

Master's Thesis

State Preparation and Readout Routine for Molecular Ions

Oliver Stadler

Quantum Engineering Master Program
Trapped Ion Quantum Information Group
Swiss Federal Institute of Technology (ETH) Zurich

Supervision:

David Holzapfel
Dr. Daniel Kienzler
Prof. Dr. Jonathan Home

Zürich, September 2023



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

First name(s):

| | |
|--|--|
| | |
| | |
| | |
| | |

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Signature(s)

| | |
|--|--|
| | |
| | |
| | |
| | |

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Abstract

The hydrogen molecular ion is the simplest molecule in our universe. At the same time, it is one of a few molecules that can be calculated to very high precision [SK05]. These two properties make it an interesting test bed to measure fundamental constants and to test new and existing theories. Even though the molecule is simple, gaining control over it is not.

This thesis discusses the design of improved and more flexible sideband cooling routines and their implementation using the M-ACTION experiment control system. Simulations are presented and experimental results are discussed.

State initialization of H_2^+ hyperfine states was further investigated. The development of a pumping sequence, that works without optical transitions within H_2^+ , is discussed. The pumping sequence makes use of quantum logic spectroscopy (QLS) as outlined in [SRL⁺05]. The algorithm developed to find such a sequence is presented and explained. Finally, simulation results and lessons learned are presented.

Acknowledgments

First, I would like to thank Prof. Jonathan Home for the opportunity to do my Master's thesis on an experiment conducting cutting-edge research and for allowing me to become a part of this amazing group of people that is the TIQI group.

I want to thank David Holzapfel for all the time we spent struggling, thinking, and debugging together. You always gave me the feeling that my work is meaningful and that we are moving forward, despite the experiment sometimes deciding otherwise.

Thanks to Nick Schwegler and Fabian Schmid for making the countless hours in the lab fun and enjoyable. I will miss our occasional rants about Ionizer.

Thank you also to Daniel Kienzler. The discussions we had were invaluable to me. You taught me a lot without ever giving me the feeling that I couldn't come up with the solution on my own.

A big thank you to Martin Stadler and Alexander Ferk for all the hours you spent debugging my code and teaching me the ways of `ionpulse_sdk`. I also want to thank you for the discussion about the pumping algorithm we had after my group meeting presentation. Your inputs are now an essential part of the algorithm presented in the H_2^+ pumping section.

Finally, I would like to thank every member of the TIQI group for being nice and welcoming from the start and having a great group culture that is maintained and valued by every member. Thank you, especially for letting me and other students take part in your group activities like the skiing weekend and the summer retreat.

Introduction

This thesis is a compilation of the work done in the last six months during my Master’s thesis in the TIQI group at ETH Zurich.

The theory in chapter 1 offers a very high-level view of how the trap works and some details about the ion species we are trapping. We will also go through some of the math required to understand Rabi oscillations and motional sidebands. These are essential to understanding sideband cooling of the ion’s motion. We will also discuss a novel technique called quantum logic spectroscopy (QLS), which will come in handy in the last chapter, where we discuss state initialization of the H_2^+ hyperfine structure.

In chapter 2 we introduce the control system developed within the TIQI group. All of the pulse sequences discussed in this work were implemented for this system.

Chapter 3 discusses the implementation of new and more flexible sideband cooling routines. The existing code of the molecules experiment was refurbished and new features were implemented to make it more resource-efficient.

Chapter 4 is about an extension of the sideband cooling process that allows cooling of the radial modes that otherwise cannot be cooled due to limitations in the setup.

Chapter 5 discusses state initialization of H_2^+ using a newly developed pumping sequence. We will provide motivation and an overview of the technique. The heart of this chapter is the algorithm that was developed to find an optimal pumping sequence. The sequences generated by the algorithm were simulated, and the simulation results will be shown and discussed.

Contents

| | | |
|----------|---|-----------|
| 1 | Theory | 4 |
| 1.1 | Paul Trap | 4 |
| 1.2 | Be ⁺ Ion | 5 |
| 1.3 | Atom - Light Interaction | 7 |
| 1.4 | Motional Sidebands | 10 |
| 1.5 | Sideband Cooling | 12 |
| 1.6 | H ₂ ⁺ Ion | 14 |
| 1.7 | Quantum Logic Spectroscopy | 16 |
| 2 | Experimental Setup | 17 |
| 2.1 | Control System | 17 |
| 2.1.1 | Sequencer Capabilities | 18 |
| 2.1.2 | Memory Managment | 20 |
| 2.1.3 | DDS Cache | 20 |
| 3 | Sideband Cooling | 22 |
| 3.1 | Overview | 22 |
| 3.1.1 | Lamb Dicke Parameter | 23 |
| 3.1.2 | Debye-Waller Factor | 23 |
| 3.1.3 | Heating | 25 |
| 3.2 | Implementation | 26 |
| 3.3 | Simulation | 27 |
| 3.3.1 | COM Cooling | 27 |
| 3.3.2 | STR Cooling | 27 |
| 3.3.3 | STR Cooling with Interleaved COM Pulses | 27 |
| 3.4 | Results | 29 |
| 3.4.1 | COM Cooling | 29 |
| 3.4.2 | STR Cooling | 30 |
| 4 | Radial Exchange Cooling | 32 |
| 4.1 | Overview | 32 |
| 4.2 | Implementation | 33 |
| 5 | H₂⁺ Pumping | 35 |
| 5.1 | Motivation | 35 |
| 5.2 | Overview | 36 |
| 5.3 | Algorithm | 39 |
| 5.3.1 | Assumptions | 39 |
| 5.3.2 | Implementation | 40 |
| 5.4 | Pumping Sequence | 42 |
| 5.5 | Simulation | 44 |
| 6 | Outlook | 50 |

Chapter 1

Theory

1.1 Paul Trap

An ion - an atom with more or less electrons than protons - can be trapped by a specific arrangements of electromagnetic fields [WMI⁺98]. A configuration of only static electric fields cannot achieve the desired trapping effect because the divergence of an electric field in free space is always zero. There are two common ways to get around this. The first option is to use an oscillating electric field in at least two of the three dimensions. If the parameters are chosen correctly, a net restoring force in all dimensions can be engineered. This arrangement is called a Paul trap. The other typical trap arrangement uses a static magnetic field and is called a Penning trap. The experiments in this thesis were conducted using a variant of the Paul trap called the linear Paul trap. The electrode arrangement of a linear Paul trap and the created potential can be seen in Figure 1.1. An image of the real trap used in the molecules experiment is shown in Figure 1.2.

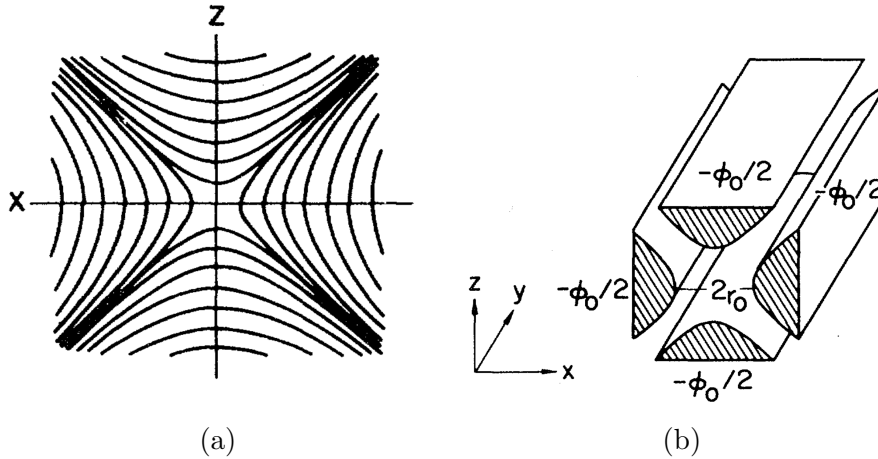


Figure 1.1: (a) Visualization of the potential created by the RF. Equipotential lines are drawn for a plane quadrupole field. (b) Electrode arrangement responsible for radial trapping. Images taken from [Pau90]

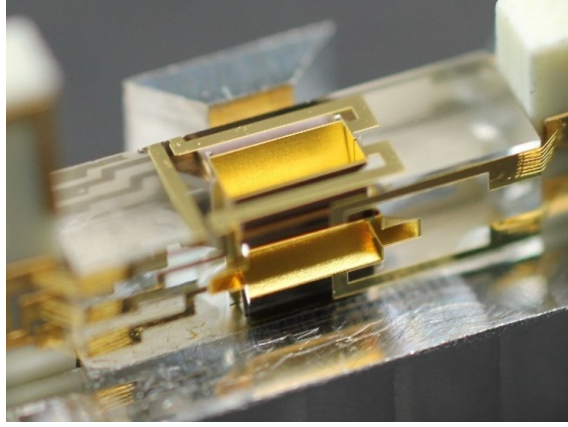


Figure 1.2: Linear Paul trap used in the molecules experiment. The trap structure is made out of glass. The electrical connections and the main electrodes are coated in gold. The big flat areas of gold in the middle of the trap are the electrodes shown in Figure 1.1b.

1.2 Be^+ Ion

Beryllium is the lightest alkaline-earth atom in the periodic table and the atom of choice for co-trapping H_2^+ in the molecules experiment. The low mass of ${}^9\text{Be}^+$ has multiple advantages. Most importantly, the mass of ${}^9\text{Be}^+$ being the same order of magnitude as the mass of H_2^+ , allows shared modes of motion to appear when both ions are trapped in the same potential. Having shared modes is essential for sympathetic cooling, as well as quantum logic spectroscopy. It allows quantum states to be transferred from H_2^+ to ${}^9\text{Be}^+$ [SHS⁺23][Hom13]. Another advantage is the ability to use higher trap frequencies, which reduces the average amount of quanta \bar{n} present in the motion after Doppler cooling. The level structure and some useful transitions are depicted in Fig. 1.3.

In the experiment, a constant magnetic field is used to lift the degeneracy of the hyperfine states. The states $|F = 2, m_F = 2\rangle = |\downarrow\rangle$ and $|1, 1\rangle = |\uparrow\rangle$ within the $S_{1/2}$ manifold are used as a qubit. Rabi oscillation between the two qubit states can be driven either by using a Microwave at 1.24 GHz (black dashed line) or a Raman transition by using two far detuned beams (solid purple lines) and a $P_{3/2}$ state. The qubit state can be read out via fluorescent photon scattering by driving the $S_{1/2} |2, 2\rangle \leftrightarrow P_{3/2} |3, 3\rangle$ transition. The $P_{3/2} |3, 3\rangle$ state has a short lifetime and only decays back into our lower qubit state. This process causes photons to scatter in a random direction. The photons can then be imaged using a camera or counted using a photomultiplier tube (PMT). We sometimes call the $|\downarrow\rangle$ state the bright state $|B\rangle$. Note that the $|\uparrow\rangle$ does not react to the driven readout transition and doesn't scatter photons. This state will appear dark on the camera and ideally shouldn't cause any photons to be counted by the PMT. This state is also called the dark state $|D\rangle$ because there are no photons scattered. The different scattering rates between the bright $|B\rangle$ and dark $|D\rangle$ is what allows us to perform a measurement of the qubit. This readout process causes the state to collapse and is usually the final step of a measurement. More details can be found in the supplemental material of [SHS⁺23], which covers the same experiment. An image of an ion in the bright state can be seen in Figure 1.4.

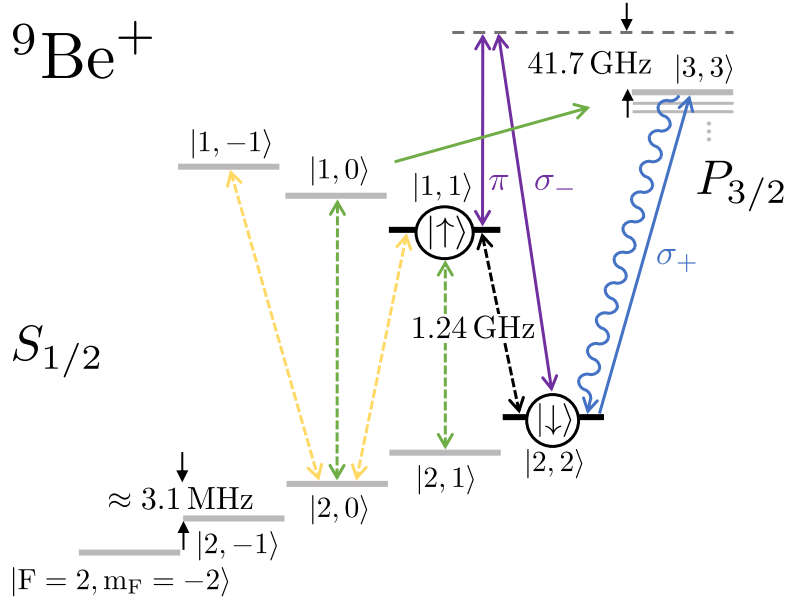


Figure 1.3: Energy level diagram of ${}^9\text{Be}^+$. The two states labeled $|\downarrow\rangle$ and $|\uparrow\rangle$ make up a qubit. The optical $|2, 2\rangle \leftrightarrow |3, 3\rangle$ transition is used for readout operations. This transition can only be driven if the system is in the $|2, 2\rangle$ state and there will be no photons scattered if the system is in the $|1, 1\rangle$ state. The yellow and green lines indicate repumpers used to move population back into the qubit states after photon scattering operations. These operations can occasionally move population out of the qubit states. Image by Nick Schwegler.

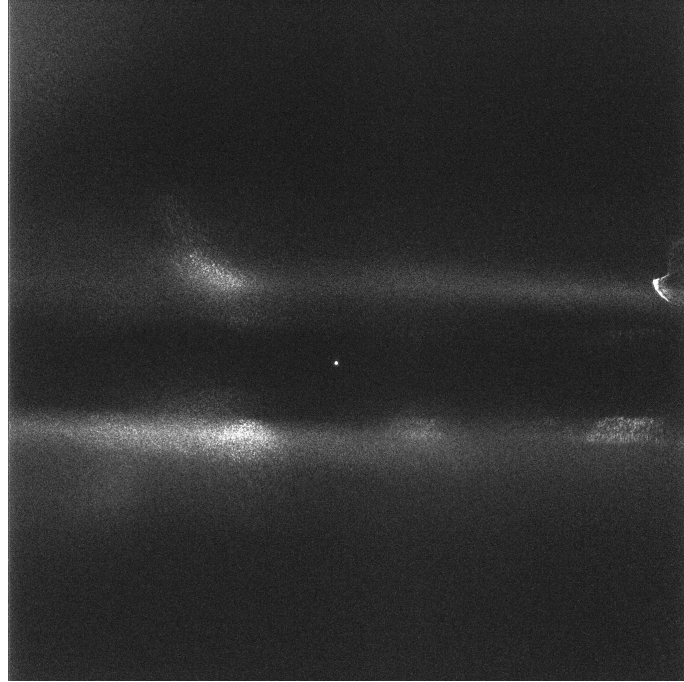


Figure 1.4: A single Be^+ ion in the bright state. When excited by the laser, it emits photons in a random direction. The trap is visible due to stray light coming from an imperfect laser beam shape or reflections in the chamber.

1.3 Atom - Light Interaction

To understand what happens when light (or any other electromagnetic wave) interacts with an atom, one needs to look at the evolution of the states under the created Hamiltonian. This section will follow a textbook approach about the interaction of a two-level system (TLS) with an electromagnetic wave based on [SZ97] and [Lou83]. The "atom" in this analysis will be a general two-level system. The result can be applied to any pair of states within an actual atom. The two states in question are the ground state $|g\rangle$ and the excited state $|e\rangle$, where the excited state $|e\rangle$ is the one with more energy. Using the dipole approximation¹, the Hamiltonian for an atom and a monochromatic, linearly polarized plane wave is given by:

$$\hat{H} = \hbar\omega_a |e\rangle \langle e| - \hat{\vec{d}}\vec{E}_0 \cos(\omega_l t) \quad (1.1)$$

with ω_a being the atomic transition frequency between the two states, $\hat{\vec{d}} = -e\hat{\vec{x}}$ the dipole operator depending on the electron charge and its position, E_0 the electric field amplitude and ω_l the frequency of the electromagnetic wave. Assuming the dipole operator to be real-valued, it can be rewritten to

$$\hat{\vec{d}} = d(|e\rangle \langle g| + |g\rangle \langle e|) \quad (1.2)$$

Where the fact that $\langle g|\hat{\vec{d}}|g\rangle = \langle e|\hat{\vec{d}}|e\rangle = 0$ was used. This is due to the parity flip of the position operator. Expressing the cosine using complex exponentials we get

$$\hat{H} = \hbar\omega_a |e\rangle \langle e| - \frac{1}{2}\hbar\Omega(|e\rangle \langle g| + |g\rangle \langle e|)(e^{i\omega_l t} + e^{-i\omega_l t}) \quad (1.3)$$

where $\Omega = \frac{\vec{d}\vec{E}_0}{\hbar}$ is the Rabi frequency. By applying a unitary transform

$$\hat{U} = e^{i\omega_a |e\rangle \langle e| t} \quad (1.4)$$

the Hamiltonian can be rewritten in the interaction picture. The calculation is left out for brevity reasons. In the interaction picture, the rotating wave approximation (RWA) can be applied, assuming $\omega_a + \omega_l \gg |\omega_a - \omega_l|$. Fast rotating terms are removed from the Hamiltonian because their effect can be neglected on timescales larger than $\frac{2\pi}{\omega_a}$. The unitary operation that got us into the interaction picture can then be reversed and we end up back in the Schrödinger picture. But the Hamiltonian now looks a bit different:

$$\hat{H} = \hbar\omega_a |e\rangle \langle e| - \frac{1}{2}\hbar\Omega(|e\rangle \langle g| e^{-i\omega_l t} + |g\rangle \langle e| e^{i\omega_l t}) \quad (1.5)$$

Let us now look at a generic state of this two-level system

$$\psi(t) = c_g(t) |g\rangle + c_e(t) |e\rangle \quad (1.6)$$

This state evolves according to the time-dependent Schrödinger equation

$$i\hbar \frac{\partial \psi(t)}{\partial t} = \hat{H} \psi(t) \quad (1.7)$$

¹If the wavelength is much larger than the extent of the electrons wavefunction around the nuclei, the spatial dependence of the electromagnetic field can be neglected

inserting 1.6 into 1.7 yields

$$i\hbar\dot{c}_g(t)|g\rangle + i\hbar\dot{c}_e(t)|e\rangle = \omega_a c_e(t)|e\rangle - \frac{1}{2}\hbar\Omega[e^{-i\omega_l t}c_g(t)|e\rangle e^{i\omega_l t}c_e(t)|g\rangle] \quad (1.8)$$

