - 1);
113 }
114
115 void setup() {
116   Serial.begin(115200);
117   pinMode(BTN_BUILTIN, INPUT_PULLUP);
118   analogWriteResolution(12);
119   analogReadResolution(12);
120   analogReferenceDAC(DAC_VREF_1V25);
121   Serial.println("Ready. Send 'sweep vStart vEnd numSteps' to run sweep.");
122 }
123
124 void loop() {
125   if (digitalRead(BTN_BUILTIN) == LOW) {
126     startRelocking = true;
127     Serial.println("Button was pressed!");
128   }
129
130   if (startRelocking) {
131     float currentVoltage = dacToVoltage(analogRead(ADC_PINS[currentChannel]));
132     Serial.print("Current Voltage: ");
133     Serial.println(currentVoltage, 4);
134     Serial.print("Current Channel: ");
135     Serial.println(currentChannel);
136     delay(5000);
137
138     if (abs(currentVoltage - finalPDVoltagesAtLockPoint[currentChannel]) > abs(finalPDVoltagesAtLockPoint[currentChannel] - unlockedPDVoltagesAtLockPoint[currentChannel]) * vDeltaPct) {
139       getLockVoltages();
140       Serial.print("Relocking, ");
141       Serial.print(currentVoltage);
142       Serial.print(", ");
143       Serial.print(finalPDVoltagesAtLockPoint[currentChannel]);
144       Serial.print(", ");
145       Serial.print(unlockedPDVoltagesAtLockPoint[currentChannel]);
146       Serial.print(", ");
```

```
147       Serial.println(vDeltaPct);
148     }
149   }
150
151   if (Serial.available()) {
152     handleInput();
153   }
154 }
155
156 void setVariable(String name, String value) {
157   if (name == "vStart") {
158     vStart = value.toFloat();
159   } else if (name == "vEnd") {
160     vEnd = value.toFloat();
161   } else if (name == "numSweeps") {
162     numSweeps = value.toInt();
163   } else if (name == "numSteps") {
164     numSteps = value.toInt();
165   } else if (name == "lockOffsetStep") {
166     lockOffsetSteps[currentChannel] = value.toInt();
167   } else if (name == "windowSize") {
168     windowSize = value.toInt();
169   } else if (name == "currentChannel") {
170     currentChannel = value.toInt();
171   } else if (name == "settleTime_us") {
172     settleTime_us = value.toInt();
173   } else if (name == "startRelocking") {
174     startRelocking = bool(value.toInt());
175   } else {
176     Serial.print("Unknown variable: ");
177     Serial.println(name);
178     return;
179   }
180
181   Serial.print("Set ");
182   Serial.print(name);
183   Serial.print(" to ");
184   Serial.println(value);
185 }
186
187 void handleInput() {
188   String input = Serial.readStringUntil('\n');  // Read full command line
189   input.trim();
190
191   // Variable to hold parsed data
192   String tokens[7];  // Max 7 tokens for 'lock' command
193
194   // Parse the input command
195   if (input.startsWith("set ")) {
196     splitString(input, ' ', tokens, 3);
197
198     String varName = tokens[1];
199     String varValue = tokens[2];
200
```

```
201     setVariable(varName, varValue);
202   } else if (input == "sweep") {
203     // Call handleSweep with parsed parameters
204     doSweep();
205
206   } else if (input == "lock") {
207     // Call handleLock with parsed parameters
208     getLockVoltages();
209
210   } else if (input.startsWith("voltage ")) {
211     splitString(input, ' ', tokens, 2);
212     float voltage = tokens[1].toFloat();
213
214     // Call handleSetVoltage with parsed voltage
215     doSetVoltage(voltage);
216
217   } else {
218     Serial.println("Unknown command. Use 'sweep vStart vEnd numSteps' to sweep or 's
      <voltage>' to set.");
219   }
220 }
221
222 void doSweep() {
223   // Run sweep logic
224   for (int i = 0; i < numSweeps; ++i)
225     runSweep(vStart, vEnd, numSteps);
226
227   int lockPointStep = getLockPoint(numSweeps, numSteps, false);
228
229   // Calculate the voltage at the lock point step
230   float voltageAtLockPoint = getVoltageFromSweep(vStart, vEnd, lockPointStep, numSteps);
231   Serial.print("Scanning down to ");
232   Serial.println(voltageAtLockPoint, 4);
233
234   runSweep(vStart, voltageAtLockPoint, lockPointStep);
235
236   Serial.println("Done. Send 'sweep' command to run again.");
237 }
238
239 void getLockVoltages() {
240   // Run lock logic
241   float vDelta = vStart - vEnd;
242   int lockPointStep;
243   float voltageAtLockPoint;
244
245   // Run the sweeps and get lock point voltage
246   for (int i = 0; i < numSweeps; ++i) {
247     runSweep(vStart, vEnd, numSteps);
248     lockPointStep = getLockPoint(15, numSteps, false);
249     voltageAtLockPoint = getVoltageFromSweep(vStart, vEnd, lockPointStep, numSteps);
250   }
251
252   Serial.print("Intermediate lock point voltage: ");
253   Serial.println(voltageAtLockPoint, 4);
254
```

```
  // Sweep up
  runSweep(vEnd, vStart, numSteps);
  // Sweep down
  runSweep(vStart, vEnd, numSteps);

  int finalLockPointStep = getLockPoint(1, numSteps, false);
  float finalSetVoltageAtLockPoint = getVoltageFromSweep(vStart, vEnd, finalLockPointStep,
numSteps);
  finalPDVoltagesAtLockPoint[currentChannel] = dacToVoltage(samples[finalLockPointStep]);

  int unlockedStep = finalSetVoltageAtLockPoint + 2 * lockOffsetSteps[currentChannel];
  unlockedPDVoltagesAtLockPoint[currentChannel] = dacToVoltage(samples[unlockedStep]);

