Programmable optical tweezer arrays

Christoph Biesek

Masterarbeit in Physik
angefertigt im Physikalischen Institut

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
Bonn

Februar 2024

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.

Bonn, ...
Date Signature

1. Gutachter:	Prof. Dr. Sebastian Hofferberth
2. Gutachter:	Prof. Dr. Simon Stellmer

Acknowledgements

I would like to thank Prof. Dr. Sebastian Hofferberth for offering me the opportunity to work in his group and for providing me such an interesting and exciting topic for my Master thesis.

I also thank Prof. Dr. Simon Stellmer for his interest in my work and for taking the role as my second supervisor.

I want to particularly thank Dr. Nina Stiesdal and Lukas Ahlheit for continuously supporting me in the laboratory. I am very grateful for learning from you and for sharing your lab experience with me. A special thanks also goes to Dr. Wolfgang Alt for his valuable advice.

Furthermore I would like to thank the whole NQO group for always being supportive during my time here and for having the opportunity to work in such a great team.

A deep gratitude belongs to my parents. Without your help my studies would not have been possible.

I would also like to thank my girlfriend Nadja who always supported me during my studies.

Contents

CHAPTER 1

Introduction

Since the invention of the laser in the 1950s, based on Einsteins description of stimulated emission of atoms from 1916, it has become a widely used tool in different areas ranging from basic research over medicine to military applications [1]. One application in physics is trapping of small particles like atoms. After reporting the acceleration and trapping of micrometer-sized particles by Arthur Ashkin in 1970 [2] he and his colleagues showed how to trap dielectric particles with the gradient forces of laser light [3]. The technique of optical tweezers can be used in various fields of different scientific areas. For example Ashkin et al. already considered these traps for manipulation of single biological cells at the very beginning [4]. Because of the ability to focus down the laser beam size to nanometer regime it is also considered as a useful tool in nanotechnology [5].

Many applications of optical tweezers need multiple controllable tweezers at the same time. This can be achieved by deflecting a single laser beam into multiple directions and focusing them at different spots in order to create arrays of tweezers. There are many different methods how these deflections can be generated. For example spatial light modulators (SLM) can manipulate light by changing the optical properties of its material to change the phase front of the light. However, for commercial devices the modulation time usually lies in the ms regime [6] which is too slow for our purposes. But it is worth mentioning that decreasing modulation speed below this limit is focus of current research [7]. Digital micromirror devices (DMD) are also able of shaping laser light into multiple optical tweezers by switching individual micrometer-sized mirrors on or off. Refresh rates in the μs regime are possible to provide high precision control for generating optical traps [8]. In this thesis two crossed acousto-optical deflectors (AOD) are used [9] which rely on the acousto-optical effect. AODs allow to dynamically steer tweezers independently [10] and by this offer the possibility to rearrange tweezer arrays to create defect-free array structures [11]. Since loading tweezers arrays with atoms is a challenging process due to its stochastic nature, it is still part of current research to find optimal routines for loading [12, 13]. The 2D AOD used in this thesis provides access time in the μs regime and has already been successfully implemented in another research group at the Institute of Applied Physics for steering of laser beams [14]. The deflected laser beams are controlled via multitone radio frequency (RF) signals and hence fast control electronics is needed in order to manipulate atoms with high fidelity.

So far direct digital synthesizer (DDS) boards are used in the rubidium quantum optics (RQO) experiment of the Nonlinear Quantum Optics group of Sebastian Hofferberth. The generation of three atom tweezers clouds is possible at the moment which shall be promoted to so called superatoms, which promise to be a versatile tool for optical quantum technology. The idea is to excite one rubidium atom in a trapped atom cloud in a Rydberg state. Due to the dipole interaction induced Rydberg blockade the other rubidium atoms in the cloud cannot be excited by the same light field during lifetime of the excited state. As a result the atom ensemble in the cloud form a collective excited state, a superatom. When the superatom spontaneously de-excites the photon is emitted along the same direction as the incident light field [15]. Placing many atom clouds along an axis defined by the exciting light field then lead to a cascaded quantum system. With such a scheme deterministic photon substraction was achieved in this group [16]. Here up to three photons of a probe light pulse were substracted by the excitation of three superatoms as shown in the sketch below.