1.8 can be multiplied by $\langle g|$ and $\langle e|$ from the left side. This gives two coupled differential equations.

$$\begin{aligned} \dot{c}_g(t) &= -\frac{1}{2}i\Omega e^{i\omega_l t}c_e(t) \\ \dot{c}_e(t) &= -i\omega_a c_e(t) - \frac{1}{2}i\Omega e^{-i\omega_l t}c_g(t) \end{aligned} \quad (1.9)$$

These equations can be simplified by substituting

$$\begin{aligned} c'_g(t) &= c_g(t) \\ c'_e(t) &= e^{i\omega_a t}c_e(t) \end{aligned} \quad (1.10)$$

resulting in the new equations

$$\begin{aligned} \dot{c}'_g(t) &= -\frac{1}{2}i\Omega e^{i\Delta t}c'_e(t) \\ \dot{c}'_e(t) &= -\frac{1}{2}i\Omega e^{-i\Delta t}c'_g(t) \end{aligned} \quad (1.11)$$

where $\Delta = \omega_l - \omega_a$ is the detuning between the electromagnetic wave frequency and the atoms transition frequency. The equations can be solved by differentiating again with respect to t . Note that this uncouples the equations.

$$\begin{aligned} \ddot{c}'_g(t) - i\Delta\dot{c}'_g(t) + \frac{\Omega}{4}c'_g(t) &= 0 \\ \ddot{c}'_e(t) + i\Delta\dot{c}'_e(t) + \frac{\Omega}{4}c'_e(t) &= 0 \end{aligned} \quad (1.12)$$

These are just ordinary linear differential equations where the solutions are exponentials with the roots of the characteristic polynomial as exponents. For the excited state, the characteristic polynomial is

$$\lambda^2 + i\Delta\lambda + \frac{\Omega}{4} \quad (1.13)$$

and its roots are

$$\lambda_{1,2} = -\frac{i}{2}\Delta \pm \frac{i}{2}\sqrt{\Delta^2 + \Omega^2} \quad (1.14)$$

Therefore the ansatz for the solution is

$$c'_e(t) = \alpha' e^{-\frac{i}{2}\Delta + \frac{i}{2}\sqrt{\Delta^2 + \Omega^2}t} + \beta' e^{-\frac{i}{2}\Delta - \frac{i}{2}\sqrt{\Delta^2 + \Omega^2}t} \quad (1.15)$$

which can be simplified to

$$c'_e(t) = e^{-\frac{i}{2}\Delta t}[\alpha \cos \frac{\Omega'}{2}t + \beta \sin \frac{\Omega'}{2}t] \quad (1.16)$$

where $\Omega' = \sqrt{\Delta^2 + \Omega^2}$ is the effective Rabi rate. Using the initial conditions $c_g(0) = c'_g(0) = 1$ and $c_e(0) = c'_e(0) = 0$ we can solve for the free parameters α and β . Note: We used 1.11 to incorporate the c'_g initial condition.

$$\begin{aligned} c'_e(0) &= \alpha = 0 \Rightarrow \alpha = 0 \\ \dot{c}'_e(0) &= \beta \frac{\Omega'}{2} = -\frac{1}{2}i\Omega \Rightarrow \beta = -i\frac{\Omega}{\Omega'} \end{aligned} \quad (1.17)$$

Plugging α and β back into our ansatz, one is able to calculate the probability to be in the excited state at any time $P_e(t)$.

$$P_e(t) = |c_e^2(t)| = |c_e'^2(t)| = \frac{\Omega^2}{\Omega'^2} \sin^2 \frac{\Omega'}{2} t \quad (1.18)$$

The resulting sinusoidal dependence on time is called Rabi oscillations. See fig. 1.5 for a graphical representation. Over time, the population moves from the ground state to the excited state and back. This comes in very handy if one would like to change the state of an atom into another one. All that needs to be done is to turn on an electromagnetic field and wait for the right moment to stop. The length of time the field has to be turned on depends on the desired final state. If the goal is to just swap population between two states the time is $\tau_\pi = \pi/\Omega'$. Also known as a π -pulse. One can now see why Ω' is called the effective Rabi rate. It is the frequency of the oscillation. It increases with higher detuning. On the other hand, the oscillation's amplitude decreases and never completely reaches the excited state.

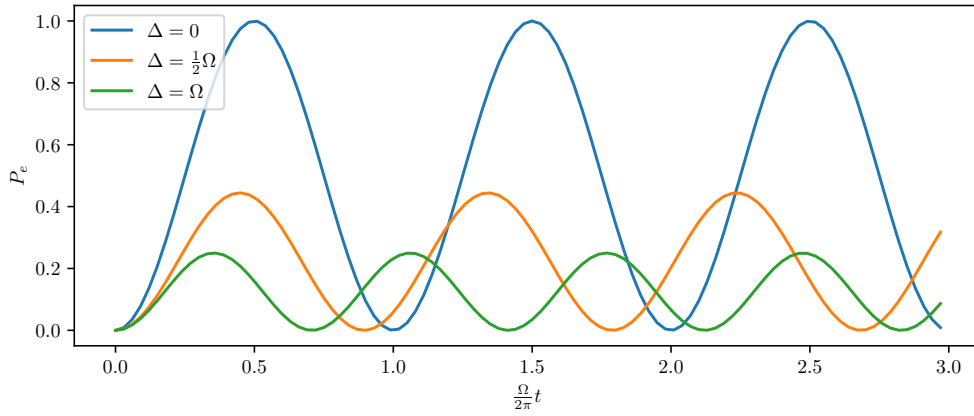


Figure 1.5: Rabi oscillations for different detunings, starting at the ground state $|g\rangle$. The y-axis shows the probability of finding the atom in the excited state after a certain amount of time while the electromagnetic field is turned on. For a perfect π -pulse we want our beam to be on resonance with the transition ($\Delta = 0$)(blue line) and stop after $t = \pi/\Omega'$. This is the case at 0.5 on the x-axis. One can see that stopping at that point in time gives us a probability of being in the excited state of $P_e = 1$.

1.4 Motional Sidebands

The Rabi oscillations shown in the previous section are not a distinct feature of atoms. Any quantum system - no matter how complex - does show this behavior when there is coupling between two states. An atom within a trapping potential is one of these more complex systems. While the atomic states and the spacial oscillation within the trap can most of the time be considered separate systems, they can be coupled using the right electromagnetic waves. This section is based on [Kie15] which in turn makes use of the work in [WMI⁺98].

The motion of the ion in the trap can be modeled very well as a quantum harmonic oscillator.

$$\hat{H}_{motion} = \hbar\omega_m \hat{a}^\dagger \hat{a} \quad (1.19)$$

with \hat{a} and \hat{a}^\dagger being the known destruction and creation operators and

$$\hat{z} = z_0(\hat{a} + \hat{a}^\dagger) \quad \text{with} \quad z_0 = \sqrt{\frac{\hbar}{2m\omega_m}} \quad (1.20)$$

where z_0 is the root-mean-square spread of the motional wavefunction in the ground state $|0\rangle_m$, which for ${}^9\text{Be}$ is $z_0 \approx 17 \text{ nm}$. The whole atom plus motion system has the combined Hamiltonian

$$\hat{H} = H_{atom} \otimes H_{motion} \quad (1.21)$$

When we looked at section 1.3 the dipole approximation was made very early on. The dipole approximation allows us to ignore the spatial dependence of the electromagnetic field if the wavelength is much larger than the electron - nuclei distance. This certainly holds for the electron in its orbital around the Be^+ ion's nucleus. But it doesn't necessarily hold for the whole ion moving inside the trapping potential. We will start by inserting the spatial dependence back into Equation 1.1. The Hamiltonian of the atom and motion together now reads:

$$\hat{H} = \hbar\omega_m \hat{a}^\dagger \hat{a} + \hbar\omega_a |e\rangle \langle e| - \hat{\vec{d}} \vec{E}_0 \cos(k\hat{z} - \omega_l t) \quad (1.22)$$

where k is the projection of the k -vector onto the ion's motional axis and \hat{z} is the ion's position. Note: The position of the electron around the nucleus (here chosen to be \hat{x} - part of \hat{d}) and the motion of the whole ion z are two different things. Following the same steps as in the beginning of chapter 1.3, we reach the Hamiltonian:

$$\hat{H}_a = \hbar\omega_a |e\rangle \langle e| - \frac{1}{2} \hbar\Omega (|e\rangle \langle g| + |g\rangle \langle e|) (e^{i(kz - \omega_l t)} + e^{-i(kz - \omega_l t)}) \quad (1.23)$$

Going again into the interaction picture using the unitary

$$\hat{U} = e^{i(\omega_m \hat{a}^\dagger \hat{a} + \omega_a |e\rangle \langle e|)t} \quad (1.24)$$

we get the new Hamiltonian

$$\hat{H}' = \hbar\Omega |e\rangle \langle g| e^{i\eta(\hat{a}e^{-i\omega_m t} + \hat{a}^\dagger e^{i\omega_m t})} e^{-i\Delta t} + \text{h.c.} \quad (1.25)$$

where $\eta = kz_0$ is called the Lamb-Dicke parameter. The Hamiltonian couples atomic and motional states if the detuning happens to be close to an integer multiple of the motional frequency ω_m . When taking a spectrum near the atomic resonance, these

integer multiples will show up as sidebands. The Rabi frequency of the coupling between the states n and $n - n_{sb}$ and swapping the atom's state is given by

$$\begin{aligned}\Omega_{n,n_{sb}} &= \Omega \left| \langle n | e^{i\eta(\hat{a}+\hat{a}^\dagger)} | n - n_{sb} \rangle \right| \\ &= \Omega e^{\frac{1}{2}\eta^2} \sqrt{\frac{(n - n_{sb})!}{n!}} \eta^{n_{sb}} L_{n-n_{sb}}^{(n_{sb})}(\eta^2)\end{aligned}\tag{1.26}$$

where $L_n^{(\alpha)}$ is the associated laguerre polynomial of degree n and order α . This formula will come in handy when calculating the π -times for sideband pulses. The relative magnitude of $\Omega_{n,n_{sb}}$ vs Ω is plotted in Figure 1.6.

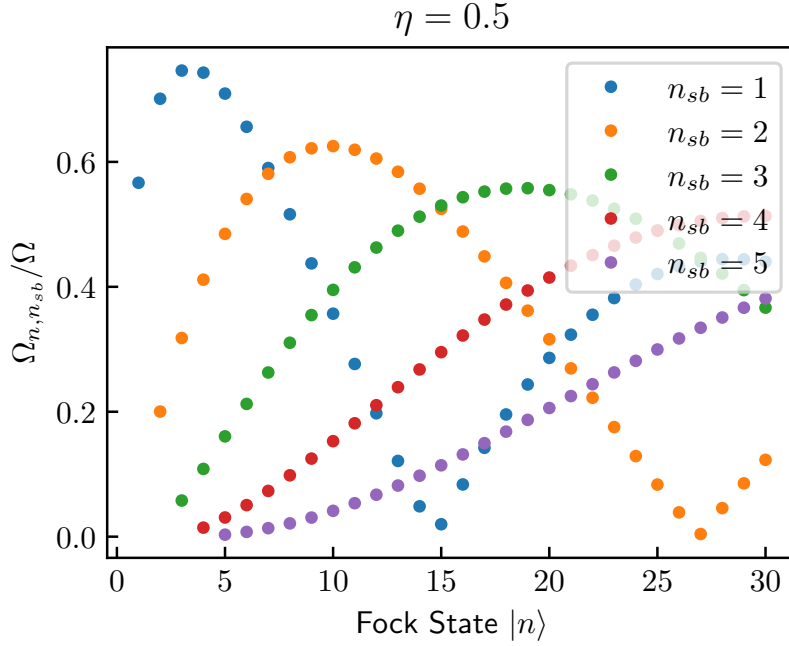


Figure 1.6: Relative magnitude of Rabi oscillation frequency for a transition between $|n\rangle$ and $|n - n_{sb}\rangle$ for $\eta = 0.5$. The Lamb-Dicke parameter is chosen arbitrarily, but similar to one of the modes within the setup. The key takeaways from this plot are not only that the Rabi frequency changes for every n and n_{sb} , but also that some curves get close to zero at some n . The Rabi frequency being close to zero leads to excessively long π -pulse times for a given state $|n\rangle$. During sideband cooling, we can move the population over these "traps" by using a different sideband. More on that later in chapter 3.

1.5 Sideband Cooling

The last section discussed sideband transitions that can couple motional states and qubit states. This section will go through a very important application that is a major part of this thesis. The application is called sideband cooling. Sideband cooling allows cooling the motion, or in other words, it can reduce the number of quanta within a given motional mode using a special sequence of laser and microwave pulses. We will first look at a diagram of the energy levels of our combined qubit/motion system depicted in Figure 1.7.

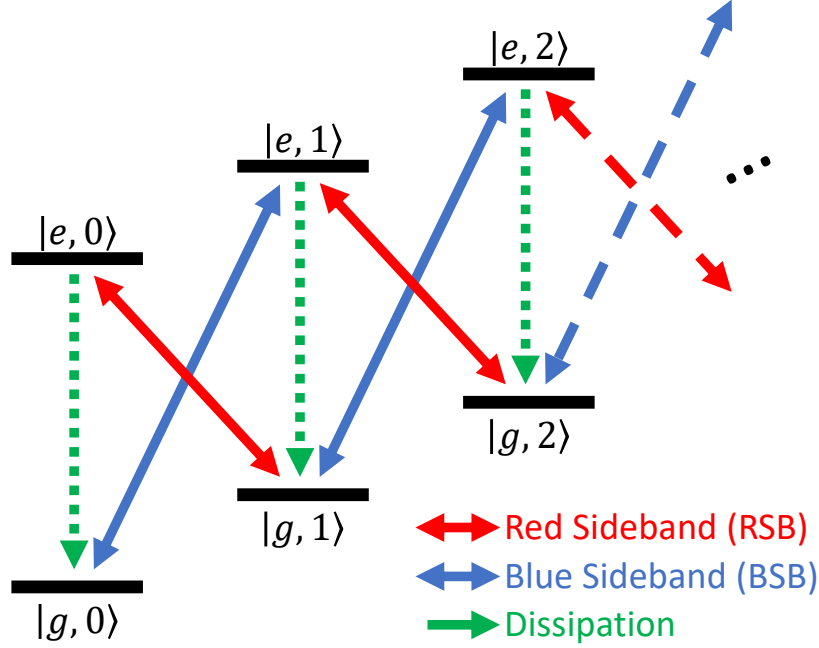


Figure 1.7: Energy level diagram of the combined atomic and motional states. The letters g and e refer to the ground and excited states of the atom. the numbers within brackets refer to the Fock state of the motion that is part of this combined state. Example: $|e, 1\rangle$ refers to the atom being in the excited state and the motion containing one quantum of energy. Carrier transitions on the atom and higher-order sideband transitions are not shown in the image. The red and blue lines show possible 1st sideband transitions. Dashed green lines show possible dissipation channels. Green lines involve the states being pumped into an intermediate state with a very short lifetime, that then decays into the atom's ground state g .

The states shown in the diagram are states of the combined system $\{|g\rangle, |e\rangle\} \otimes \{|n\rangle \mid n = 0, 1, 2, \dots\}$. Only the first sidebands are shown in the image. There are two kinds of sideband transitions. A sideband transition using higher frequency photons than the qubit's internal frequency is called a blue sideband. A sideband transition with a lower frequency is called a red sideband.

Sideband cooling makes use of red sideband transitions and dissipation channels to get from a random initial state of the motional mode into the motional ground state. One basic unit of a sideband cooling pulse sequence is the red sideband followed by repumping, where repumping includes dissipation as well as collecting the population that leaked out of the qubit subsystem. Applying this sequence to any state of the ladder in the atomic ground state $|g, n\rangle$ will first transfer the

population into the excited atomic state with one motional excitation less $|e, n - 1\rangle$. The second step will then cause the excited state to dissipate into the atomic ground state without changing the motional state, which results in the state $|g, n - 1\rangle$. One iteration of this basic sequence allows all the population present in the atomic ground states of the ladder to be moved to a state with one motional excitation less. Multiple iterations can be performed to move the motional population further down the ladder. Once the population reaches the ground state, the pulses do not apply anymore. This causes the ground state population to grow until all of the population is collected or the sequence has to be stopped for time reasons.

Note that this process wouldn't work without the dissipation part. If instead alternating red and blue sidebands were used to move down the ladder, then any population reaching the motional ground state would miss either a red or a blue sideband and then start going the opposite way up the ladder. Another explanation for why cooling cannot work using only sideband pulses is entropy. Sideband pulses are unitary transformations on the qubit/motion system. Unitary operations can neither reduce nor increase the entropy of a quantum system. But cooling requires a process that can lower the entropy by ejecting it out of the system. The final goal of sideband cooling is to reach a pure ground state. But a pure state has an entropy of 0 whereas the entropy of a thermal state is > 0 . Unitary operations alone can never get from a thermal state to a ground state.

1.6 H_2^+ Ion

The hydrogen molecular ion consists of just two protons and one electron. It offers a challenge to students being new to the field of quantum mechanics and at the same time, it offers rich physics to still be explored even today. It has been shown that H_2^+ might be an excellent candidate for molecular clocks [Kar14][SRL⁺05]. It might also be used to measure physical constants on par with other leading experiments [KHKK16][SK05].