  Serial.print("Final set lock point voltage: ");
  Serial.println(finalSetVoltageAtLockPoint, 4);
  Serial.print("Final PD lock point voltage: ");
  Serial.println(finalPDVoltagesAtLockPoint[currentChannel], 4);
  Serial.print("Unlocked PD voltage: ");
  Serial.println(unlockedPDVoltagesAtLockPoint[currentChannel], 4);

  Serial.print("Sweeping from ");
  Serial.print(vStart, 4);
  Serial.print(" to ");
  Serial.println(finalSetVoltageAtLockPoint, 4);

  // Sweep up
  runSweep(vEnd, vStart, numSteps);
  // Sweep down
  runSweep(vStart, finalSetVoltageAtLockPoint, numSteps);
}

void doSetVoltage(float voltage) {
  // For setting specific voltage
  int dacVal = voltageToDac(voltage);

  analogWrite((_dac_channel_t)(DAC_MAP[currentChannel]), dacVal);

  Serial.print("Set Analog channel ");
  Serial.print(currentChannel);
  Serial.print(" to ");
  Serial.print(voltage, 3);
  Serial.print(" V (DAC = ");
  Serial.print(dacVal);
  Serial.println(")");
}

void runSweep(float vStart, float vEnd, int numSteps) {
  for (int step = 0; step < numSteps; step++) {
    float voltage = vStart + step * (vEnd - vStart) / (numSteps - 1);
    int dacVal = voltageToDac(voltage);

    analogWrite((_dac_channel_t)(DAC_MAP[currentChannel]), dacVal);
    delayMicroseconds(settleTime_us);

    int adcVal = analogRead(ADC_PINS[currentChannel]);
```

```
309     samples[step] = adcVal;
310   }
311
312   Serial.println("Finished sweep");
313 }
314
315 int getLockPoint(int numSweeps, int numSteps, bool do_print) {
316   float window[windowSize];
317   int windowCount = 0;
318   int windowIndex = 0;
319
320   float maxStdDev = -INFINITY;
321   int maxStdDevStep = -1;
322
323   if (do_print) {
324     Serial.println("START_PLOT");
325     Serial.println("Step,ADC_avg,Voltage,StdDev");
326   }
327
328   for (int step = 0; step < numSteps; step++) {
329     float lastMeasuredADC = samples[step];
330     float measuredVoltage = lastMeasuredADC * (vRef / adcMax);
331
332     // Update circular buffer
333     window[windowIndex] = lastMeasuredADC;
334     windowIndex = (windowIndex + 1) % windowSize;
335     if (windowCount < windowSize) windowCount++;
336
337     // Compute stddev if enough points
338     float stddev = 0.0;
339     if (windowCount == windowSize) {
340       float mean = 0.0;
341       for (int i = 0; i < windowSize; i++) {
342         mean += window[i];
343       }
344       mean /= windowSize;
345
346       float variance = 0.0;
347       for (int i = 0; i < windowSize; i++) {
348         float diff = window[i] - mean;
349         variance += diff * diff;
350       }
351       variance /= (windowSize - 1);
352       stddev = sqrt(variance);
353
354       // Update maximum tracking
355       if (stddev > maxStdDev) {
356         maxStdDev = stddev;
357         maxStdDevStep = step;
358       }
359     }
360
361     if (do_print) {
362       // Print immediately
```

```
363         Serial.print(step);
364         Serial.print(", ");
365         Serial.print(lastMeasuredADC);
366         Serial.print(", ");
367         Serial.print(measuredVoltage, 4);
368         Serial.print(", ");
369         Serial.println(stddev, 4);
370       }
371     }
372
373     if (do_print) {
374       Serial.println("END_PLOT");
375     }
376
377     int lockpoint = maxStdDevStep - windowSize / 2 - lockOffsetSteps[currentChannel];
378
379     // Output final result
380     Serial.println("START_LOCKPOINT");
381     Serial.print("Maximum StdDev = ");
382     Serial.print(maxStdDev, 4);
383     Serial.print(" at step ");
384     Serial.println(maxStdDevStep);
385     Serial.print("Lockpoint ");
386     Serial.println(lockpoint);
387     Serial.print("Offset: ");
388     Serial.println(lockOffsetSteps[currentChannel]);
389     Serial.print("Voltage: ");
390     Serial.println(getVoltageFromSweep(vStart, vEnd, lockpoint, numSteps), 4);
391     Serial.println("END_LOCKPOINT");
392
393     return lockpoint;
394 }
```

C. Calculate tweezer sorting probability

```python
import numpy as np
import scipy

def n_times(f, x, n):
    result = x
    for _ in range(n):
        result = f(result, x)
    return result

def disjunct_probability(p1, p2):
    return p1 + p2 - p1*p2

def cumulative_binomial(index_start, index_end, p):
    cums = []
    for maxn in range(index_start, index_end+1):
        r = np.arange(index_start, maxn+1)
        binom_pmf = scipy.stats.binom.pmf(r, maxn, p)
        s = np.cumsum(binom_pmf)[-1]

        cums.append(s)

    return np.array(cums)

def compute_tweezer_probability(target_num_rows, max_rows_available, min_cols_available, max_cols_available):
    x = np.arange(target_num_rows, max_rows_available+1)
    results = []

    for num_cols_available in range(2, max_cols_available + 2):
        y = cumulative_binomial(target_num_rows, max_rows_available, 1 / 4)

        num_combs = int(scipy.special.binom(num_cols_available, 2))
        y = n_times(disjunct_probability, y, num_combs - 1)

        f = 1 - num_cols_available / (max_cols_available - 2 + 4)
        results.append((x, y, f, num_cols_available))

    return results

# EXAMPLE USAGE
results = compute_tweezer_probability(target_num_rows=10, max_rows_available=80, min_cols_available=2, max_cols_available=10)
```

D. Python code for constant jerk ramps

The following is the code used in the lab to generate ramps for sorting atoms:

```python
def interp1d_cubic(start, end, npts=1000):
    """
    Creates a numpy array of points that are interpolated cubically from start to end
    """
    x = np.linspace(-1.0, 1.0, npts)
    fx = -0.5*(x**3-3*x)

    avg = (start+end)/2
    diff = end-start
    return avg + diff/2*fx

def _round_frequency(sampling_rate, num_samples, f):
    """
    Rounds the frequency f to the nearest tooth

    """
    t_window = num_samples / sampling_rate
    f_window = 1/t_window

    num_tooth = np.round(f * t_window)

    if num_tooth % 2 == 1:
        num_tooth += 1
    f_new = num_tooth/t_window

    return f_new, num_tooth

def get_freq_const_jerk(t, fstart, fend):
    """
    Returns a frequency ramp that starts from fstart and ends at fend over t (a 1D numpy array)
    """
    dt = t[1]-t[0]
    T = t[-1]-t[0]+dt
    t = t-t[0]-T/2

    j0 = -12 * (fend-fstart) / T**3
    a0 = 0
    v0 = -1/8 * j0 * T**2
    f0 = (fstart+fend)/2

    C3 = 1/6 * j0
    C2 = 1/2 * a0
    C1 = v0
    C0 = f0

    return C3 * t**3 + C2 * t**2 + C1 * t + C0
```

E. Python code for generating spectrograms

```python
from PIL import Image
from matplotlib import colormaps as cm
import numpy as np

def spectrogram_to_image(times, frequencies, spectrogram_data, output_filename=None,
width=None, height=None, f_lim=None, t_lim=None):
    """
    Converts spectrogram data from scipy.signal.spectrogram into a PIL image
    """

    # Convert the data to dB scale
    spectrogram_data_db = 10 * np.log10(spectrogram_data)

    # Normalize the data between 0 and 255 for 8-bit image representation
    spectrogram_normalized = np.clip((spectrogram_data_db - spectrogram_data_db.min()) /
(spectrogram_data_db.max() - spectrogram_data_db.min()), 0, 1)
    spectrogram_normalized = (spectrogram_normalized * 255).astype(np.uint8)

    # Apply a colormap
    colormap = cm.get_cmap('viridis')
    colored_data = colormap(spectrogram_normalized)  # RGBA data

    # Remove alpha channel
    colored_data_rgb = (colored_data[:, :, :3] * 255).astype(np.uint8)

    if f_lim is None:
        f_lim = (frequencies[0], frequencies[-1])

    if t_lim is None:
        t_lim = (times[0], times[-1])

    # Apply cropping
    f_min_idx = np.searchsorted(frequencies, f_lim[0])
    f_max_idx = np.searchsorted(frequencies, f_lim[1])
    frequencies = frequencies[f_min_idx:f_max_idx]
    colored_data_rgb = colored_data_rgb[f_min_idx:f_max_idx, :]

    t_min_idx = np.searchsorted(times, t_lim[0])
    t_max_idx = np.searchsorted(times, t_lim[1])
    times = times[t_min_idx:t_max_idx]
    colored_data_rgb = colored_data_rgb[:, t_min_idx:t_max_idx]

    # Create an image
    img = Image.fromarray(np.flipud(colored_data_rgb))

    # Resize image
    new_width, new_height = img.size
    new_width = width or new_width
    new_height = height or new_height
```