The Schrödinger equation for the H_2^+ was already solved as early as 1927 [Bur27]. It was shown that this molecule indeed has a stable electronic configuration in its ground state - as expected from the fact that these molecules exist in nature. A short derivation can be found here [Fit10]. The old way to simplify the calculations was to use the Born-Oppenheimer approximation² [BO27]. Modern numeric approaches have solved the Schrödinger equation for the 3-body coulomb problem to basically arbitrary precision [SK05]. But they do not take relativistic corrections into account, which are now limiting the precision. The energy level diagrams with all its contributions can be seen in Figure 1.8. The contributions, sorted from high

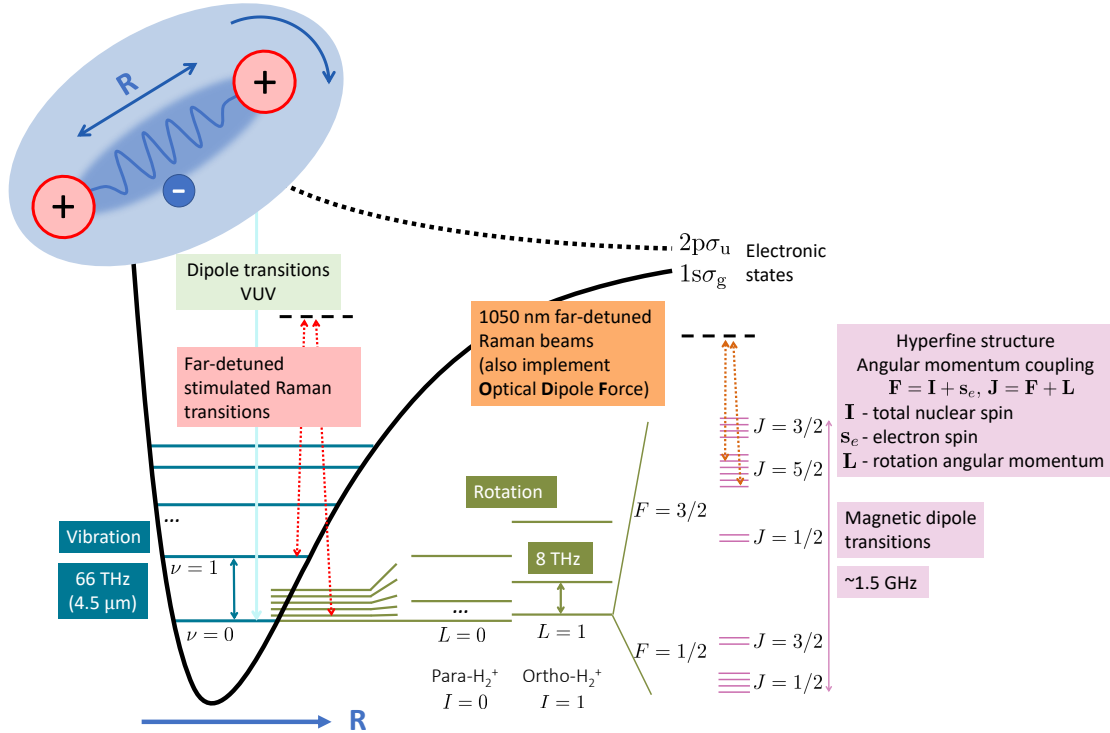


Figure 1.8: Energy level diagram of H_2^+ . Only the electronic ground state is interesting here because even the first excited electronic state (dashed black line) has no minimum in energy for any finite nuclei distance R , and is therefore not a stable state. Para and ortho states of the rotation can be discriminated by applying an optical dipole force (ODF)(orange lines), which applies with different strengths to para and ortho. Image by Fabian Schmid, Nick Schwegler and David Holzapfel

to low, are electronic > vibration > rotation > hyperfine structure. The vibration

²Under the Born-Oppenheimer approximation it is assumed that relatively heavy objects like protons move on a very different timescale compared to the lighter electrons. The Schrödinger equation can then be solved assuming fixed positions of the nuclei, which simplifies the equations a lot.

(quantum number ν) and rotation (quantum number L) are in combination sometimes called *rovibrational* states. From the electronic energy levels in Figure 1.8 one of the challenges of trapping H_2^+ becomes visible. There is no optical transition within H_2^+ that could be used for cooling or detection and that doesn't disassociate (break apart) the molecule since the higher energy states are unbound. This can be seen from the excited energy level only having a minimum at $R \rightarrow \infty$. The only way to cool this ion to the motional ground state is via sympathetic cooling³ by co-trapping another ion that has an optical transition - like Be .

At the usual temperature within the experiments cryostat of 10 K the average energy is

$$\omega_{thermal} = \frac{10 \text{ k}_B}{\hbar} \approx 208 \text{ GHz} \quad (1.27)$$

This is much smaller than the vibrational energy splitting of 66 THz and thus one might expect the system to be in the vibrational ground state in thermal equilibrium. But this is not the case. In the setup, H_2^+ is generated from electron-impact ionization of H_2 from rest gas within the vacuum chamber [SHS⁺23]. This process causes the vibration to end up in one of many vibrational states as shown by [KDH12]. Combining this with the lifetime of these states in the order of weeks [SHS⁺23], the need for another cooling process emerges. Buffer gas cooling⁴ [SKHV⁺17] is used by the experiment to prepare the vibrational ground state and either the $L = 0$ or $L = 1$ state of the rotation. One might ask why there are exactly two states after buffer gas cooling. And the answer is that the rotation is a bit more complicated. There exist basically two species of hydrogen molecular ions. There are the ones where the nuclear spin wavefunctions are symmetric (ortho) or antisymmetric (para). The conversion between the ortho and para ground states have been calculated to happen once every 6.4×10^5 years by [KB23]. The conversion can - for all intents and purposes - be neglected. This also means that there are no conversions happening after one H_2^+ ion falls into our trap. After loading, it has to be checked whether the loaded ion is para or ortho H_2^+ . If it is not the one required for the current experiment, it has to be ejected out of the trap and the trap has to be reloaded until the right species is present.

One of the goals of the experiment is to perform spectroscopy of the hyperfine structure of H_2^+ . Since para H_2^+ doesn't have a hyperfine structure, only the ground state of ortho $L = 1$ is of interest.

³The coulomb interaction allows two objects in the trap to exchange motional energy and they will thermalize over time. If one of them gets cooled, the other one will be cooled as well.

⁴Buffer gas cooling introduces another gas, like He in our case, into the trap that can collide with the H_2^+ ion and take some of the vibrational or rotational energy out of the system. More info in [SKHV⁺17].

1.7 Quantum Logic Spectroscopy

After talking about the issues in measuring states within H_2^+ , let us look at one of the possible solutions and the reason why H_2^+ has gotten much attention lately, namely Quantum Logic Spectroscopy (QLS) outlined in [SRL⁺05]. This technique allows to extract quantum information about one ion in a trap, called the spectroscopy ion, via a shared motional mode and another ion, called the logic ion. The logic ion should be one where state readout can be easily performed. The logic ion should also be similar in mass to the spectroscopy ion, such that the motion is coupled and information and heat exchange is possible. This allows transporting of information via the motional states. It also lets the ions exchange thermal energy. The spectroscopy ion can then be cooled via laser cooling of the logic ion. This process is called sympathetic cooling.

We will now have a deeper look at how quantum logic spectroscopy works. Looking at only a pair of states in the spectroscopy ion $\{|g\rangle_S, |e\rangle_S\}$ and the logic ion $\{|g\rangle_L, |e\rangle_L\}$, where e refers to the state with more energy, and the motional states in one of the shared modes $\{|n\rangle_M \mid n \in \{0, 1, 2, \dots\}\}$, the initial state is given by:

$$\psi_0 = |g\rangle_L \otimes |0\rangle_M \otimes \alpha |g\rangle_S + \beta |e\rangle_S \quad (1.28)$$

where the information we want to get is contained in α and β . We start by applying at red sideband (RSB) pulse between the motion and the spectroscopy ion. The red sideband can be written as $RSB = |0\rangle |e\rangle \langle 1| \langle g| + |1\rangle |g\rangle \langle 0| \langle e|$. Applying this operation to our initial state ψ_0 gives:

$$\psi_1 = |g\rangle_L \otimes \alpha |0\rangle_M + \beta |1\rangle_M \otimes |g\rangle_S \quad (1.29)$$

Now another red sideband can be applied, but this time between the logic ion and the motion. The resulting final state is:

$$\psi_2 = \alpha |g\rangle_L + \beta |e\rangle_L \otimes |0\rangle_M \otimes |g\rangle_S \quad (1.30)$$

The final state $\alpha |0\rangle_L + \beta |e\rangle_L$ of the logic ion can now be read out and the result will be exactly the state that the spectroscopy ion was in in the beginning. How can this be used for spectroscopy one might ask? Assuming that we do not know the transition frequencies of the spectroscopy ion exactly - which is the reason one would do spectroscopy in the first place - the frequency of the spectroscopy ions transition can be swept. If the transition frequency is hit, the spectroscopy qubit should undergo rabi oscillations. This change in state can then be read out by QLS. But QLS can not only be used for spectroscopy. It offers a general tool that can be used to get all the benefits of an atom with optical transitions and use them when looking at an ion without these transitions. In the rest of this work, we might refer to QLS as the QLS process or QLS procedure, hinting at the fact that the usage is not about spectroscopy. We will make great use of the QLS procedure in chapter 5.

Chapter 2

Experimental Setup

2.1 Control System

The M-ACTION experiment control system used by the TIQI group has been developed by many people over many years. The development of the core system is well documented in Vlad Negnevitsky's PhD Thesis [Neg27]. An overview of the whole system is depicted in Figure 2.1. We will only discuss the parts that are relevant for this work.

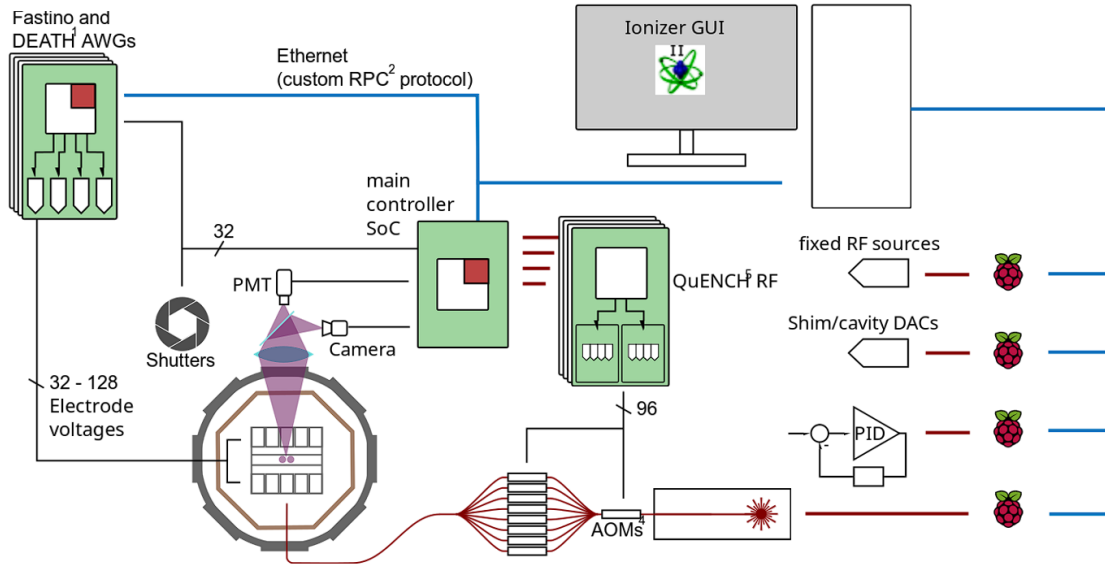


Figure 2.1: Experiment control system overview. The M-ACTION system uses a dedicated ethernet network and runs a custom remote procedure call (RPC) protocol on top (left side). Asynchronous devices like PID controllers managed by Raspberry Pis are connected to the lab network. M-ACTION is further described here [Neg27]. Image by Martin Stadler.

We will go through all the steps that the user and the system have to perform to complete a successful measurements. Everything starts with programming an experiment in C++ using the ionpulse software development kit (SDK). The process is shown in Figure 2.2. After compiling and uploading the executable (blue line) to the main controller System on Chip (SoC), the graphical user interface (GUI), called Ionizer, can be connected and will show all the programmed experiments and their parameters grouped within experiment pages. The experiment can be added to the queue of scheduled experiments by the user. Once the main controller

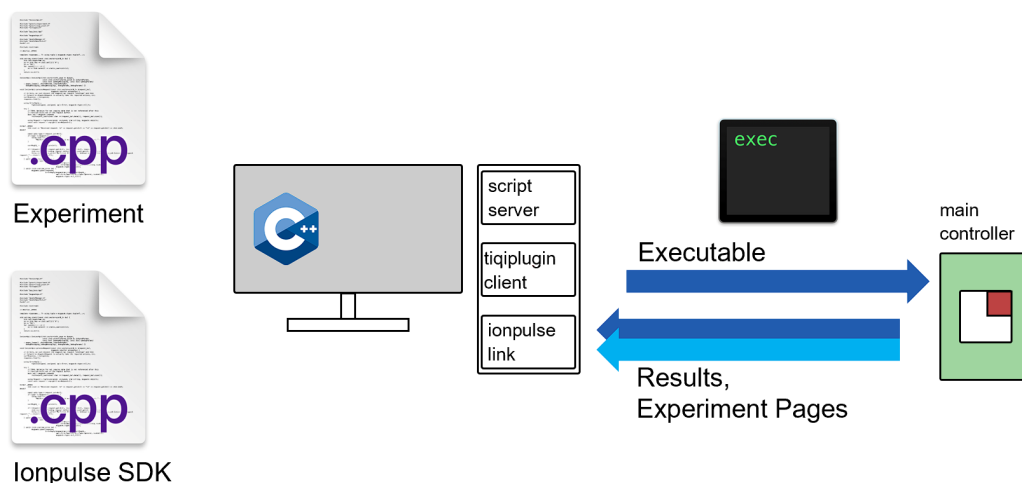


Figure 2.2: An experiment is written in C++ making use of the framework present in the ionpulse SDK. The experiment can read parameters from its generated experiment page inside the Ionizer GUI software. The compiled C++ code has to be uploaded to the main controller SoC to be available in Ionizer. Image by Alexander Ferk. C++ logo from [Fou23].

SoC reaches that experiment in the queue, its code will be run. The experiment code is now responsible for adding all the pulses it needs to a sequence. It can specify parameters like frequency, phase and amplitude for every pulse and also what channel that pulse should run on. TTLs can also be controlled in the same way and can be synchronized with the rest of the system. Once the pulse sequence is assembled, control is returned to the core code. The core will now upload the sequence to the FPGA’s internal memory blocks. Once everything is ready, the FPGA will start to play the sequence of pulses in a loop for a specified amount of iterations. These iterations are called *shots*. After every shot, the photomultiplier tube (PMT) signal is read out and the result is stored in a FIFO queue. The main controller will read this queue at the end of the experiment and send the resulting data to Ionizer. Ionizer will then plot the data and perform post-processing like curve fitting.

This was just a general overview. Over time many features were added to the way the pulse sequence gets compiled and interpreted. Some of these features will be discussed in the following subsection 2.1.1. There is also a small section about memory management in ionpulse SDK because a lot of time was spent optimizing the code to fit into the FPGA’s memory. More on that in subsection 2.1.2. Parts of the code developed for the molecules experiment even made it into the SDK. See subsection 2.1.3 for that.

2.1.1 Sequencer Capabilities

Under the hood, the system uses a custom instruction set, from which a pulse sequence can be assembled. The instruction set contains, for example, the loading of pulse information from memory or to jump to a specific point in the instruction sequence. This allows to implement loops and function calls. However, due to the lack of a stack, only a limited amount of nesting of functions and loops can be done. Even though this might seem limiting at first glance, a lot of sequences can

be implemented this way. Due to the loading of pulse information from a memory address, the same pulse can be reused many times. This saves a lot of memory since every pulse has to store at least frequency, phase, amplitude and duration. The instruction sequence is usually not manipulated by the user directly. Most users use the higher-level application programming interface (API) that is part of ionpulse SDK and let the SDK do the instructions assembly for them. The higher-level API was also used for this work and the core features it provides are listed here.

- **adding a pulse**

A pulse can be created by providing all the necessary information like frequency, phase, amplitude and time of the pulse and a pointer where the resulting pulse object should be stored. Creating another pulse with the exact same values and the same pointer again in the future will not create a new pulse but will rather be compiled into instructions that will reuse the same information. This drastically reduces the amount of memory required on the FPGA. More memory management in subsection 2.1.2.

- **loops**

Some usage patterns (e.g. sideband cooling) require a lot of pulses that have almost all parameters in common except a few (e.g. the duration or the amplitude). Loops and looped-settings were added to the API exactly for that case. A loop is defined by a set of instructions and a list of looped-settings that should change on every iteration of the loop. These settings need to provide a list of values with as many values as there are loop iterations. The SDK will then create pulse and jump instructions to form that loop. The changing values will be implemented by adding special loading instructions that override a certain parameter on every iteration. This way, no redundant information will be stored in the FPGA memory.

- **nested loops**

Nested loops offer even more flexibility in places where a simple loop is not enough. They consist of an outer loop and one or more inner loops. The inner loops do not only get one list of looped-settings but one list for every iteration of the outer loop. This feature is used for radial exchange cooling described in section 4.2.

- **forks**

Forks allow for two or more instruction sequences to be assembled where only one of them will be executed depending on some condition. The condition can either be a real-time readout of the PMT or it can be a deterministic decision that is stored somewhere in the FPGA's memory. Real-time forking can be used for quantum computing operations where it might be desirable to perform a measurement and then apply some gates depending on the result. Applications are for example state initialization or error correction. Deterministic forking only makes sense within a loop and the SDK only allows this usage pattern. The path that the fork will take is updated on every iteration the same way other looped-settings are updated. The decisions have to be precomputed by the user and another looped-settings object, that is linked to one of the forks, must be created and added to the list of looped-settings belonging to the loop. This feature is used in chapter 5.

2.1.2 Memory Managment

Every object stored in FPGA memory also has an object representing it in the CPU RAM. When the sequence is first assembled, these RAM objects are linked together to form a representation of the whole sequence. Then, in a second step, that representation is loaded into the FPGA memory. When creating RAM objects, the user has to provide a pointer to a pointer where the SDK can store the address of the object it creates. This pointer later allows to reuse the same object and therefore saves memory on the FPGA side. An example code snippet can be seen in Listing 2.1.

```
1 // globals
2 dds::FPA *fpa = nullptr;
3 dds::Time *time = nullptr;
4 dds::ttl_dds_pul *pulse = nullptr;
5
6 void simple_pulse_example(){
7     bp->use_fpa(&fpa, 2.0, 0.0, 50.0);
8     bp->use_time(&time, 10.0);
9     bp->make_cap(&pulse, fpa, time, ...); // _/"\ <-- cap pulse
10    bp->run(*pulse, ...);
11 }
```

Listing 2.1: Minimum code required to run a pulse. A 10 μ s long pulse with a frequency of 2 MHz, a phase of 0° and an amplitude of 50 % is created and added to the sequence. The object pointed to by **fpa** and **time** can be reused by other pulses.