```
48      img = img.resize((new_width, new_height), Image.Resampling.BILINEAR)
49
50      # Save image
51      if output_filename is not None:
52          img.save(output_filename)
53
54      return img
55
56  def spectrogram_into_axes(ax, times, frequencies, spectrogram_data, output_filename=None,
    width=None, height=None, f_lim=None, t_lim=None):
57      """
58      Inserts spectrogram data from scipy.signal.spectrogram into a matplotlib axis
59      """
60      img = spectrogram_to_image(times, frequencies, spectrogram_data, width=width,
        height=height, f_lim=f_lim, t_lim=t_lim)
61
62      if f_lim is None:
63          f_lim = (frequencies[0], frequencies[-1])
64
65      if t_lim is None:
66          t_lim = (times[0], times[-1])
67
68      ax.imshow(np.array(img), extent=[t_lim[0], t_lim[1], f_lim[0], f_lim[1]], aspect="auto")
69
70  """
71  EXAMPLE BELOW
72  """
73  from scipy.signal import spectrogram
74  import matplotlib.pyplot as plt
75
76  sampling_rate = 420e6
77  duration = 10e-3
78  num_points = int(sampling_rate*duration)
79  t = np.linspace(0, duration, num_points)
80  f = np.linspace(10, 30, num_points)
81  f = 100e6
82
83  plt.figure()
84
85  signal = np.sin(2*np.pi*t*f) ** 5
86  frequencies, times, spectrogram_data = spectrogram(signal, fs=sampling_rate, nperseg=2**12,
    window="hann", noverlap=2**11)
87  spectrogram_into_axes(plt.gca(), times/1e-3, frequencies/1e6, spectrogram_data)
88
89  plt.xlabel("Time (ms)")
90  plt.ylabel("Frequency (MHz)")
91
92  plt.show()
```

Bibliography

[1] N. Bohr, I. On the constitution of atoms and molecules, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **26**, 1 (1913).

[2] M. Jammer, The conceptual development of quantum mechanics, (1966).

[3] R. P. Feynman, Simulating physics with computers, Feynman and computation 133 (2018).

[4] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[5] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Reviews of Modern Physics **86**, 153 (2014).

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge university press, 2010).

[7] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, Quantum computers, nature **464**, 45 (2010).

[8] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, Linear optical quantum computing with photonic qubits, Reviews of modern physics **79**, 135 (2007).

[9] M. H. Devoret and R. J. Schoelkopf, Superconducting circuits for quantum information: an outlook, Science **339**, 1169 (2013).

[10] C. Monroe and J. Kim, Scaling the ion trap quantum processor, Science **339**, 1164 (2013).

[11] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: progress and challenges, Journal of Physics B: Atomic, Molecular and Optical Physics **49**, 202001 (2016).

[12] D. Hayes, D. N. Matsukevich, P. Maunz, D. Hucul, Q. Quraishi, S. Olmschenk, W. Campbell, J. Mizrahi, C. Senko, and C. Monroe, Entanglement of atomic qubits using an optical frequency comb, Physical review letters **104**, 140501 (2010).

[13] D. P. DiVincenzo, Topics in quantum computers, Mesoscopic electron transport 657 (1997).

[14] K. Wintersperger, F. Dommert, T. Ehmer, A. Hoursanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin, and S. Luber, Neutral atom quantum computing hardware: performance and end-user perspective, EPJ Quantum Technology **10**, 32 (2023).

[15] H. Levine et al., Parallel implementation of high-fidelity multiqubit gates with neutral atoms, Physical review letters **123**, 170503 (2019).

[16] T. Graham, M. Kwon, B. Grinkemeyer, Z. Marra, X. Jiang, M. Lichtman, Y. Sun, M. Ebert, and M. Saffman, Rydberg-mediated entanglement in a two-dimensional neutral atom qubit array, Physical review letters **123**, 230501 (2019).

[17] C. Gross and I. Bloch, Quantum simulations with ultracold atoms in optical lattices, Science **357**, 995 (2017).

[18] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch, Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms, nature **415**, 39 (2002).

[19] W. S. Bakr, J. I. Gillen, A. Peng, S. Fölling, and M. Greiner, A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice, Nature **462**, 74 (2009).

[20] J. F. Sherson, C. Weitenberg, M. Endres, M. Cheneau, I. Bloch, and S. Kuhr, Single-atom-resolved fluorescence imaging of an atomic Mott insulator, Nature **467**, 68 (2010).

[21] I. Bloch, Ultracold quantum gases in optical lattices, Nature physics **1**, 23 (2005).

[22] O. Morsch and M. Oberthaler, Dynamics of Bose-Einstein condensates in optical lattices, Reviews of modern physics **78**, 179 (2006).

[23] W. Ketterle and N. Van Druten, Evaporative cooling of trapped atoms, Advances in atomic, molecular, and optical physics **37**, 181 (1996).

[24] A. J. Olson, R. J. Niffenegger, and Y. P. Chen, Optimizing the efficiency of evaporative cooling in optical dipole traps, Physical Review A—Atomic, Molecular, and Optical Physics **87**, 53613 (2013).

[25] T. Lauber, J. Küber, O. Wille, and G. Birkl, Optimized Bose-Einstein-condensate production in a dipole trap based on a 1070-nm multifrequency laser: Influence of enhanced two-body loss on the evaporation process, Physical Review A—Atomic, Molecular, and Optical Physics **84**, 43641 (2011).

[26] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).

[27] D. Barredo, S. De Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, Science **354**, 1021 (2016).

[28] A. Kumar, T.-Y. Wu, F. Giraldo, and D. S. Weiss, Sorting ultracold atoms in a three-dimensional optical lattice in a realization of Maxwell's demon, Nature **561**, 83 (2018).

[29] N. Schlosser, G. Reymond, I. Protsenko, and P. Grangier, Sub-poissonian loading of single atoms in a microscopic dipole trap, Nature **411**, 1024 (2001).

[30] A. M. Kaufman and K.-K. Ni, Quantum science with optical tweezer arrays of ultracold atoms and molecules, Nature Physics **17**, 1324 (2021).

[31] F. Nogrette, H. Labuhn, S. Ravets, D. Barredo, L. Béguin, A. Vernier, T. Lahaye, and A. Browaeys, Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries, Physical Review X **4**, 21034 (2014).

[32] D. Barredo, V. Lienhard, S. De Leseleuc, T. Lahaye, and A. Browaeys, Synthetic three-dimensional atomic structures assembled atom by atom, Nature **561**, 79 (2018).

[33] D. Wei, D. Adler, K. Srakaew, S. Agrawal, P. Weckesser, I. Bloch, and J. Zeiher, Observation of brane parity order in programmable optical lattices, Physical Review X **13**, 21042 (2023).

[34] H. J. Manetsch, G. Nomura, E. Bataille, K. H. Leung, X. Lv, and M. Endres, A tweezer array with 6100 highly coherent atomic qubits, arXiv preprint arXiv:2403.12021 (2024).

[35] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).

[36] S. J. Evered et al., High-fidelity parallel entangling gates on a neutral-atom quantum computer, Nature **622**, 268 (2023).

[37] D. Bluvstein et al., Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).

[38] D. Bluvstein et al., A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).

[39] H. Bernien et al., Probing many-body dynamics on a 51-atom quantum simulator, Nature **551**, 579 (2017).

[40] D. Yavuz, P. Kulatunga, E. Urban, T. A. Johnson, N. Proite, T. Henage, T. Walker, and M. Saffman, Fast ground state manipulation of neutral atoms in microscopic optical traps, Physical Review Letters **96**, 63001 (2006).

[41] Y. Wu, S. Kolkowitz, S. Puri, and J. D. Thompson, Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays, Nature communications **13**, 4657 (2022).

[42] E. Knill and R. Laflamme, Theory of quantum error-correcting codes, Physical Review A **55**, 900 (1997).

[43] B. Grinkemeyer, E. Guardado-Sanchez, I. Dimitrova, D. Shchepanovich, G. E. Mandopoulou, J. Borregaard, V. Vuletić, and M. D. Lukin, Error-detected quantum operations with neutral atoms mediated by an optical cavity, Science **387**, 1301 (2025).

[44] D. Schrader, S. Kuhr, W. Alt, M. Müller, V. Gomer, and D. Meschede, An optical conveyor belt for single neutral atoms, Applied Physics B **73**, 819 (2001).

[45] A. Radnaev et al., A universal neutral-atom quantum computer with individual optical addressing and non-destructive readout, arXiv preprint arXiv:2408.08288 (2024).

[46] M. Martinez-Dorantes, W. Alt, J. Gallego, S. Ghosh, L. Ratschbacher, Y. Völzke, and D. Meschede, Fast nondestructive parallel readout of neutral atom registers in optical potentials, Physical review letters **119**, 180503 (2017).

[47] S. Pucher, V. Klüsener, F. Spriestersbach, J. Geiger, A. Schindewolf, I. Bloch, and S. Blatt, Fine-structure qubit encoded in metastable strontium trapped in an optical lattice, Physical Review Letters **132**, 150605 (2024).

[48] C. Hölzl, A. Götzelmann, M. Wirth, M. S. Safronova, S. Weber, and F. Meinert, Motional ground-state cooling of single atoms in state-dependent optical tweezers, Physical Review Research **5**, 33093 (2023).

[49] A. Pagano, D. Jaschke, W. Weiss, and S. Montangero, Optimal control transport of neutral atoms in optical tweezers at finite temperature, Physical Review Research **6**, 33282 (2024).

[50] J. Dalibard and C. Salomon, Trapping Neutral Atoms with Electromagnetic Fields, Europhysics News **19**, 68 (1988).

[51] C. Adams and E. Riis, Laser cooling and trapping of neutral atoms, Progress in quantum electronics **21**, 1 (1997).

[52] K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. Kurn, and W. Ketterle, Bose-Einstein condensation in a gas of sodium atoms, Physical review letters **75**, 3969 (1995).

[53] C. E. Wieman, D. E. Pritchard, and D. J. Wineland, Atom cooling, trapping, and quantum manipulation, Reviews of Modern Physics **71**, S253 (1999).

[54] R. Grimm, M. Weidemüller, and Y. B. Ovchinnikov, Optical dipole traps for neutral atoms, Advances in atomic, molecular, and optical physics **42**, 95 (2000).

[55] D. Heinzen and D. Wineland, Quantum-limited cooling and detection of radio-frequency oscillations by laser-cooled ions, Physical Review A **42**, 2977 (1990).

[56] A. Steane, The ion trap quantum information processor, Applied Physics B **64**, 623 (1997).

[57] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, Quantum dynamics of single trapped ions, Reviews of Modern Physics **75**, 281 (2003).

[58] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, Applied Physics Reviews **6**, (2019).

[59] H. Häffner, C. F. Roos, and R. Blatt, Quantum computing with trapped ions, Physics reports **469**, 155 (2008).

[60] W. D. Phillips, Nobel Lecture: Laser cooling and trapping of neutral atoms, Reviews of Modern Physics **70**, 721 (1998).

[61] C. Baroni, G. Lamporesi, and M. Zaccanti, Quantum mixtures of ultracold gases of neutral atoms, Nature Reviews Physics 1 (2024).

[62] R. Tao, M. Ammenwerth, F. Gyger, I. Bloch, and J. Zeiher, High-fidelity detection of large-scale atom arrays in an optical lattice, Physical Review Letters **133**, 13401 (2024).

[63] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller, Cold bosonic atoms in optical lattices, Physical Review Letters **81**, 3108 (1998).