The global pointers **fpa**, **time** and **pulse** do not necessarily have to be globals, but they are required to have a lifetime that matches or exceeds the end of the experiment. That means they either have to be globals/static variables or need to be provided by some other means. The currently most common usage pattern is to use globals with rather complicated names to avoid name clashes. It also creates issues if a function is run twice with different arguments and the pulses somehow depend on the arguments. For every set of parameters a new global would have to be used. But globals are fixed at compile time and their number can not increase at runtime. There have been workarounds by creating global containers that can grow at runtime or even creating globals for every possible combination of parameters that can be used. Both approaches led to overly complicated dependencies throughout the codebase and countless bugs. The core issue is, that the user code needs to care about lifetimes that are beyond the scope of his code. We will talk about the solution to this problem in the next section.

2.1.3 DDS Cache

To solve the problem mentioned in subsection 2.1.2 one has to think about what the user actually wants to do. The user's only goal is to create an object, get some kind of reference to that object, and then provide the reference to another function to make use of it. The user does not care about the lifetime of the created object. That is a problem for the API to solve in the background. To make sure that the same pulses, being used in different functions, still share the same underlying object, some kind of lookup is required. This not only allows us to share objects with the same value but makes it the default behavior, which is another nice feature. This new feature was named DDS Cache due to objects like `dds::FPA` or `dds::Time` inheriting from a class called `DDSBramObject` and due to them being stored in a structure that

returns the same objects during one experiment and then deletes all of them when no longer needed - basically acting like a cache.

The cache was implemented using a global `std::unordered_map<std::tuple, std::unique_ptr<T>>` where the `std::tuple` is used to lookup objects by values and `T` refers to the specific type of the objects stored. The lookup process works by creating a tuple instance of the values provided (e.g. frequency, phase, amplitude for a `dds::FPA`) and using that tuple as a key. This either returns a pointer to an existing object or creates a new empty pointer in the cache. The empty pointers address can then be passed to `use_fpa(&ptr, ...)` to create the object. Finally, a C++ reference to the created object is returned to the user. This reference can now be used to create more objects that depend on it. The same example code as in 2.1 but using the new feature is shown in 2.2

```
1 // No more globals :)
2
3 void simple_pulse_example_with_cache(){
4     dds::FPA& fpa = bp->fpa_cache.get_fpa(2.0, 0.0, 50.0);
5     dds::Time& time = bp->time_cache.get_time(10.0);
6     dds::ttl_dds_pul& pulse = bp->cap_cache.make_cap(fpa, time, ...);
7     bp->run(pulse, ...);
8 }
```

Listing 2.2: The same code as in listing 2.1 but this time using the new cache infrastructure built into the SDK. Note that the globals are gone and that the user now only has to provide values for the pulses.

The cache also has the advantage of reusing resources whenever they can be shared. A code example showing this feature is shown in listing 2.3.

```
1 void foo(){
2     dds::FPA& fpa1 = bp->fpa_cache.get_fpa(2.0, 0.0, 50.0);
3     dds::FPA& fpa2 = bp->fpa_cache.get_fpa(2.0, 0.0, 50.0);
4 }
5
6 void bar(){
7     dds::FPA& fpa = bp->fpa_cache.get_fpa(2.0, 0.0, 50.0);
8 }
9
10 void experiment(){
11     foo();
12     bar();
13 }
```

Listing 2.3: Example containing two function calls and a total of three calls to `get_fpa`. Only one `dds::FPA` object will be created during the execution of `experiment`. Code using the old usage pattern would usually create two objects - one per function. Note that sharing of these objects is safe because they have constant values.

The DDS Cache infrastructure was used for all new code written in the following chapters 3, 4 and 5.

Chapter 3

Sideband Cooling

3.1 Overview

The two-ion crystal has six degrees of motional freedom (ignoring ion internal motion). There are two axial modes and four radial modes. This section is focused on cooling the two axial modes. Cooling radial modes will be discussed in chapter 4. The first axial mode, the in-phase mode of motion, also called center of mass mode, is usually abbreviated as COM. A COM excitation involves both ions moving in the same direction within the trapping potential. The second axial mode, the out-of-phase mode is commonly called STR because of the stretching motion of the two ions in this mode.

The most commonly used form of quantum logic spectroscopy requires at least one of the motional modes of the ion crystal to be in the ground state. In our experiment, either the COM or STR mode can be used. After Doppler cooling, these modes are expected to be in a thermal state with $\bar{n}_{COM} \approx 10$ and $\bar{n}_{STR} \approx 2.4$ [SHS⁺23]. In the Fock basis, these thermal states populate a wide range of Fock states. Using red sideband pulses on our logic ion, we can move the population toward the lower-numbered states. The goal is to collect as much of the population as reasonably possible to get a high-fidelity ground state.

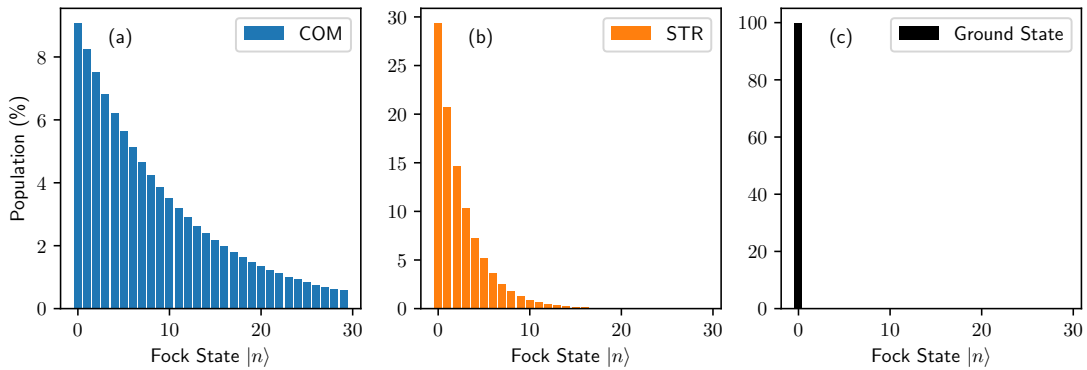


Figure 3.1: Motional states after doppler cooling for (a) COM $\bar{n} = 10$, (b) STR $\bar{n} = 2.4$. The ground state (c) is what we want to achieve.

There are some difficulties arising from the underlying physics. They are discussed in the following subsections.

3.1.1 Lamb Dicke Parameter

Looking back at equation 1.26 from chapter 1.4, the Rabi frequency for a sideband transition from Fock state n to $n - n_{sb}$ using the n_{sb}^{th} sideband with a Lamb-Dicke parameter η is proportional to

$$\Omega_{rabi}(n, n_{sb}, \eta) \propto e^{\frac{1}{2}\eta^2} \sqrt{\frac{(n - n_{sb})!}{n!}} \eta^{n_{sb}} L_{n-n_{sb}}^{(n_{sb})}(\eta^2) \quad (3.1)$$

where $L_n^{(\alpha)}$ is the associated laguerre polynomial of degree n and order α . This is important because we can now see that the Rabi frequency depends on the number of the Fock state we want to cool (n). This means we have to calculate the π -pulse time for every n given the sideband number n_{sb} and the η for the mode. The time for a π -pulse can be calculated as follows:

$$t_\pi(n, n_{sb}, \eta) = t_{\pi,1 \rightarrow 0,calibrated} \frac{\Omega_{rabi}(1, 1, \eta)}{\Omega_{rabi}(n, n_{sb}, \eta)} \quad (3.2)$$

The resulting t_π times for the first 5 sidebands and up to $n = 9$ are shown in Table 3.1. Additionally, the Rabi frequencies are plotted in figure 3.2.

| COM | $n = 1$ | $n = 2$ | $n = 3$ | $n = 4$ | $n = 5$ | $n = 6$ | $n = 7$ | $n = 8$ | $n = 9$ |
|--------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| $n_{sb} = 1$ | 1.0 | 0.84 | 0.83 | 0.88 | 0.98 | 1.15 | 1.41 | 1.85 | 2.67 |
| $n_{sb} = 2$ | - | 2.49 | 1.61 | 1.28 | 1.12 | 1.04 | 1.0 | 0.99 | 1.01 |
| $n_{sb} = 3$ | - | - | 7.62 | 4.14 | 2.85 | 2.2 | 1.82 | 1.57 | 1.41 |
| $n_{sb} = 4$ | - | - | - | 26.9 | 12.8 | 7.93 | 5.56 | 4.21 | 3.37 |
| $n_{sb} = 5$ | - | - | - | - | 106 | 45.7 | 25.8 | 16.7 | 11.8 |
| STR | $n = 1$ | $n = 2$ | $n = 3$ | $n = 4$ | $n = 5$ | $n = 6$ | $n = 7$ | $n = 8$ | $n = 9$ |
| $n_{sb} = 1$ | 1.0 | 0.71 | 0.58 | 0.51 | 0.46 | 0.42 | 0.39 | 0.37 | 0.35 |
| $n_{sb} = 2$ | - | 14.4 | 8.36 | 5.93 | 4.61 | 3.77 | 3.2 | 2.78 | 2.46 |
| $n_{sb} = 3$ | - | - | 255 | 128 | 81.0 | 57.4 | 43.5 | 34.5 | 28.2 |
| $n_{sb} = 4$ | - | - | - | 5205 | 2332 | 1349 | 885 | 627 | 468 |
| $n_{sb} = 5$ | - | - | - | - | 118764 | 48563 | 26000 | 15947 | 10648 |

Table 3.1: Pulse time t_π required for a π -pulse from state n to $n - n_{sb}$ for $t_{\pi,1 \rightarrow 0,calibrated} = 1$. $\eta_{COM} = 0.567$ and $\eta_{STR} = 0.098$.

Using table 3.1, we can create a pulse sequence with the right π -pulse times to - in theory - get all of the population from one Fock State to a Fock state with a lower number. Starting at a high number, all of the levels can be cleared one after the other. If everything works out, the system should end up in the ground state. Two example sequences are shown in Seq. 3.1.

3.1.2 Debye-Waller Factor

The six harmonic oscillators, describing our two ion crystal with six degrees of motional freedom, can only be considered independent to some degree. How they depend on one another is well explained in section 4.4.5 in the summary [WMI⁺98] by Wineland et al. In short: The actual Rabi rates, as seen by the experiment, are different from shot to shot and follow a certain probability distribution. The analysis by Windeland et al. looks at the mean and squared expected Rabi rate

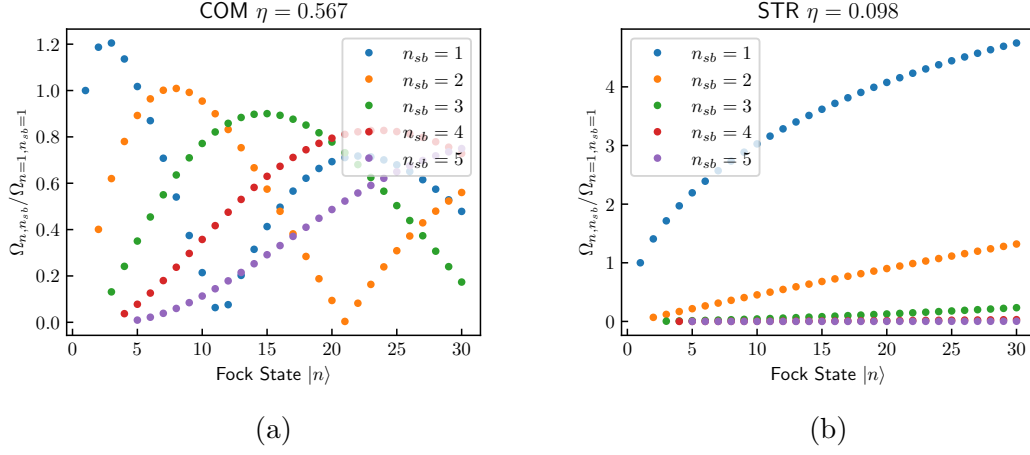
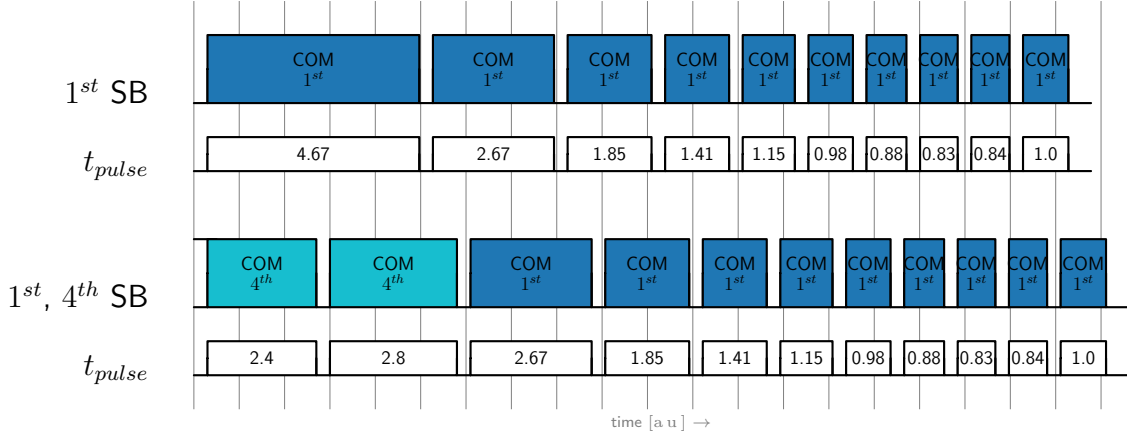


Figure 3.2: Rabi frequencies for different Fock states and sidebands for the COM mode (a) and the STR mode (b). The curves for the STR mode do not cross within the range of Fock states relevant to us. This means that the First sideband (blue curve) always has the fastest Rabi rate and therefore the shortest π -pulses. The COM side (a) is more complicated. Which sideband to use for the fastest π -pulse for a given state has to be calculated or looked up in a table like Table 3.1



Sequence 3.1: Cooling sequences with pulse times from Table 3.1. The first sequence uses only the 1st sideband. The second sequence uses the shortest available pulse from a set of sidebands. In this case the 1st and the 4th. The time axis uses arbitrary units.

$\bar{\Omega}_{n_k, n'_k}$ for a $|g\rangle |n\rangle_k \leftrightarrow |e\rangle |n'\rangle_k$ transition using the motional mode k assuming all other modes are in a thermal state with expected vibrational number \bar{n} . Adapting equations (124) from [WMI⁺98] to our needs we get

$$\begin{aligned}\bar{\Omega}_{n_k, n'_k} &= \Omega_{n_k, n'_k} \prod_{p \neq k} e^{-(\eta_p)^2 (\bar{n}_p + 1/2)} \\ \bar{\Omega}_{n_k, n'_k}^2 &= \Omega_{n_k, n'_k}^2 \prod_{p \neq k} e^{-2(\eta_p)^2 (\bar{n}_p + 1/2)} I_0 \left(2(\eta_p)^2 \sqrt{\bar{n}_p (\bar{n}_p + 1)} \right)\end{aligned}\tag{3.3}$$

The Raman beams used for sideband cooling have an effective wavevector along the axial direction. Therefore the Lamb-Dicke parameters along the radial directions will be approximately zero and don't have to be considered in the product in Equation 3.3. Using the known values $\eta_{COM} = 0.567$ and $\eta_{STR} = 0.098$ and the ex-

pected temperature after Doppler cooling $\bar{n}_{COM,Doppler} \approx 10$ and $\bar{n}_{STR,Doppler} \approx 2.4$ [SHS⁺23], we can approximate the effect of one axial mode being in a doppler cooled thermal state while trying to cool the other axial mode. The mean Rabi rate divided by the Rabi rate, without the Debye-Waller effect calculated in $\Omega_{(0)}$, is given by (C = COM, S = STR):

$$\begin{aligned}\bar{\Omega}_{n_C, n'_C} / \Omega_{(0), n_C, n'_C} &= e^{-(\eta_S)^2(\bar{n}_S+1/2)} = 0.973 \\ \bar{\Omega}_{n_S, n'_S} / \Omega_{(0), n_S, n'_S} &= e^{-(\eta_C)^2(\bar{n}_C+1/2)} = 0.034\end{aligned}\tag{3.4}$$

The mean squared Rabi rate is:

$$\begin{aligned}\bar{\Omega}_{n_C, n'_C}^2 / \Omega_{(0), n_C, n'_C}^2 &= e^{-2(\eta_S)^2(\bar{n}_S+1/2)} I_0 \left(2(\eta_S)^2 \sqrt{\bar{n}_S(\bar{n}_S+1)} \right) = 0.947 \\ \bar{\Omega}_{n_S, n'_S}^2 / \Omega_{(0), n_S, n'_S}^2 &= e^{-2(\eta_C)^2(\bar{n}_C+1/2)} I_0 \left(2(\eta_C)^2 \sqrt{\bar{n}_C(\bar{n}_C+1)} \right) = 0.156\end{aligned}\tag{3.5}$$

and the standard deviation $\Delta = \sqrt{\bar{\Omega}^2 - \bar{\Omega}^2}$ divided by the mean is:

$$\begin{aligned}\Delta_C / \bar{\Omega}_{n_C, n'_C} &= 0.027 \\ \Delta_S / \bar{\Omega}_{n_S, n'_S} &= 11.49\end{aligned}\tag{3.6}$$

The result is quite significant. While the Rabi frequency for COM pulses almost doesn't change, the STR pulses become very slow on average due to the low mean Rabi rate (Equation 3.4). Even worse for the STR cooling is the fact that the Rabi rate has a standard deviation roughly ≈ 11 times larger than its mean (Equation 3.6). This means the Rabi rate fluctuates wildly from shot to shot. For the COM mode, the standard deviation is negligible.

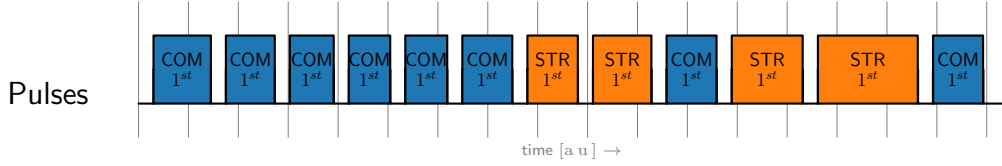
The same calculation can be done again for the STR mode but this time with COM at $\bar{n}_C = 0.07$, which is roughly the expected state after ground state cooling [SHS⁺23].

$$\begin{aligned}\text{for } \bar{n}_C &= 0.07 \\ \bar{\Omega}_{n_S, n'_S} / \Omega_{(0), n_S, n'_S} &= 0.833 \\ \bar{\Omega}_{n_S, n'_S}^2 / \Omega_{(0), n_S, n'_S}^2 &= 0.699 \\ \Delta_S / \bar{\Omega}_{n_S, n'_S} &= 0.088\end{aligned}\tag{3.7}$$

Which is much better compared to the case where the COM mode is at the Doppler temperature. This is why we always cool COM first and only then STR.