[64] A. Browaeys and T. Lahaye, Many-body physics with individually controlled Rydberg atoms, Nature Physics **16**, 132 (2020).

[65] A. M. Kaufman, B. J. Lester, and C. A. Regal, Cooling a single atom in an optical tweezer to its quantum ground state, Physical Review X **2**, 41014 (2012).

[66] J. Nomura, T. Momma, Y. Kojima, Y. Hisai, T. Kobayashi, D. Akamatsu, and F.-L. Hong, Direct loading of Yb atoms into a 3D magneto-optical trap from a dispenser atomic source, AIP Advances **13**, (2023).

[67] J. Samland, Pumping the Zeeman Slower, 2019.

[68] H. Werij, C. H. Greene, C. Theodosiou, and A. Gallagher, Oscillator strengths and radiative branching ratios in atomic Sr, Physical Review A **46**, 1248 (1992).

[69] J. Sansonetti and G. Nave, Wavelengths, transition probabilities, and energy levels for the spectrum of neutral strontium (SrI), Journal of Physical and Chemical Reference Data **39**, (2010).

[70] S. Pucher, S. L. Kristensen, and R. M. Kroeze, ^{}{88} Sr Reference Data, arXiv preprint arXiv:2507.10487 (2025).

[71] D. Tsevas, *Phd Thesis in Preparation*, (unpublished).

[72] *Atomic Weights and Isotopic Compositions for Strontium*, https://physics.nist.gov/cgi-bin/Compositions/stand_alone.pl?ele=Sr.

[73] N. Lorenz, A Rydberg Tweezer Platform with Potassium Atoms, Doctoral dissertation, 2021.

[74] L. Festa, Black-Body Radiation Induced Correlated Excitation of Potassium Rydberg Atoms in Tweezer Arrays, Doctoral dissertation, 2021.

[75] J. Menssen, A White Light Interferometer for the Precise Characterization of Optical Surfaces, 2023.

[76] C. Makrides, D. S. Barker, J. A. Fedchak, J. Scherschligt, S. Eckel, and E. Tiesinga, Elastic rate coefficients for Li+ H 2 collisions in the calibration of a cold-atom vacuum standard, Physical Review A **99**, 42704 (2019).

[77] F. G. Huber, A Novel Apparatus for Experiments with Ultracold Fermions, Doctoral dissertation, 2009.

[78] H. J. Lewandowski, *Coherences and Correlations in an Ultracold Bose Gas* (University of Colorado at Boulder, 2002).

[79] A.-M. Brass and J. Chêne, Hydrogen uptake in 316L stainless steel: Consequences on the tensile properties, Corrosion science **48**, 3222 (2006).

[80] J. A. Fedchak, J. K. Scherschligt, S. Avdiaj, D. S. Barker, S. P. Eckel, B. Bowers, S. O'Connell, and P. Henderson, Outgassing rate comparison of seven geometrically similar vacuum chambers of different materials and heat treatments, Journal of Vacuum Science & Technology B **39**, (2021).

[81] *Material Properties of Titanium Grade 2*, https://www.matweb.com/search/datasheet.aspx?MatGUID=24293fd5831941ec9fa01dce994973c7&ckck=1.

[82] *Material Properties of 316ln Stainless Steel*, https://www.matweb.com/search/datasheet.aspx?MatGUID=3a413dabd215462da3408e6e8b761349.

[83] X. Baillard, M. Fouché, R. Le Targat, P. G. Westergaard, A. Lecallier, Y. Le Coq, G. D. Rovera, S. Bize, and P. Lemonde, Accuracy evaluation of an optical lattice clock with bosonic atoms, Optics Letters **32**, 1812 (2007).

[84] X. Baillard et al., An optical lattice clock with spin-polarized 87 Sr atoms, The European Physical Journal D **48**, 11 (2008).

[85] H. Miyake, N. C. Pisenti, P. K. Elgee, A. Sitaram, and G. K. Campbell, Isotope-shift spectroscopy of the S 0 1⟶ P 1 3 and S 0 1⟶ P 0 3 transitions in strontium, Physical review research **1**, 33113 (2019).

[86] A. V. Taichenachev, V. Yudin, C. Oates, C. Hoyt, Z. Barber, and L. Hollberg, Magnetic Field-Induced Spectroscopy of Forbidden Optical Transitions<? format?> with Application to Lattice-Based Optical Atomic Clocks, Physical review letters **96**, 83001 (2006).

[87] I. S. Madjarov, *Entangling, Controlling, And Detecting Individual Strontium Atoms in Optical Tweezer Arrays* (California Institute of Technology, 2021).

[88] D. J. Griffiths, *Introduction to Electrodynamics* (Cambridge University Press, 2023).

[89] C. A. Manogue and T. Dray, *Wire*, in *The Geometry of Static Fields* (Oregon State University, 2025).

[90] F. Kiesel, K. Karpov, A. De Martino, J. Auch, and C. Gross, Long-term stable laser injection locking for quasi-CW applications, EPJ Techniques and Instrumentation **12**, 1 (2025).

[91] R. Waxier and G. Cleek, Refractive indices of fused silica at low temperatures, Journal of Research of the National Bureau of Standards. Section A, Physics and Chemistry **75**, 279 (1971).

[92] C. Weitenberg, Single-Atom Resolved Imaging and Manipulation in an Atomic Mott Insulator, Doctoral dissertation, 2011.

[93] A. Kaufman, Laser-cooling atoms to indistinguishability: Atomic Hong-Ou-Mandel interference and entanglement through spin-exchange, Ph. D. Thesis (2015).

[94] Z. Zhang, *Phd Thesis in Preparation*, (unpublished).

[95] J. Samland, S. Bennetts, C.-C. Chen, R. G. Escudero, F. Schreck, and B. Pasquiou, Optical pumping of $5s4d\,^1D_2$ strontium atoms for laser cooling and imaging, Physical Review Research **6**, 13319 (2024).

[96] N. Schlosser, G. Reymond, and P. Grangier, Collisional blockade in microscopic optical dipole traps, Physical review letters **89**, 23005 (2002).

[97] D. Tsevas, Precise Shaping of Coherent Laser Beams with a Liquid-Crystal-on-Silicon Spatial Light Modulator, Doctoral dissertation, 2021.

[98] P. Osterholz, Freely Configurable Holographic Trap Arrays for the Trapping of Single Atoms, Doctoral dissertation, 2020.

[99] E. Hecht, *Optics 2nd Edition* (1987).

[100] I. Bloch, J. Dalibard, and S. Nascimbene, Quantum simulations with ultracold quantum gases, Nature Physics **8**, 267 (2012).