3.1.3 Heating

In our current setup, the motional modes heat up by ≈ 14 quanta per second [SHS⁺23]. For QLS, we usually require COM and STR mode to be in the ground state at the same time. And since we need to cool the COM mode first (due to dependence, see 3.1.2), the COM mode will heat up a small amount before the STR mode is also cooled. To get around this issue, we could add a few COM cooling pulses at the end of the sequence. And in fact, this is what was done before on this setup [SHS⁺23]. Another option - the one implemented here - is to interleave cooling pulses for different modes. The idea would be to insert COM pulses into the STR cooling sequence to keep the COM mode cool while we are cooling the STR mode. This would not allow the \bar{n}_{COM} to increase significantly because the last COM pulse would never be far back in time. An example can be seen in Sequence 3.2.



Sequence 3.2: Cooling COM, then STR with interleaved COM pulses. Time in arbitrary units.

3.2 Implementation

We want to be able to customize the cooling process for each separate experiment. Therefore the sideband cooling process needs to be parameterized. The chosen parameters are listed in table 3.2. The cooling function is implemented using a

| Parameter | Meaning |
|--------------------|--|
| mode | COM or STR |
| keep_cool_mode | [optional] only used to keep COM cool when mode is STR |
| keep_cool_every | [optional] Insert keep cool pulse after this many pulses |
| n_start_{com, str} | Highest numbered fock state we want to collect |
| n_repetitions | Number of times a single pulse is repeated |
| n_cycles | Number of times the whole cooling sequence is repeated |
| allowed_sidebands | The set of calibrated sidebands the algorithm can use |

Table 3.2: Sideband Cooling Parameters

control-system loop and looped-settings. See section 2.1.1 for more details. Every pulse in the sequence corresponds to one loop iteration. To make it possible for the loop to change the pulse frequency, amplitude and time, all of these parameters need to be looped-settings. The fact that we use Raman beams for sideband cooling further complicates this loop. The Raman beam requires two pulses to be run in parallel, with a frequency offset corresponding to the mode-sideband combo we want to drive. This leads to a total of 4 looped settings required to run an arbitrary sideband cooling sequence. The Beams used in our experiment are named R90 and RCO1. The looped settings for these two beams are shown in table 3.3. Each

| Beam | Type | Comment |
|----------|-----------|------------------------|
| R90/RCO1 | Time | Shared between the two |
| RCO1 | Frequency | |
| RCO1 | Amplitude | |
| R90 | Amplitude | |

Table 3.3: Looped-settings used for sideband cooling.

looped-settings requires a vector of values to be prepared. Setting up these vectors can be done by iterating from $n_start_{\{com, str\}}$ to 1 and finding the shortest pulse from the *allowed_sidebands* for each iteration. The frequency and amplitude can be looked up for the sideband with the lowest time required for a pulse. Then, all the values are pushed back onto the vectors at the end of the iteration.

3.3 Simulation

Simulations for both COM and STR cooling were performed to ensure that the sequence is working as expected and, in the case of STR cooling, to see whether interleaved COM cooling makes sense or not. The simulations were done using the parameters that were most commonly used in the lab. The details about the simulation parameters can be found in the figure's captions.

All simulations contain an artificial atom with states $|g\rangle$ and $|e\rangle$ with an arbitrary frequency chosen to be 10 MHz. This frequency is chosen because it is similar enough to the motional frequencies of $f_{COM} = 1.4$ MHz and $f_{STR} = 3.4$ MHz to not unnecessarily complicate the numerical simulations but also far enough from resonance. The motion is simulated using a truncated Fock state basis with the truncation limit $DIM_{COM,STR}$ being one of the simulation parameters. The couplings between different states and sidebands are calculated using Equation 1.26 from section 1.4. The results can be seen in Figure 3.3.

All simulations were performed using the Quantum toolbox in Python - QuTiP package for Python [JNN12][JNN13][VRD09]. Numerical simulations were performed using the *mesolve* functions from QuTiP. *mesolve* is a master equation solver that can simulate time-dependent terms of the Hamiltonian like a drive coming from an electric field. Time-dependent collapse operators are also supported. The collapse operators were used to achieve the same effect as repumping on the atom part of the state.

3.3.1 COM Cooling

For the COM cooling simulation, the STR mode was ignored completely to reduce computation time and because COM cooling does not depend on STR that much. See subsection 3.1.2 for details. Heating of the COM mode at a rate of $\dot{n} = 14$ quanta/s was also included. The heating rate was published in [SHS⁺23].

3.3.2 STR Cooling

The main goal of this simulation is to find out, how much the COM mode heats up during a normal STR cooling sequence. The heating rate for the COM mode is set again at $\dot{n}_{COM} = 14$ quanta/s. The starting $\bar{n}_{STR,thermal} = 2.4$ is taken again from [SHS⁺23]. Results in Figure 3.4. The final population of the ground state $|g\rangle$ after 800 μ s was 98.9%.

3.3.3 STR Cooling with Interleaved COM Pulses

This is the same simulation as in subsection 3.3.2 but this time with added interleaved COM pulses. The idea behind this pulse sequence is to keep the COM mode cool during STR cooling. Note: Debye-Waller factors (as discussed in subsection 3.1.2) are not included in the calculation in this version of the simulation. This would be a good next step to make it come closer to reality. Results are shown in Figure 3.5. The final population of the ground state $|g\rangle$ after 920 μ s was 99.97%.

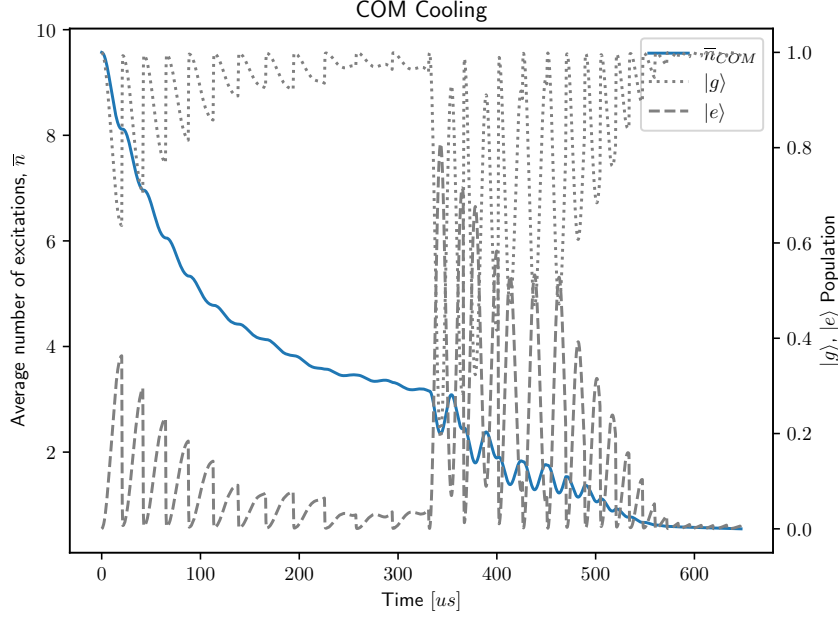


Figure 3.3: COM sideband cooling simulation. The change between using the 4th and 1st sideband is clearly visible. After every cooling pulse, the qubit was forced back into the ground state using a very fast decay. This should simulate the repumping. The blue line not starting at 10 can be explained by the finite Fock basis used to make the simulation possible. Cooling parameters used: $n_start_com = 15$, $n_repetitions = 2$. Simulation parameters used: $DIM_{COM} = 50$, $\bar{n}_{COM,thermal} = 10$, heating rate $\dot{n} = 14$ quanta/s, $t_{\pi,COM,|1\rangle \rightarrow |0\rangle} = 12.8 \mu s$.

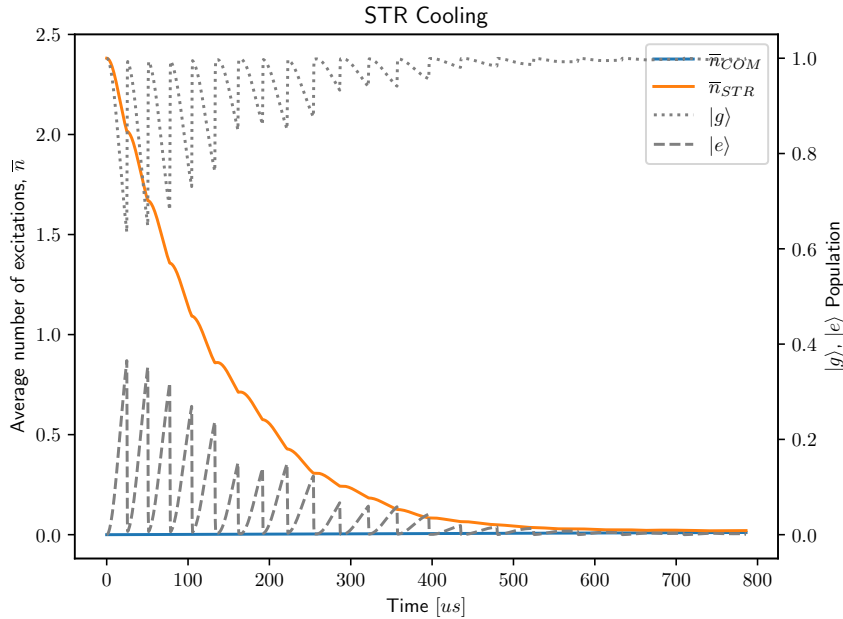


Figure 3.4: STR sideband cooling simulation without interleaved COM pulses. Cooling parameters used: $n_start_str = 10$, $n_repetitions = 2$. Simulation parameters used: $DIM_{COM} = 4$, $DIM_{STR} = 20$, $\bar{n}_{STR,thermal} = 2.4$, heating rate $\dot{n} = 14$ quanta/s, $t_{\pi,STR,|1\rangle \rightarrow |0\rangle} = 75 \mu s$, $t_{\pi,COM,|1\rangle \rightarrow |0\rangle} = 12.8 \mu s$.

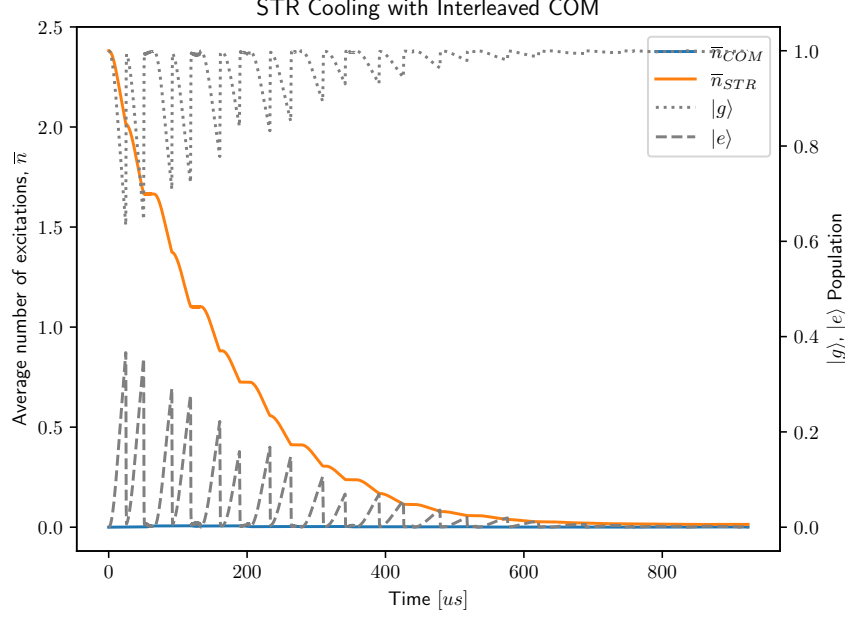


Figure 3.5: STR sideband cooling simulation with interleaved COM pulses. Cooling parameters used: $n_{start_str} = 10$, $n_{repetitions} = 2$, $keep_cool_every = 2$. Simulation parameters used: $DIM_{COM} = 4$, $DIM_{STR} = 20$, $\bar{n}_{STR,thermal} = 2.4$, heating rate $\dot{\bar{n}} = 14$ quanta/s, $t_{\pi,STR,|1\rangle \rightarrow |0\rangle} = 75 \mu\text{s}$, $t_{\pi,COM,|1\rangle \rightarrow |0\rangle} = 12.8 \mu\text{s}$.

3.4 Results

Data was taken to show the effect of sideband cooling on the presence of the red and blue sideband transitions after cooling. In a perfect motional ground state, we should only be able to drive the blue sideband. The red sideband should vanish due to that transition requiring one quanta of motion to be taken out of the system. But in the ground state there are no quanta present and therefore that transition is impossible to drive. To show that sideband cooling has any effect at all, also the states after doppler cooling are shown. There are many different motional states occupied at the doppler temperature and the Rabi rate for a sideband transition is different for each state as shown in section 1.4. Our experiment only drives the sidebands at the Rabi frequency $\Omega_{|1\rangle \leftrightarrow |0\rangle}$ between the one phonon and the zero phonon state. If the system is in any other state, the Rabi frequency will not match exactly and the executed pulse will be too long or too short. This will decrease the contrast in the plots.

All data presented here was taken on the same day within 20 minutes and with the same pair of ions.

3.4.1 COM Cooling

Figure 3.6 shows the first blue and red sidebands of the COM mode. The left image (a) shows the sidebands after doppler cooling, the right one (b) shown the same measurement after sideband cooling. Sideband cooling is clearly working as the red sideband almost vanishes behind the noise floor. This means that we were not able to drive a transition from $|g\rangle |n\rangle$ to $|e\rangle |n-1\rangle$, implying that $n = 0$ since there are no Fock states with negative numbers.

The Doppler cooled image also shows a difference between the red and the blue

sideband. This difference can be well explained by noticing that the blue sideband can be driven from all Fock states $| \geq 0 \rangle$ while the red sideband can only be driven from Fock states with at least one quantum of motion $| \geq 1 \rangle$. This makes the red sideband less likely to be driven depending on the temperature.

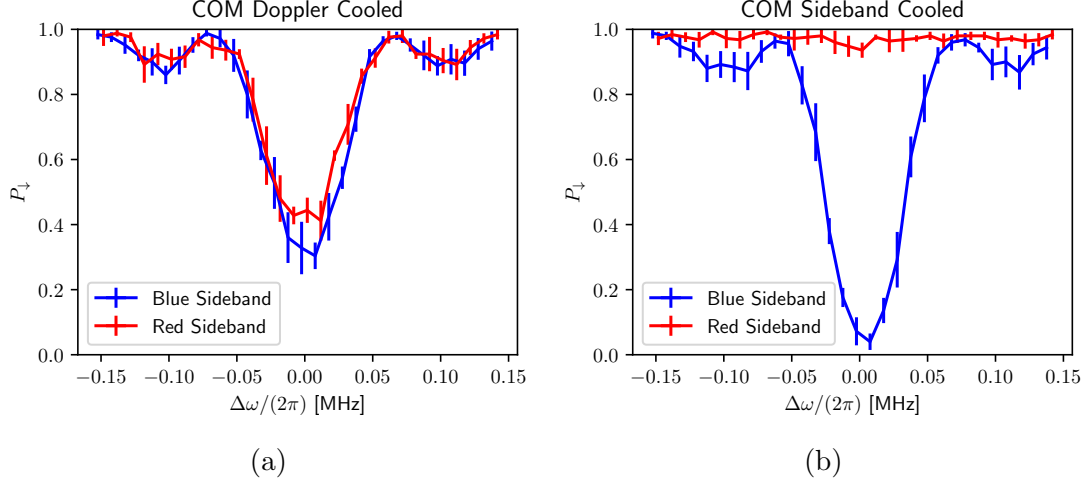


Figure 3.6: COM Cooling. (a) State of after Doppler cooling. (b) State after sideband cooling on the COM mode using the implementation described in section 3.2. The difference between the red and blue curves in (a) can be explained by the already low temperature, making the motional ground state more likely to be occupied - which causes the red sideband to disappear in that case.

3.4.2 STR Cooling

For the STR part, the most interesting aspect is the difference between the sideband cooling without (Figure 3.7b) and with interleaved COM cooling pulses (Figure 3.7c). While neither of them shows any activity on the red sideband, the blue sidebands have different contrast. This can be explained by the Debye-Waller factor from the COM mode influencing the Rabi rates if the COM mode is not kept cool at all times. The Debye-Waller factors cause different Rabi rates depending on the COM's state and this causes the calibrated time for a π -pulse to either be too long or too short. The end effect being an imperfect π -flop on the sideband.

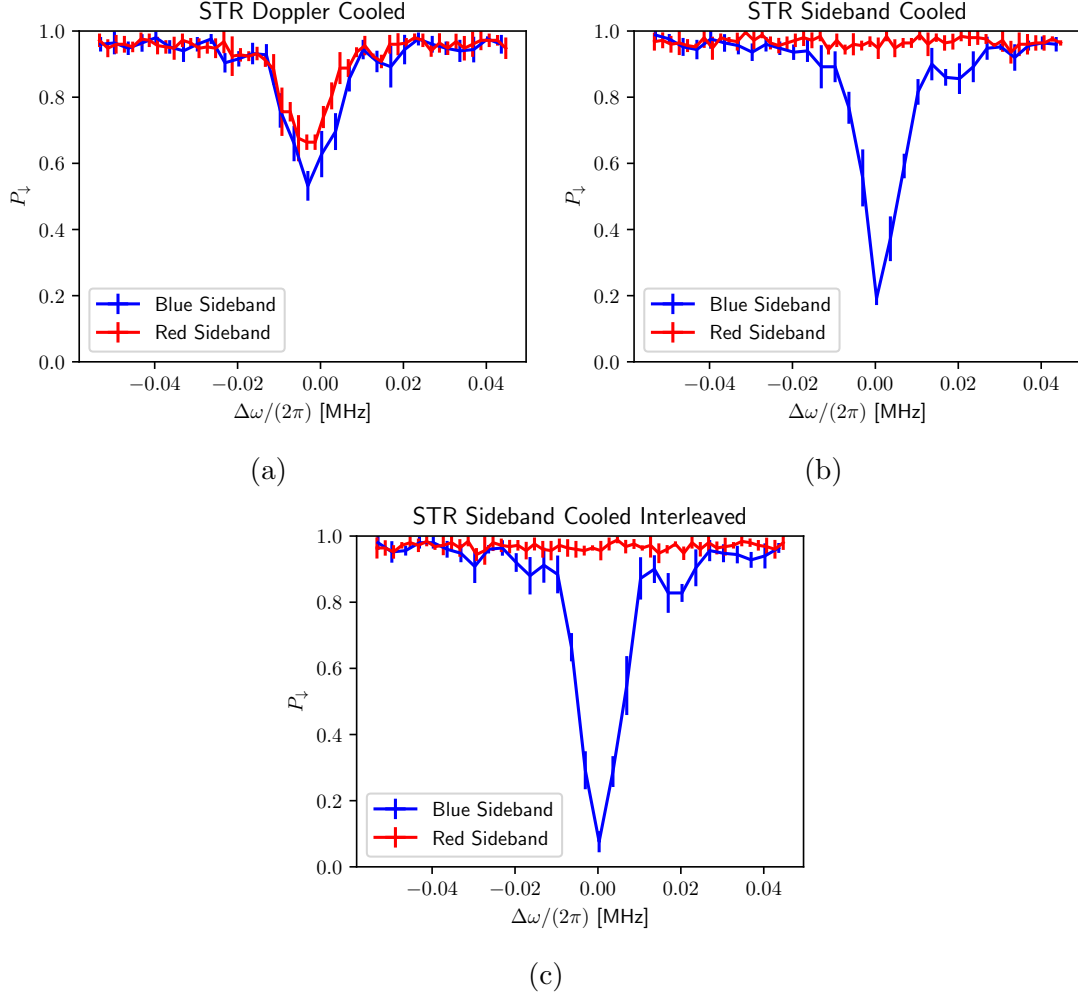


Figure 3.7: STR Cooling. (a) State after Doppler cooling. (b) State after COM, then STR cooling without added COM pulses during the STR sequence. (c) COM, then STR cooling with interleaved COM pulses during the STR cooling sequence. Keeping the COM mode cool ensures that only one state, the ground state, is occupied in the COM mode. This keeps the Rabi rate for the STR cooling constant because there is only one Debye-Waller factor influencing the Rabi rate, which goes into the calibrated time used for the sideband π -pulses.

Chapter 4

Radial Exchange Cooling

4.1 Overview

The last chapter covered the cooling of the two axial modes of the $Be^+ - H_2^+$ ion crystal. In this chapter, we will discuss the cooling of the four radial modes. The four radial modes have the interesting property of having very binary participation ratios. The participation ratio is a measure used to quantify how much one ion moves compared to the other when oscillating in a given mode. Sideband cooling using one of the ion's internal states, can only cool modes in which this ion participates. This can be seen in Equation 1.22 in section 1.4. The motional sidebands only appear if the ion, that houses the qubit transition, is not stationary. If we can consider it stationary, then the dipole approximation becomes valid and the equation reduces to Equation 1.1 from section 1.3, where no sidebands show up in the equations anymore. And without sidebands, there is also no sideband cooling possible. In our $Be^+ - H_2^+$ ion crystal, Be^+ is used as the ion for sideband cooling. Only modes with some participation of the Be^+ 's motion can be cooled this way. We call the two modes where almost only H_2^+ is moving the H_2^+ radials. They can not be cooled via Be^+ in a straightforward way. But there is a way to get around this. An interaction between the H_2^+ radials and the axial modes can be engineered using optical dipole forces. We will not go into how this is specifically done in the lab but the resulting interaction is mathematically equivalent to a beamsplitter interaction and can be described by the following equation:

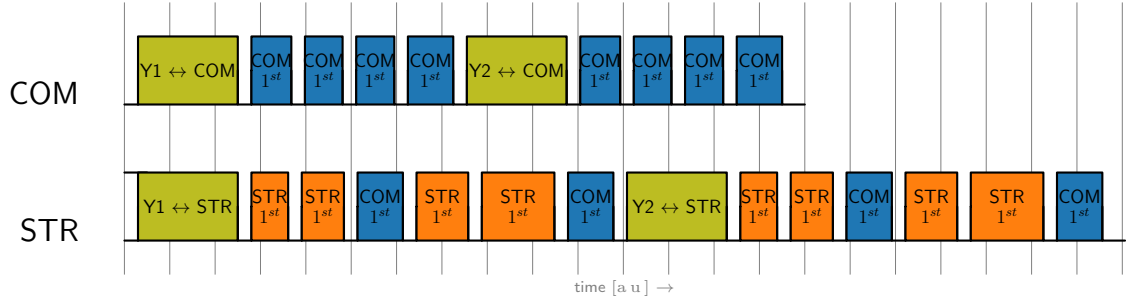
$$|p\rangle_{\text{radial}} \otimes |q\rangle_{\text{axial}} \rightarrow |q\rangle_{\text{radial}} \otimes |p\rangle_{\text{axial}} \quad (4.1)$$

After cooling at least one of the axial modes into the ground state, we can use the beamsplitter interaction to swap that ground state with the thermal population in one of the radial modes. After the interaction, the axial mode should now be in a thermal state and the radial mode should be in the ground state. After that, we can cool the axial mode again and reach our desired state of having radial and axial modes cold. We need to repeat this process for every radial we want to be cold.

For QLS to work in the experiment, we require the H_2^+ radials to be reasonably cold as they also introduce a Debeye-Waller effect to the sideband operations on H_2^+ . This is due to the fact that the Raman beam pair operating on H_2^+ has a projection onto all modes.

Two example pulse sequences involving exchanges between the axial and radial modes followed by cooling are shown in Sequence 4.1. The H_2^+ radial modes are labeled Y1 and Y2. The first sequence uses the axial COM mode as an exchange destination mode. After the exchange, that mode then has to be cooled as can be

seen by the sequence of COM pulses following each exchange. The second sequence uses the axial STR mode as the exchange destination mode. Cooling the axial STR mode has some intricacies discussed in the previous chapter. Axial COM pulses have to be added to avoid the Debye-Waller effect between the two axial modes.



Sequence 4.1: Exchange pulses for the H_2^+ radial modes Y1, Y2 followed by a sideband cooling sequence on COM or STR. Time in arbitrary units.

4.2 Implementation

We want to be able to customize the cooling process for each separate experiment. Therefore the sideband cooling process needs to be parameterized. The chosen parameters are listed in table 3.2.

Like the sideband cooling process, exchange cooling was also made to be customizable. Here mainly the radial modes that we want to cool are important. The parameters are listed in table 4.1. The exchange cooling function is implemented

| Parameter | Meaning |
|------------------|--|
| exchange_modes | Array containing a subset of radial modes $\{Y1, Y2, Z1, Z2\}$ |
| destination_mode | COM or STR - Mode to exchange with |
| radial_n_start | Highest numbered Fock state we want to cool after exchange |
| exchange_cycles | Number of times the whole exchange sequence is repeated |

Table 4.1: Exchange Cooling Parameters

using one nested loop with two inner loops. See section 2.1.1 for more details about nested loops. The first inner loop runs only one pulse, the exchange pulse. This single pulse has to be wrapped in a loop to fit into the framework for nested loops. The second inner-loop is the one responsible for sideband cooling and it is implemented in a very similar way. The sideband cooling part also inherits most of the parameters from normal sideband cooling. The only parameters that the exchange cooling changes compared to sideband cooling are the *radial_n_start* instead of *n_start* and *exchange_cycles* instead of *n_cycles*. The reason is that the radial modes might have a very different thermal \bar{n} after Doppler cooling. One should be able to set different starting Fock states for COM, STR and radial modes. The number of cycles is also specific to exchange cooling because it might need a lot of repetitions compared to normal sideband cooling, depending on the quality of the exchange interaction. Experimental results have shown that the exchange operation is not working as well as expected in our experiment. That is why this feature was requested.

The exchange operation uses a different pair of Raman beams than the sideband cooling. This beam pair must also be able to change between 4 different radial modes. Therefore its frequency needs to be varied to facilitate exchange on different modes. Also, the amplitude of the beams might have to be calibrated. This results in another 4 looped-settings having to be prepared in addition to the ones used for sideband cooling. The beams used for the exchange are identified as RA1 and RA2. The looped-settings required for them are shown in table 4.2

| Beam | Type | Comment |
|---------|-----------|------------------------|
| RA1/RA2 | Time | Shared between the two |
| RA1 | Frequency | |
| RA1 | Amplitude | |
| RA2 | Amplitude | |

Table 4.2: Looped-settings used for exchange pulses. These are in addition to the ones in table 3.3.

Chapter 5

H₂⁺ Pumping

5.1 Motivation

The current main goal of the experiment is to do quantum logic spectroscopy (QLS) as described in section 1.7. In the theory chapter we looked at a pair of states within the spectroscopy ion and labeled the states as $|g\rangle$ and $|e\rangle$. The real world is a bit more challenging than that. Given that we are in the rovibrational ground state $\nu = 0, L = 1$ of ortho H_2^+ , there are a total of 18 states within the hyperfine structure that we are interested in. The hyperfine structure of ortho H_2^+ in the rovibrational ground state is shown in Figure 5.1. After reaching the rovibrational

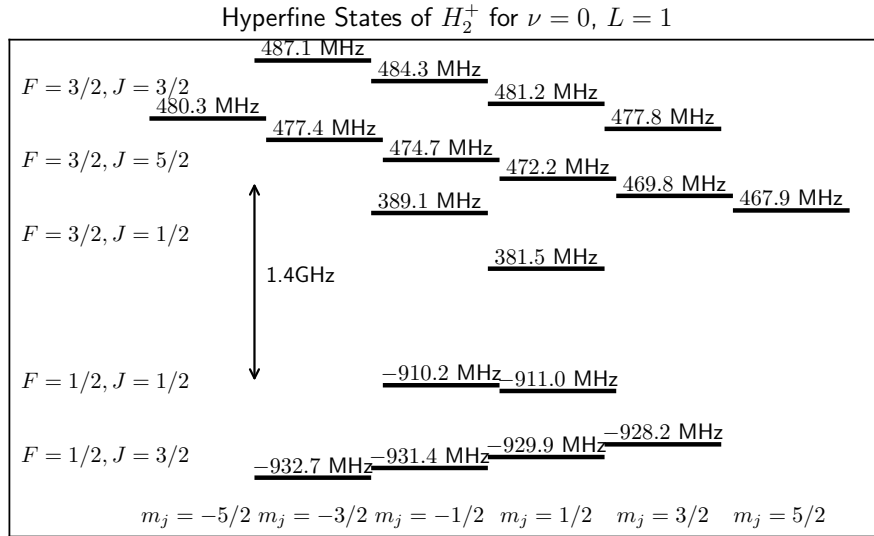


Figure 5.1: Hyperfine states and energies of ortho H_2^+ in the rovibrational ground state. Spacing between F, J manifolds adjusted for better visuals.

ground state due to buffer gas cooling, the system could be in any single one of these 18 states or a superposition or a statistical mixture. Currently, we do not possess any information that would tell us something about which state we are more likely to see than others. This is an issue for QLS because it can only read information about a pair of states. The QLS procedure in section 1.7 would give wrong results if there is any population in a state outside of the $\{|g\rangle, |e\rangle\}$ basis. The common way to get the population back into a desired subspace is to use optical pumping as explained in [Hap72] and [WMI⁺98]. Optical pumping cannot be performed directly on H_2^+ , because there is no usable optical transition. This is also the reason why quantum

logic spectroscopy is required. Pumping of CaH^+ using QLS has been proven to work [wCKH⁺17]. As already mentioned in section 1.7, QLS can not only be used for readout but for state transport in general. Any pumping sequence needs some way to eject entropy out of the system. In optical pumping, population is usually pumped into an unstable state that decays quickly into a lower energy state while scattering one or more photons. The photons can take entropy out of the system by scattering in a random direction. If the system does not remember where these photons went exactly, the process cannot be reversed and the entropy within the system is reduced. This can also be thought of as tracing out all the information contained in the emitted light. Every pumping sequence needs some way of ejecting entropy. But where the entropy is lost does not matter. By cleverly making use of the QLS procedure, we can transport the entropy contained in a pair of states within H_2^+ over to the Be^+ ion, where an optical transition is present to eject the entropy out of the system. This process is explained in more detail in the next section.

5.2 Overview

There are three steps required to combine the population of a pair of states in H_2^+ into one state. We will now go through these steps one after another, providing more details to understand each step of the process.

Both the Be^+ qubit and one of the motional modes must be prepared in the ground state. This is a prerequisite for the process to work. H_2^+ can start in any state. The goal is to combine the two states labeled $|g\rangle$ and $|e\rangle$ on the H_2^+ side. The three steps and the final state are shown in Figure 5.2, 5.3, 5.4 and 5.5. We will go through them in order.

• Step 1

A pair of Raman beams aimed at H_2^+ is used to perform a red sideband (RSB) π -pulse. This operation maps the state present in H_2^+ onto the motional qubit. Note that the no-cloning theorem is not violated because the H_2^+ state ends up in a pure ground state.

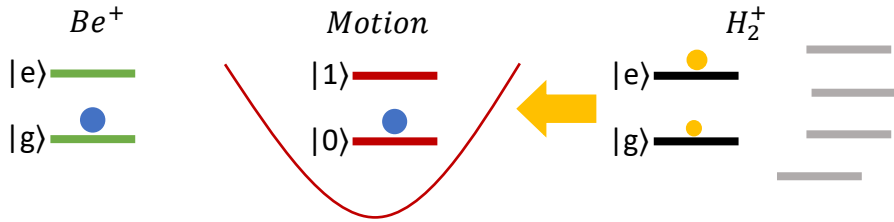


Figure 5.2: Step 1: A red sideband (RSB) pulse on H_2^+ is used to transport the H_2^+ qubit state into the motional qubit.

- **Step 2**

Another pair of Raman beams aimed at the Be^+ ion is used to perform a red sideband (RSB) π -pulse between Be^+ and the motional qubit. This is the same Raman pair used for sideband cooling. The operation transports the quantum state out of the motion and into the Be^+ 's internal qubit.

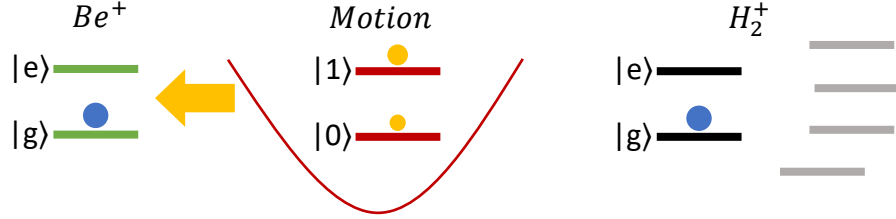


Figure 5.3: Step 2: A red sideband (RSB) on Be^+ transports the state from the motional qubit into the Be^+ internal qubit.

- **Step 3**

Up to this point, the entropy within the whole system did not change. All operations involved were unitary operations. The repumping sequence on Be^+ is now used to reset the Be^+ ion into the initial state that it had before step 1. This process scatters a photon and ejects the entropy out of the system.

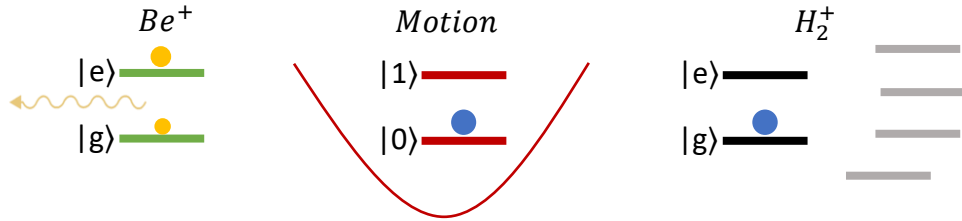


Figure 5.4: **Step 3:** Repumping of the Be^+ qubit causes a photon to scatter in a random direction. This causes the qubit to decay into the $|g\rangle$ state.

- **Final State**

The final state is reached once Be^+ is back in the ground state. Be^+ and the motion are now in the same state as they were at the beginning of the sequence. The only difference is present on the H_2^+ side. The two states were successfully combined into one. The procedure could now be repeated for another pair out of the 18 states within H_2^+ .

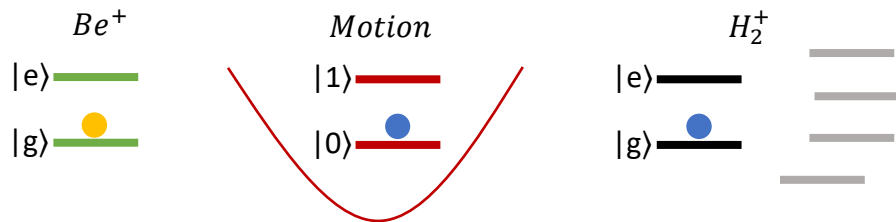


Figure 5.5: **Final State:** Both ions and the motion are in the ground state. The entropy on the H_2^+ side was successfully removed and the populations combined.

We managed to combine 2 out of 18 populations. However, the main goal is to combine the population of all 18 states. We can combine more of them by doing more pairwise combining operations until only one state contains all the population. Combining all 18 states required 17 rounds of these operations. This might sound simple, but in the lab, it's a bit more complicated. There are a few points that make the intuitive initial idea of just combining one state after the other unfeasible.

- **Selection Rules**

Not all neighboring states are connected by a strong transition due to selection rules of the Raman beams. More information about Raman transitions and selection rules can be found in [WMI⁺98].

- **Experimental Limitations**

The acousto optical modulators (AOMs), used to shift the frequency of one of the Raman beams, can only go up to a frequency in the 10s of MHz range. This causes some transitions to become inaccessible even if the transition is allowed. The set of all possible Raman transitions in the experiment is shown in Figure 5.6.