[101] I. Bloch, J. Dalibard, and W. Zwerger, Many-body physics with ultracold gases, Reviews of modern physics **80**, 885 (2008).

[102] P. Zupancic, P. M. Preiss, R. Ma, A. Lukin, M. Eric Tai, M. Rispoli, R. Islam, and M. Greiner, Ultra-precise holographic beam shaping for microscopic quantum control, Optics express **24**, 13881 (2016).

[103] A. Jenkins, J. W. Lis, A. Senoo, W. F. McGrew, and A. M. Kaufman, Ytterbium nuclear-spin qubits in an optical tweezer array, Physical Review X **12**, 21027 (2022).

[104] M. Schroeder, Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (Corresp.), IEEE transactions on Information Theory **16**, 85 (1970).

[105] C. Maier, A quantum simulator with long strings of trapped ions, (2020).

[106] S. Ma, A. P. Burgers, G. Liu, J. Wilson, B. Zhang, and J. D. Thompson, Universal gate operations on nuclear spin qubits in an optical tweezer array of Yb 171 atoms, Physical Review X **12**, 21028 (2022).

[107] S. B. Nagel, Ultracold Collisions in Atomic Strontium, Doctoral dissertation, 2008.

[108] S. Stellmer, F. Schreck, and T. C. Killian, Degenerate quantum gases of strontium, Annual Review of Cold Atoms and Molecules 1 (2014).

[109] J. hyun Annie Park, Towards Quantum Simulations with Strontium in Cavity-Enhanced Optical Lattices, Doctoral dissertation, 2021.

[110] M. Brown, T. Thiele, C. Kiehl, T.-W. Hsu, and C. Regal, Gray-molasses optical-tweezer loading: controlling collisions for scaling atom-array assembly, Physical Review X **9**, 11057 (2019).

[111] M. Takamoto, F.-L. Hong, R. Higashi, and H. Katori, An optical lattice clock, Nature **435**, 321 (2005).

[112] J. Ye, H. Kimble, and H. Katori, Quantum state engineering and precision metrology using state-insensitive light traps, science **320**, 1734 (2008).

[113] J. D. Jackson and R. F. Fox, Classical electrodynamics, (1999).

[114] R. Taïeb, R. Dum, J. Cirac, P. Marte, and P. Zoller, Cooling and localization of atoms in laser-induced potential wells, Physical Review A **49**, 4876 (1994).

[115] V. V. Ivanov and S. Gupta, Laser-driven Sisyphus cooling in an optical dipole trap, Physical Review A—Atomic, Molecular, and Optical Physics **84**, 63417 (2011).

[116] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters, The Journal of chemical physics **76**, 637 (1982).

[117] M. Norcia, A. Young, and A. Kaufman, Microscopic control and detection of ultracold strontium in optical-tweezer arrays, Physical Review X **8**, 41054 (2018).

[118] J. Dalibard and C. Cohen-Tannoudji, Laser cooling below the Doppler limit by polarization gradients: simple theoretical models, Journal of the Optical Society of America B **6**, 2023 (1989).

[119] M. Gehm, K. O'hara, T. Savard, and J. Thomas, Dynamics of noise-induced heating in atom traps, Physical Review A **58**, 3914 (1998).

[120] N. Zheng, W. Liu, J. Wu, Y. Li, V. Sovkov, and J. Ma, *Parametric Excitation of Ultracold Sodium Atoms in an Optical Dipole Trap*, in *Photonics*, Vol. 9 (2022), p. 442.

[121] C. Monroe, W. Swann, H. Robinson, and C. Wieman, Very cold trapped atoms in a vapor cell, Physical Review Letters **65**, 1571 (1990).

[122] F. Gyger, M. Ammenwerth, R. Tao, H. Timme, S. Snigirev, I. Bloch, and J. Zeiher, Continuous operation of large-scale atom arrays in optical lattices, Physical Review Research **6**, 33104 (2024).

[123] M. R. Lam, N. Peter, T. Groh, W. Alt, C. Robens, D. Meschede, A. Negretti, S. Montangero, T. Calarco, and A. Alberti, Demonstration of quantum brachistochrones between distant states of an atom, Physical Review X **11**, 11035 (2021).

[124] P. Carruthers and M. Nieto, Coherent states and the forced quantum oscillator, American Journal of Physics **33**, 537 (1965).

[125] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, Experimental issues in coherent quantum-state manipulation of trapped atomic ions, Journal of research of the National Institute of Standards and Technology **103**, 259 (1998).

[126] K. Mours, *Phd Thesis in Preparation*, (unpublished).

[127] I. S. Madjarov, J. P. Covey, A. L. Shaw, J. Choi, A. Kale, A. Cooper, H. Pichler, V. Schkolnik, J. R. Williams, and M. Endres, High-fidelity entanglement and detection of alkaline-earth Rydberg atoms, Nature Physics **16**, 857 (2020).

[128] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Randomized benchmarking of quantum gates, Physical Review A—Atomic, Molecular, and Optical Physics **77**, 12307 (2008).

[129] B. Dirkse, J. Helsen, and S. Wehner, Efficient unitarity randomized benchmarking of few-qubit Clifford gates, Physical Review A **99**, 12315 (2019).

[130] R. Finkelstein, R. B.-S. Tsai, X. Sun, P. Scholl, S. Direkci, T. Gefen, J. Choi, A. L. Shaw, and M. Endres, Universal quantum operations and ancilla-based read-out for tweezer clocks, Nature **634**, 321 (2024).

[131] S. Bourdeauducq et al., *ARTIQ 1.0*, https://doi.org/10.5281/zenodo.51303.

[132] *Real-Time Systems by Adwin*, https://www.adwin.de/us/start/echtzeit.html.

[133] P. P. Kulik, Controller Module for Real-Time Quantum Applications, 2019.

[134] M. Oliv, A. Matic, T. Messerer, and J. M. Lorenz, Evaluating the impact of noise on the performance of the variational quantum eigensolver, arXiv preprint arXiv:2209.12803 (2022).

[135] A. Matic, M. Monnet, J. M. Lorenz, B. Schachtner, and T. Messerer, *Quantum-Classical Convolutional Neural Networks in Radiological Image Classification*, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2022), pp. 56–66.

[136] J. Muniz et al., Repeated ancilla reuse for logical computation on a neutral atom quantum computer, arXiv preprint arXiv:2506.09936 (2025).