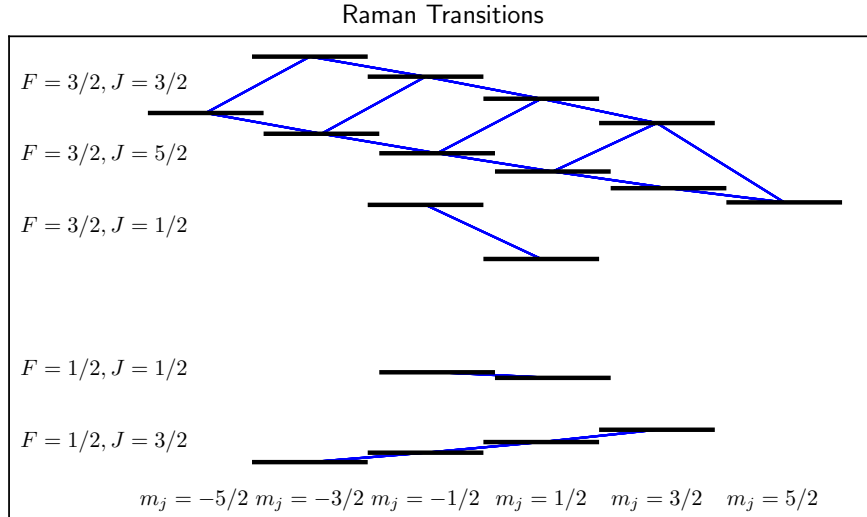


Figure 5.6: All Raman transitions between the hyperfine states of ortho H_2^+ in the rovibrational ground state that can be driven in the lab. There are two things required for a Raman transition to be considered possible. 1.) The transition needs to be allowed. 2.) It must be within the frequency range of the AOMs used for the Raman beams.

- **Calibration**

Every Raman beam needs to be calibrated. The Raman transition tends to be slow. Slower pulses are less broad in frequency space and therefore need to hit the transition frequency precisely. This does not apply to microwave pulses because they can be very powerful and fast. We are currently using the theoretical frequencies of the transition for them until there is some evidence that this is not good enough. The theoretical energies of the states expressed as frequencies can be found in Table 5.2

There are ways to get around these issues. One way is to make use of microwave pulses. Microwaves are roughly one or two orders of magnitude faster than Raman

pulses in our setup. But they do not couple to the motion due to the long wavelength of ≈ 20 cm at 1.5 GHz. They are useful nevertheless for bridging gaps in the hyperfine structure, especially between specific F, J manifold, that cannot be bridged with Raman transitions. The possible microwave transitions are shown in Figure 5.7. We now have a set of possible Raman and microwave transitions. The

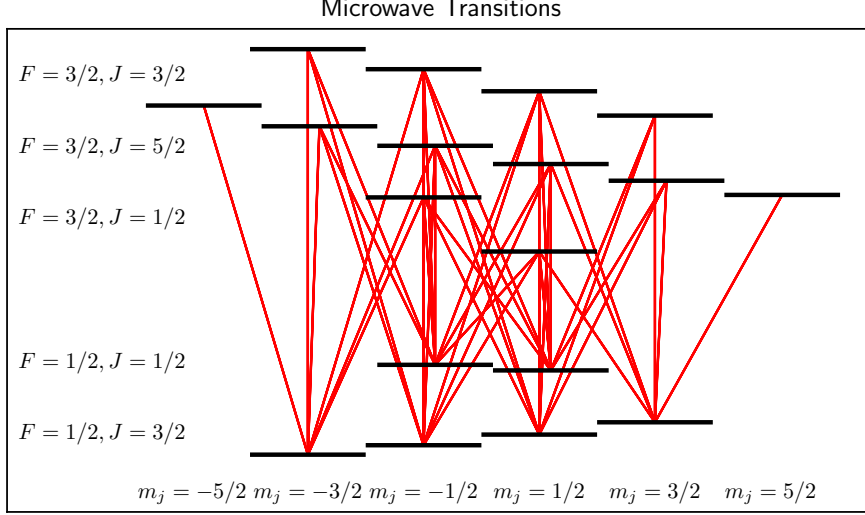


Figure 5.7: All microwave transitions between the hyperfine states of ortho H_2^+ in the rovibrational ground state that can be driven in the lab. The gap between the $F = 1/2$ and $F = 3/2$ is around 1 to 1.5 GHz - which lies perfectly within the range of the microwave. Other transitions are too low in frequency for the microwave to be used.

next goal is to find a sequence of pulses that combines all of the populations into one predetermined state without requiring an unreasonable amount of time to do so. This is a necessity to perform spectroscopy on H_2^+ . The algorithm found to solve this problem is discussed in the next section.

5.3 Algorithm

This section discusses the implementation of an algorithm that finds the population combining pulse sequence that requires the least amount of time to run. The assumptions made to simplify the problem are discussed first.

5.3.1 Assumptions

Our approach requires the following simplifying assumptions:

- **(1) Perfect operations**

Any operation performed by Raman beams or microwave pulses are optimal in the sense that these operations can be described by a perfect π -rotation on the Bloch sphere.

- **(2) Perfect motional ground state**

The motional mode used for sidebands is in a perfect ground state at the beginning of the QLS procedure. We further ignore heating of the motional modes.

- **(3) No unpredictable change of state**

The H_2^+ hyperfine state can change due to collision with atoms or molecules in the trap. We assume that no collision or any other disturbances are acting on the hyperfine state during the runtime of the pulse sequence.

Assumptions (1) and (2) allow for a great simplification of the problem. One state within the hyperfine structure will be completely cleared after the QLS procedure described in section 5.2 is run. Perfect microwave π -pulses only allow to swap two states. The amount of cleared states does not change in that case. The final goal of the algorithm is to clear all states except the one that is designated as the target state. This target state then must contain 100 % of the population. Note that we never care about the precise amount of population within a state. We only care about the state being populated at all or not. This allows to encode an arbitrary state of the hyperfine structure in a set of 18 bits. Where a logical 1 means that there might be population there and a logical 0 means that the state was cleared.

5.3.2 Implementation

Thinking of these binary vectors as nodes and the operations and their times as edges, one can interpret the problem as finding the shortest path in a graph. An example illustration can be found in Figure 5.8. One famous solution to this problem is Dijkstra’s algorithm [Dij22]. Dijkstra’s algorithm can be improved by adding a heuristic that gives the algorithm an idea of the direction it should explore first. One of these improved versions is called A* (A Star) and was first published by [HNR68] and is what was implemented here. How these algorithms work can be read in their papers and they won’t be further discussed here except when looking at some implementation details.

There were some difficulties encountered during the implementation. These and their solutions are listed here:

- **Memory Consumption**

There are a total of $2^{18} = 262'144$ nodes in the graph. It is common to store a graph in a connectivity matrix where the matrix element $[i, j]$ contains the weight of the edge from node i to j . Doing this using a dense matrix would require memory for 2^{36} matrix elements. Or roughly 68 GB if the entries are only one byte large. However, these matrices are usually sparse in nature, and using a sparse matrix format can get rid of most of the memory consumption.

- **Data Structure Setup**

Setting up this matrix requires iterating over all states and transitions. This process is just as computationally expensive as the pathfinding itself. There would be a benefit if the matrix could be reused for many runs. A map application for example doesn’t have to rebuild the graph of all streets when someone wants to look up the shortest path between two cities. The connectivity matrix can be reused for many queries in that case. In our case, we can just calculate the edges for a given node on demand. That means, we let the A* algorithm run and whenever it needs to know the possible connections from a node, we calculate them. Since A* visits every node at most once, we might not even have to build up the whole graph. Once A* knows that there cannot be a shorter path, it can stop looking at the remaining nodes and edges.

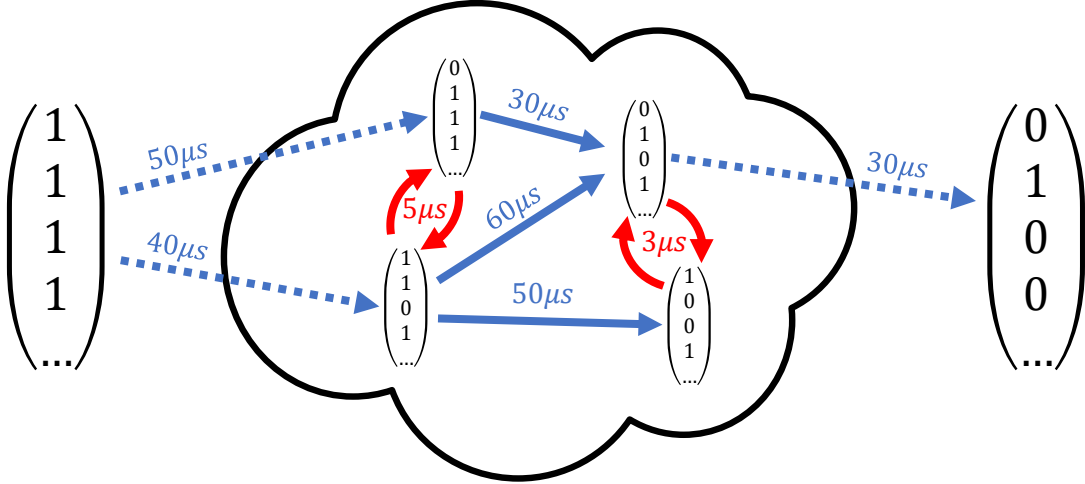


Figure 5.8: Example graph. Each binary vector stores one bit per state in the hyperfine structure. The bit being logical 1 indicates that the state might contain a population ≥ 0 . Logical 0 represents a cleared state. Red lines show microwave transitions that swap populations. A microwave connection is only present if one of the transitions shown in Figure 5.7 can be applied to that state and only if it swaps bits. Note that the connections in this example graph are not connected to the real transitions possible in our setup. The same principle applies to the Raman transitions. Only Raman transitions that are physically possible and are combining bits are part of the graph. Note that every Raman combining operation is actually a full QLS procedure running. But the only time-varying part of the QLS procedure is the H_2^+ Raman part. It is sufficient to do the time optimization over the time-varying part only. Credits to Martin Stadler for coming up with this idea during a group meeting.

A custom version of A* with on-demand calculation of node neighbors and a binary heap-based priority queue was implemented in Python [VRD09]. The priority queue contains the nodes which should be visited next. The queue was augmented with an additional hash table that maps nodes to their entry in the queue. This allows a node to be moved forward in the queue if it should be visited sooner according to the heuristic used. The use of a heuristic function is the most significant difference between the original Dijkstra's algorithm and A*. The heuristic function should return a lower bound for the distance left from the current node to the target node. The value can be used by the algorithm to get a sense of closeness to the target node. It will prefer to explore graph nodes that are closer to the target and will spend less time exploring in all directions. Going back to the map analogy, a good heuristic there would be the Euclidean distance between two points on the map. This is a good lower bound for the shortest road between these points. A* would then prefer to explore roads that tend to go towards the target city. Note that the usage of the heuristic function does not change the optimality of the solution as long as it is always a lower bound on the distance between two nodes. This was proven by the authors of A* in [HNR68]. The heuristic function used in our case was

$$h(\text{node}) = (\text{sum_bits}(\text{node}) - \text{sum_bits}(\text{target})) * t_{\min, \text{Raman}} \quad (5.1)$$

Credits to Alexander Ferk for coming up with this heuristic. This heuristic can be

used due to the fact that only Raman beams can remove set bits from a node's vector of bits. Going from a vector of 18 logical 1's to 1 logical 1 requires at least 17 Raman pulses. Using the minimum Raman pulse time in the heuristic assures that it is always a lower bound for the actual time needed to go from a given node to the target node.

Once A^* has finished running, it will return a data structure that contains the previous node on the shortest path to every visited node. Starting at the target state and going in reverse, the whole path can be reconstructed. The resulting sequence will be discussed in the next section.

5.4 Pumping Sequence

The output of the algorithm depends on the calculated transition matrix elements of both the microwave and Raman transitions, as well as the relative time between the two kinds of operations. The matrix elements depend on the polarization of the beams and are proportional to the Rabi frequency that one would achieve when driving that transition. An example output of the algorithm can be seen in Table 5.1. The mapping between numbers and states in ket-notation can be found in Table 5.2. The example sequence was calculated using an arbitrary polarization and with Raman beams being roughly one order of magnitude slower than the microwaves. We can see that the algorithm found one Raman transition that it likes to use. This is one of the Raman transition with the largest transition matrix element and is therefore considered one of the fastest by the algorithm. The microwaves, being roughly one order of magnitude faster, are mainly used to move population towards one of the two states connected by the fast Raman.

| Pulse # | From \rightarrow To | Type |
|---------|-------------------------|-----------|
| 0 | (13) \rightarrow (10) | QLS Proc. |
| 1 | (4) \rightarrow (13) | Microwave |
| 2 | (10) \rightarrow (13) | QLS Proc. |
| 3 | (2) \rightarrow (10) | Microwave |
| 4 | (10) \rightarrow (13) | QLS Proc. |
| 5 | (9) \rightarrow (2) | Microwave |
| 6 | (2) \rightarrow (10) | Microwave |
| 7 | (10) \rightarrow (13) | QLS Proc. |
| 8 | (1) \rightarrow (10) | Microwave |
| 9 | (13) \rightarrow (10) | QLS Proc. |
| 10 | (7) \rightarrow (4) | Microwave |
| 11 | (4) \rightarrow (13) | Microwave |
| 12 | (13) \rightarrow (10) | QLS Proc. |
| 13 | (15) \rightarrow (4) | Microwave |
| 14 | (4) \rightarrow (13) | Microwave |
| 15 | (10) \rightarrow (13) | QLS Proc. |
| 16 | (16) \rightarrow (2) | Microwave |
| 17 | (2) \rightarrow (10) | Microwave |
| 18 | (10) \rightarrow (13) | QLS Proc. |
| 19 | (12) \rightarrow (1) | Microwave |
| 20 | (1) \rightarrow (10) | Microwave |
| 21 | (10) \rightarrow (13) | QLS Proc. |
| 22 | (17) \rightarrow (1) | Microwave |
| 23 | (1) \rightarrow (10) | Microwave |
| 24 | (10) \rightarrow (13) | QLS Proc. |
| 25 | (0) \rightarrow (12) | Microwave |

| Pulse # | From \rightarrow To | Type |
|---------|-------------------------|-----------|
| 26 | (12) \rightarrow (1) | Microwave |
| 27 | (1) \rightarrow (10) | Microwave |
| 28 | (10) \rightarrow (13) | QLS Proc. |
| 29 | (14) \rightarrow (0) | Microwave |
| 30 | (0) \rightarrow (12) | Microwave |
| 31 | (12) \rightarrow (1) | Microwave |
| 32 | (1) \rightarrow (10) | Microwave |
| 33 | (10) \rightarrow (13) | QLS Proc. |
| 34 | (11) \rightarrow (1) | Microwave |
| 35 | (1) \rightarrow (10) | Microwave |
| 36 | (10) \rightarrow (13) | QLS Proc. |
| 37 | (5) \rightarrow (17) | Microwave |
| 38 | (17) \rightarrow (1) | Microwave |
| 39 | (1) \rightarrow (10) | Microwave |
| 40 | (10) \rightarrow (13) | QLS Proc. |
| 41 | (6) \rightarrow (5) | Microwave |
| 42 | (5) \rightarrow (17) | Microwave |
| 43 | (17) \rightarrow (1) | Microwave |
| 44 | (1) \rightarrow (10) | Microwave |
| 45 | (13) \rightarrow (10) | QLS Proc. |
| 46 | (3) \rightarrow (13) | Microwave |
| 47 | (13) \rightarrow (10) | QLS Proc. |
| 48 | (8) \rightarrow (3) | Microwave |
| 49 | (3) \rightarrow (13) | Microwave |
| 50 | (13) \rightarrow (10) | QLS Proc. |

Table 5.1: Example sequence for arbitrarily chosen polarizations and relative times between microwave and Raman beams. The target state was chosen to be state # (10). Red arrows indicate microwave transitions. Blue arrows stand for one iteration of the QLS process.

| State # | State | Energy [MHz] |
|---------|--|--------------|
| (0) | $ F = 1/2, J = 3/2, m_j = -3/2\rangle$ | -932.72 |
| (1) | $ F = 1/2, J = 3/2, m_j = -1/2\rangle$ | -931.4 |
| (2) | $ F = 1/2, J = 3/2, m_j = 1/2\rangle$ | -929.92 |
| (3) | $ F = 1/2, J = 3/2, m_j = 3/2\rangle$ | -928.2 |
| (4) | $ F = 1/2, J = 1/2, m_j = 1/2\rangle$ | -910.97 |
| (5) | $ F = 1/2, J = 1/2, m_j = -1/2\rangle$ | -910.19 |
| (6) | $ F = 3/2, J = 1/2, m_j = 1/2\rangle$ | 381.54 |
| (7) | $ F = 3/2, J = 1/2, m_j = -1/2\rangle$ | 389.1 |
| (8) | $ F = 3/2, J = 5/2, m_j = 5/2\rangle$ | 467.86 |
| (9) | $ F = 3/2, J = 5/2, m_j = 3/2\rangle$ | 469.84 |
| (10) | $ F = 3/2, J = 5/2, m_j = 1/2\rangle$ | 472.18 |
| (11) | $ F = 3/2, J = 5/2, m_j = -1/2\rangle$ | 474.73 |
| (12) | $ F = 3/2, J = 5/2, m_j = -3/2\rangle$ | 477.45 |
| (13) | $ F = 3/2, J = 3/2, m_j = 3/2\rangle$ | 477.8 |
| (14) | $ F = 3/2, J = 5/2, m_j = -5/2\rangle$ | 480.35 |
| (15) | $ F = 3/2, J = 3/2, m_j = 1/2\rangle$ | 481.2 |
| (16) | $ F = 3/2, J = 3/2, m_j = -1/2\rangle$ | 484.28 |
| (17) | $ F = 3/2, J = 3/2, m_j = -3/2\rangle$ | 487.08 |

Table 5.2: Hyperfine states of ortho H_2^+ in the rovibrational ground state, numbered and sorted by energy. The state numbers start at 0 because they are also used to index into the binary arrays during graph operations of our algorithm.

5.5 Simulation

From looking at table Table 5.1, it is not entirely obvious why this sequence should work. The effect of the sequence on a given state and its dependence on imperfections in the experiment was simulated, and the results are discussed here. There are a few things that can go wrong in the experiment. The operations performed on the ions are never perfect, especially when the transitions are only known from theory. This was calculated in by artificially worsening the transfer of populations. This sequence will refer to the quality of the operations either as operation fidelity or beam fidelity. Defining the fidelity of a quantum operation is not an easy task. The common way to define the fidelity of an operation is to calculate the fidelity of the ideal result and the average result of pure and uniformly chosen input state [PMM07]. These simulations use a simpler definition. Assuming a π -pulse operation σ_π has a fidelity of α , then applying that transition to a pure state $|g\rangle$ would result in

$$\sigma_\pi |g\rangle \langle g| \sigma_\pi \rightarrow \alpha |e\rangle \langle e| + (1 - \alpha) |g\rangle \langle g| \quad (5.2)$$

The equation can be understood as moving a factor α of the population from one state to the other and a factor of $(1 - \alpha)$ of the population stays behind. The simulation uses this definition of fidelity to describe the movement of population within the hyperfine states of the H_2^+ ion.