[137] S. Kukita, H. Kiya, and Y. Kondo, General off-resonance-error-robust symmetric composite pulses with three elementary operations, Physical Review A **106**, 42613 (2022).

[138] I. Trond et al., Quantum error correction below the surface code threshold, Nature **638**, 27 (2025).

[139] J. Viszlai, S. F. Lin, S. Dangwal, J. M. Baker, and F. T. Chong, An architecture for improved surface code connectivity in neutral atoms, arXiv preprint arXiv:2309.13507 (2023).

[140] S. Sunami, A. Goban, and H. Yamasaki, Transversal Surface-Code Game Powered by Neutral Atoms, arXiv preprint arXiv:2506.18979 (2025).

[141] S. Jandura and G. Pupillo, Surface code stabilizer measurements for Rydberg atoms, arXiv preprint arXiv:2405.16621 (2024).

[142] P. Kos and G. Styliaris, Circuits of space and time quantum channels, Quantum **7**, 1020 (2023).

Acknowledgments

Before I even entered kindergarten, I riddled my parents with all sorts of philosophical and scientific questions. I was highly interested in astronomy. I remember a large poster next to my bed with information on every planet and the first significant achievements in space travel, which I must've read every night before sleeping. I always wanted to understand how everything works and why things are the way they are. At some point, my parents succumbed to my pressure and told me I needed to study physics if I wanted to know these things. As I was only five years old back then, I had some waiting ahead of me. I never lost interest in studying physics; in fact, as time went on, I was more eager to start the process.

This thesis marks a significant milestone in my life. Even though I finished my master's degree in 2020 already, I felt like the study could only be complete if I managed to contribute to science. I am thrilled to have been given this opportunity to learn more about the world, to never stop asking questions, and to stay curious.

This journey has only been possible due to the support from people around me. I want to thank the teachers from my school who endured my lack of attention but still believed I should be allowed to continue. I want to thank all my friends from Heidelberg, with whom I spent most of my time back then. I enjoyed my time at the MPQ, which was only possible thanks to my peers. In fact, I owe a special thanks to Karsten, with whom I did a HiWi. He taught me not just electronics but also introduced me to the institute. Many thanks also to Anton, who taught me mechanics and who was always available for a good chat when times were tough. Also, thanks to Felix, who helped make this experiment a reality. Together with Anton, they fixed problems in our mechanical designs, helped us find new ideas, and constructed the models while assembling important parts accurately.

Initially, I continued my PhD at Christian Groß's potassium experiment, which moved from the MPQ to the University of Tübingen. He is a very caring group leader and made an effort to join us during lunch. I really enjoyed working with and for him. Unfortunately, my mental state didn't allow me to stay in Tübingen, and so I had to move back to Munich. Christian couldn't have handled the situation better; he was very

supportive, even though I had left a hole in the group, and I will always be grateful for his understanding.

It was enjoyable building this experiment, even though it wasn't always easy. My colleagues and I went through all the hard times. Thanks, Lorenzo, with whom I have worked for six years. He is always fun to be around and has lots of great insights. With Dimitrios, I was able to communicate freely, and we had a good time together in Arizona. I always welcomed Max's input when we encountered roadblocks, and he managed to offer new viewpoints. Zhao has great focus and always makes an effort to help when needed. I performed many measurements with Kevin, who has a lot of drive, and he helped me continue working when my motivation started to fade. I want to thank him for taking the time and effort to take good pictures of the experiment. Most other experiments were performed with Eran, who always had great insights and helped me structure measurements so that we wouldn't get lost.

We couldn't have made many decisions in the experiment without our group leaders. Thanks, Sebastian, who shared his experiences and steered us in the right direction. Johannes was always available for personal issues, and he made sure to check in the lab to help navigate complex experiments. Andrea has lots of great ideas, and it was fun to brainstorm new technologies or ways to move forward. And many thanks to Immanuel, who enabled this whole journey, and his experience helped push the experiment to where it is now.

My family is a great support network, and I wouldn't have managed this milestone without them. My parents always listened to my interests and supported me in every decision I made. My siblings were always a good resource for communication and, in general, for fun activities. Of course, my grandparents, great-grandmother, aunts, uncles, cousins, and niece all have their share in my life, and family events wouldn't be complete without them. Finally, I want to thank Melli, who had to endure me during my most difficult times. She was always there for me when I needed her most and was understanding when I came home feeling drained and unmotivated. I will always be grateful to everyone who helped me become a scientist.