• Simulation 1

The first simulation was done using the pumping sequence from Table 5.1. The results are in Figure 5.9. Details about the simulation can be found in the caption. This first simulation shows, rather surprisingly, that more repetitions do not increase the final state fidelity. This seems strange, as pumping

sequences usually increase the fidelity with more repetitions. Something seems to be off. It turns out that the algorithm tends to sometimes pump population in and out of the target state. It does so, because it was built with the assumption that all operations are perfect. And if the operations are perfect, the simulation show that it works indeed. But in reality, shuffling population around with imperfect operations rather spreads than combines the population. The algorithm can save a bit of time by doing that. But the costs on the fidelity side are way to big if the operations aren't perfect. Something had to be changed.

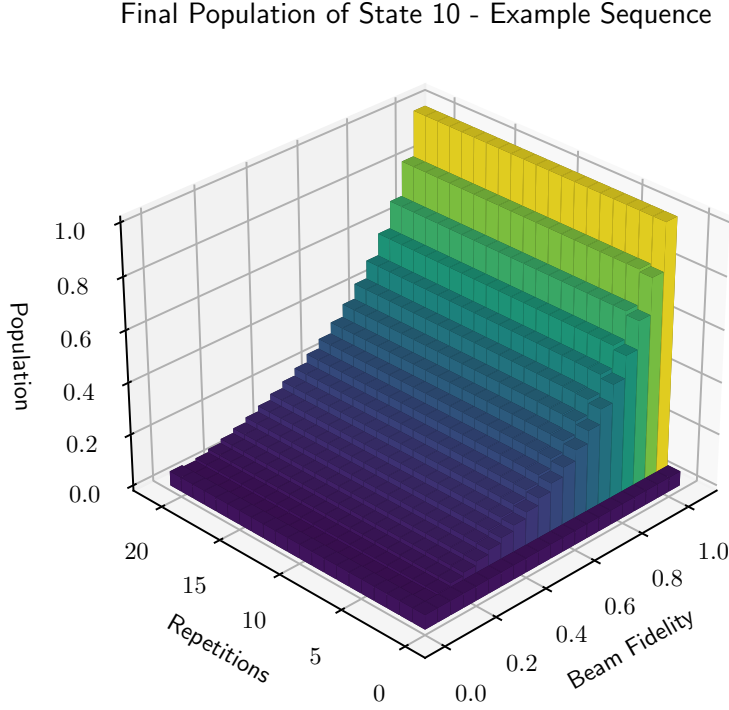


Figure 5.9: Final population of the target state (10) (z-axis) using the example pumping sequence from Table 5.1. The initial state was uniformly distributed. The sequence was repeated 0 to 20 times and the beam fidelities were varied from 0 % to 100 %. It can be seen that the system reaches a steady-state population $P_{10} < 1$ for imperfect beams. This is rather surprising as one would expect a pumping sequence to converge to the target state even for imperfect operations.

• Simulation 2

A second pulse sequence was generated, but this time with the additional rule that Raman beams where only allowed to pump from the higher energy state to the lower state. This prevents the algorithm from returning a solution that uses the same transition on both directions multiple times. This led to a sequence where the target state (10) was always the receiving state of a Raman transition. Pumping now clearly works as can be seen by the results in Figure 5.10. Details about the simulation can be found in the caption. With the pumping sequence working we can go on and simulate another imperfection in the experiment, the ground state fidelity of the motion.

Final Population of State 10 - One Way Pumping

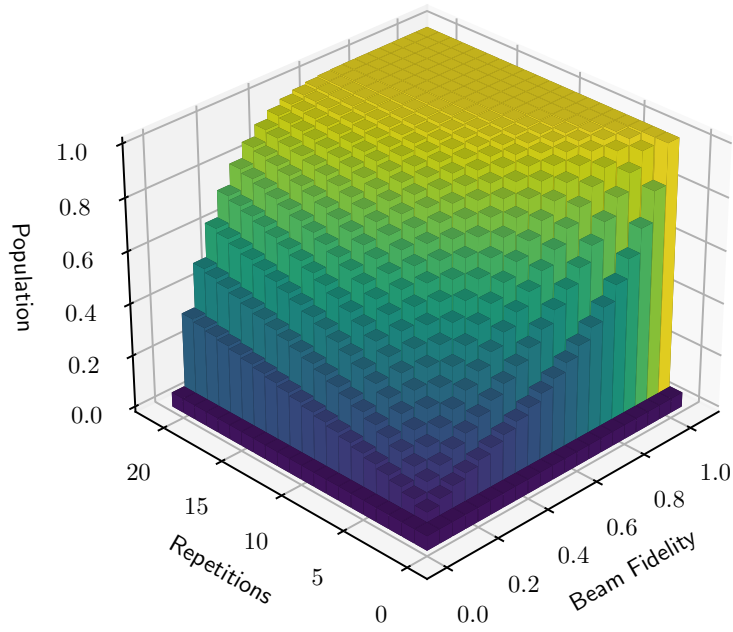


Figure 5.10: The same simulation as in Figure 5.9, but this time with an adjusted sequence. The one-way sequence only uses Raman transitions in one way. In fact it only uses $|13\rangle \rightarrow |10\rangle$ transition. The difference is quite significant. 100 % target state fidelity can now be reached even using bad operations. One-way pumping is clearly the way to go.

- **Simulation 3**

The operation fidelities are now fixed at 50 % and instead of operation fidelity, the motional mode is now simulated to be a thermal state with average excitation number \bar{n} . The thermal \bar{n} is now swept. The results are plotted in Figure 5.11. Details in the caption. It is clearly visible that the system reaches a steady state depending on the temperature of the motion. Any population in a motional state, except the ground state, occasionally causes the QLS procedure to fail, depending on which specific state the system was in during one shot of the measurement. In that case, the QLS procedure loses its capability to combine two states and instead become swapping operations like the microwave. These operations cannot reduce the entropy of the system and progress stagnates.

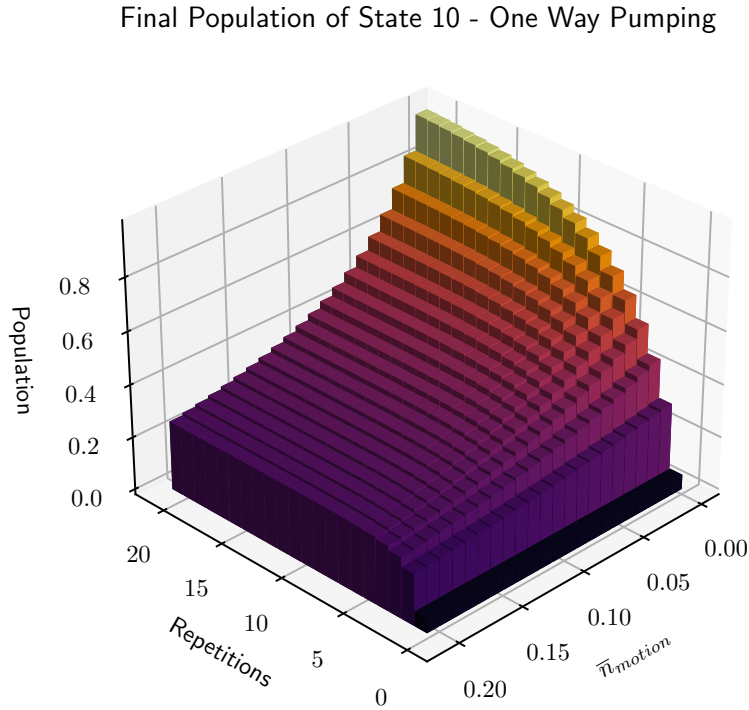


Figure 5.11: Similar simulation as Figure 5.10, but this time with a fixed operations fidelity of 0.5. The right axis is now varying a new parameter, the average number of excitation \bar{n} of thermal state of the motion. The QLS procedure requires one of the motional modes to be in the ground state. If another state than the ground state is present, step 1 in Figure 5.2 does not combine the population on the H_2^+ side but rather swap the populations the $|g\rangle$ and $|e\rangle$ states. This effect leads to a steady-state where $P_{10} < 1$ never reaches perfect fidelity.

Looking back at all three simulations, number 3 has probably shown the most important results. It shows how important proper motional ground state cooling is for QLS to work. Understanding the math behind the steady-state could also provide some insight into ways to improve the pumping. An attempt to explain the final state given a certain motional \bar{n} is provided here:

A steady state is reached if the average Raman pulse has no effect anymore. There is no effect if the population pumped into the target state equals the population coming back due to the imperfect motional ground state. This can be put into an equation:

$$S_t(1 - P_{gs}) = \bar{S}_{!t} \quad (5.3)$$

where S_t is the population of the target state, P_{gs} is the probability to be in the motional ground state and $\bar{S}_{!t}$ is the average population in all states other than the target state. We also know that the populations have to sum up to one, which for 18 states means:

$$S_t + 17 * \bar{S}_{!t} = 1 \quad (5.4)$$

Inserting Equation 5.3 into Equation 5.4 we get

$$S_t + 17 * S_t(1 - P_{gs}) = 1 \quad (5.5)$$

which can be rewritten to

$$S_t = \frac{1}{1 + 17 * (1 - P_{gs})} \quad (5.6)$$

The curve is plotted in Figure 5.13. The blue curve matches the simulation steady state from Figure 5.11 quite well. The steady-state equation Equation 5.3 hints at one of the problems this sequence has. It tries to combine small populations into one big one and gets a lot of population back in return due to the slightly imperfect ground state. We could try to reduce the population gap between these states by adding one intermediate state S_i in between. We will now pump into this intermediate state first and then into S_t . The equations are

$$\begin{aligned} S_t(1 - P_{gs}) &= S_i \\ S_i(1 - P_{gs}) &= \bar{S}_{!_{\{t,i\}}} \end{aligned} \quad (5.7)$$

resulting in

$$S_t = \frac{1}{1 + 1 * (1 - P_{gs}) + 16 * (1 - P_{gs})^2} \quad (5.8)$$

This curve is also plotted in Figure 5.11. It is much more promising than the first sequence and should give a very clear QLS signal, even for mediocre cooling.

Final Population of State 10 - One Way Pumping

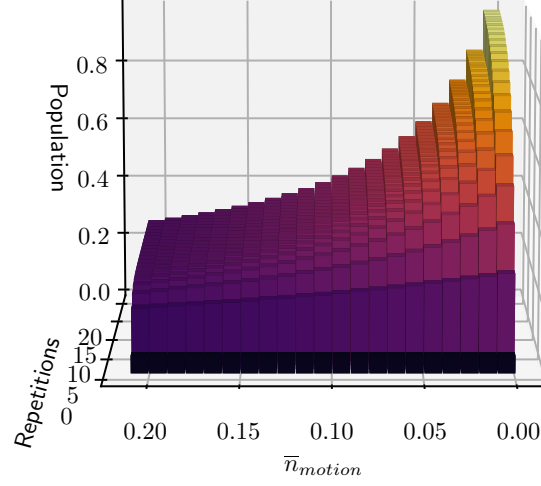


Figure 5.12: Side view of the simulation data in Figure 5.11.

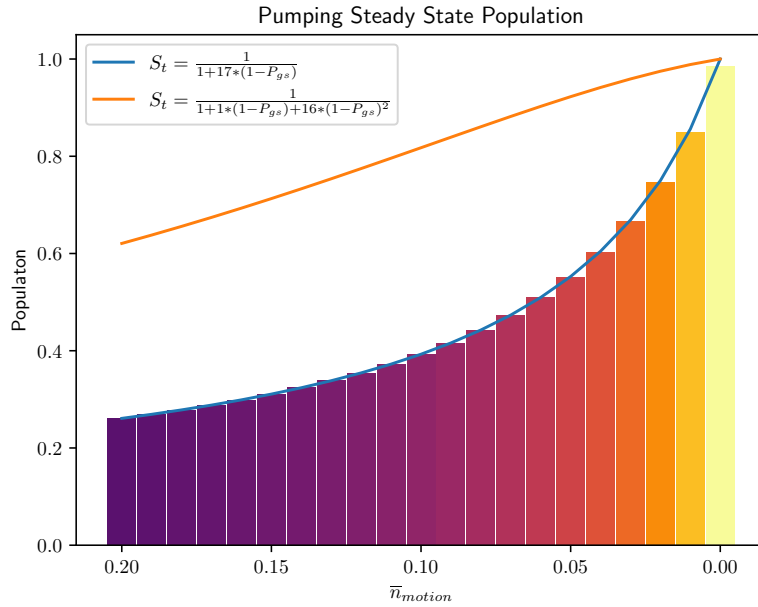


Figure 5.13: Final Population of State 10 in the steady state, calculated from the formulas shown in the figure (blue and orange lines). A slice of simulation data from Figure 5.11 at Repetitions = 20 is plotted (colored bars). The calculated curve matches the simulation data very well. Using an intermediate state (orange line) improves the final state fidelity a lot when compared to the sequence without the intermediate state, especially for imperfect ground states.

Chapter 6

Outlook

The new sideband cooling implementation is working well and exchange cooling will soon be tested further. Following the promising simulation results from section 5.5, we are now trying to confirm our findings in the lab. A working pumping sequence would allow for a much stronger QLS signal, which would allow us to start mapping out the hyperfine states of H_2^+ and confirm the results of numerical calculations. So far, we haven't seen any improvement in the signal coming from the pumping sequence. However, not all options discussed in this work have been tried in the lab so far. The one-way pumping sequence was implemented but the signal wasn't significantly better. The pumping via an intermediate state, as discussed in the simulation sections, is currently being tested. These are certainly exciting times for the molecules team and we are looking forward to new discoveries waiting just around the corner.

Bibliography

- [BO27] M. Born and R. Oppenheimer. Zur quantentheorie der molekeln. *Annalen der Physik*, 389(20):457–484, 1927.
- [Bur27] Øjvind Burrau. Berechnung des Energiewertes des Wasserstoffmolekels (H_{2}^{+}) im Normalzustand. *Naturwissenschaften*, 15(1):16–17, January 1927.
- [Dij22] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pages 287–290. Association for Computing Machinery, 2022.
- [Fit10] Richard Fitzpatrick. Quantum mechanics course. <https://farside.ph.utexas.edu/teaching/qmech/Quantum/node129.html>, July 2010. Accessed: 14/9/2023.
- [Fou23] Standard C++ Foundation. C++ logo. <https://isocpp.org/>, 2023. Accessed: 18/9/2023.
- [Hap72] William Happer. Optical pumping. *Rev. Mod. Phys.*, 44:169–249, Apr 1972.
- [HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [Hom13] Jonathon P. Home. Chapter 4 - quantum science and metrology with mixed-species ion chains. In Ennio Arimondo, Paul R. Berman, and Chun C. Lin, editors, *Advances in Atomic, Molecular, and Optical Physics*, volume 62 of *Advances In Atomic, Molecular, and Optical Physics*, pages 231–277. Academic Press, 2013.
- [JNN12] J.R. Johansson, P.D. Nation, and Franco Nori. QuTiP: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, aug 2012.
- [JNN13] J.R. Johansson, P.D. Nation, and Franco Nori. QuTiP 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, apr 2013.
- [Kar14] J.-Ph. Karr. H_2^+ and HD^+ : Candidates for a molecular clock. *Journal of Molecular Spectroscopy*, 300:37–43, jun 2014.
- [KB23] Vladimir I. Korobov and D. Bakalov. Forbidden ortho-para electric dipole transitions in the h_2^+ ion. *Phys. Rev. A*, 107:022812, Feb 2023.

- [KDH12] Jean-Philippe Karr, A. Douillet, and Laurent Hilico. Photodissociation of trapped H_2^+ ions for rempd spectroscopy. *Applied Physics B*, 107:1043, 05 2012.
- [KHKK16] J.-Ph. Karr, L. Hilico, J. C. J. Koelemeij, and V. I. Korobov. Hydrogen molecular ions for improved determination of fundamental constants. *Phys. Rev. A*, 94:050501, Nov 2016.
- [Kie15] Daniel Kienzler. *Quantum Harmonic Oscillator State Synthesis by Reservoir Engineering*. Doctoral thesis, ETH Zurich, Zürich, 2015.
- [Lou83] R. Loudon. *The Quantum Theory of Light*. Oxford science publications. Clarendon Press, 1983.
- [Neg27] Vlad Negnevitsky. *Feedback-stabilised quantum states in a mixed-species ion system*. Doctoral thesis, ETH Zurich, Zurich, 2018-09-27.
- [Pau90] Wolfgang Paul. Electromagnetic traps for charged and neutral particles. *Rev. Mod. Phys.*, 62:531–540, Jul 1990.
- [PMM07] Line Hjortshøj Pedersen, Niels Martin Møller, and Klaus Mølmer. Fidelity of quantum operations. *Physics Letters A*, 367(1):47–51, 2007.
- [SHS⁺23] N. Schwegler, D. Holzapfel, M. Stadler, A. Mitjans, I. Sergachev, J. P. Home, and D. Kienzler. Trapping and ground-state cooling of a single H_2^+ . *Phys. Rev. Lett.*, 131:133003, Sep 2023.
- [SK05] S. Schiller and V. Korobov. Tests of time independence of the electron and nuclear masses with ultracold molecules. *Phys. Rev. A*, 71:032505, Mar 2005.
- [SKHV⁺17] S. Schiller, I. Kortunov, M. Hernández Vera, F. Gianturco, and H. da Silva. Quantum state preparation of homonuclear molecular ions enabled via a cold buffer gas: An ab initio study for the H_2^+ and the D_2^+ case. *Phys. Rev. A*, 95:043411, Apr 2017.
- [SRL⁺05] Piet Schmidt, Till Rosenband, C. Langer, Wayne Itano, James Bergquist, and David Wineland. Spectroscopy using quantum logic. *Science*, 309(5735), 2005-07-29 2005.
- [SZ97] Marlan O. Scully and M. Suhail Zubairy. *Atom-field interaction – semiclassical theory*, page 145–192. Cambridge University Press, 1997.
- [VRD09] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [wCKH⁺17] Chin wen Chou, Christoph Kurz, David B. Hume, Philipp N. Plessow, David R. Leibbrandt, and Dietrich Leibfried. Preparation and coherent manipulation of pure quantum states of a single molecular ion. *Nature*, 545(7653):203–207, may 2017.
- [WMI⁺98] 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, 1998.