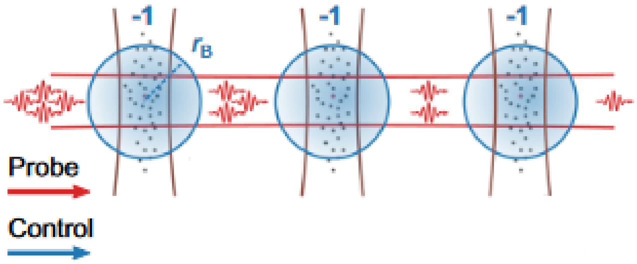


Figure 1.1: Sketch of photon subtraction from a probe light pulse due to excitation of Rydberg superatoms enabled by a control beam; the picture was adapted from [16]

By this mechanism it is possible to deterministically manipulate photon statistics. In order to extend the capabilities of such experiments, in the RQO group the number of tweezers and control shall be extended, which is the main part of this work. This thesis deals with making tweezer arrays with highly controllable arbitrary behavior in time and space. The essential physical theory to understand the AOD and tweezer mechanism will be explained in the next chapter. As a RF source a programmable Spectrum Instrumentation (SI) arbitrary waveform generator (AWG) is used. Because the signal computations are very heavy a GPU is used for this task. Chapter 3 explains the calculations and output characterizations of the SI card. In chapter 4 the optical test setup, which forms the main part of the thesis, will be shown. Here all system components for generating stable tweezer patterns are explained and possible optimization techniques are discussed. Chapter 5 deals with the integration into the main experiment which consists of technical implementation as well as testing the tweezer ouput for atom trapping. This thesis closes with a summary and an outlook how the insights of this work could be utilized in the future of the main experiment.

CHAPTER 2

Theory

This chapter considers the basic physical concepts used in this thesis in order to understand the experimental realization. Firstly the acousto-optic effect and its use for acousto-optic modulators and deflectors will be explained. Then the governing equations for optical tweezers with laser light will be shown and interpreted.

2.1 Acousto-optic deflection

Both acousto-optic modulators (AOMs) and AODs rely on the acousto-optic effect. First the phenomenon is explained for an isotropic medium. This principle will be extended for a birefringent crystal to describe the behavior of the AOD which is used in this thesis.

If a sound wave travels through an optical medium matter is compressed or rarefacted. Thus the mediums density varies over the region where the sound wave is present and therefore its refractive index is changed. If light passes this region it is altered by the sound wave. This result is named acousto-optic effect. Usually the sound wave is created by a piezo element driven by a radio-frequency signal.

In order to derive the governing equations one can use the picture of Bragg Diffraction. Here it is assumed that both light and sound are plane waves with wavelengths $\lambda = \lambda_0/n$ and Λ respectively where λ_0 is the lights vacuum wavelength and n the unperturbed refractive index. Light enters the perturbed region under an angle θ as it is shown in Figure 2.1 on the left side. We assume now that reflection occurs at density maxima and partial light strains can interfere constructively if their path length differs by a multiple of λ. Geometric considerations show that the path length difference is $2\Lambda\sin(\theta)$ (see Figure 2.1 right side). Hence light is diffracted at the same angle which is given by

$$\sin(\theta) = \frac{m\lambda}{2\Lambda} \tag{2.1}$$

where $m \in \mathbb{Z}$. In the following we will only consider the first order ($m = 1$) because it gets the highest diffraction intensity. A more rigorous derivation is done in Ref. [17] by using wave optics. It shows that the frequency of the light wave is altered by the sound for $\pm\theta$ as

$$\omega_d = \omega \pm \Omega$$

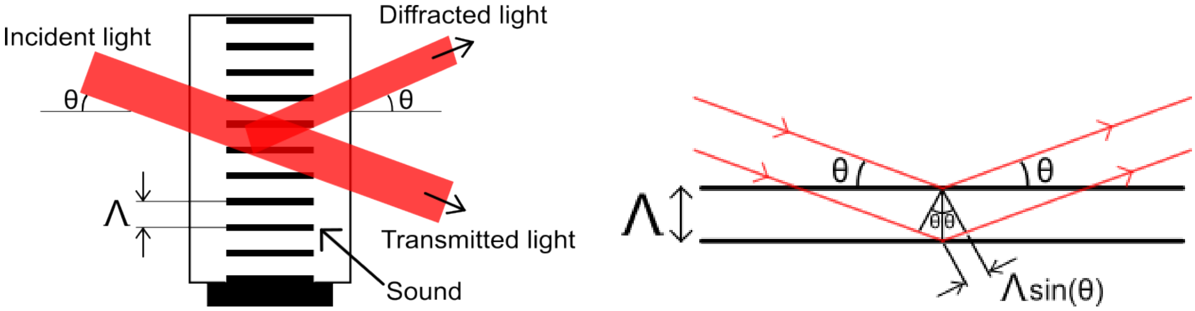


Figure 2.1: Left: Bragg diffraction planar scheme adapted from [17]: Incident light impinging with Bragg angle θ is diffracted partly into light leaving with Bragg angle
Right: Bragg Condition for an AOM: the Bragg angle can be derived from a constructive interference condition where the additional path length of laser light is a multiple of the wavelength

where ω_d is the frequency of the diffracted light wave, ω the original one and Ω the frequency of the sound wave. Usually the condition $\Omega \ll \omega$ holds and the assumption that the wavelength stays the same is valid which leads to a simple quantum mechanical picture of the process: A photon of frequency ω absorbs a phonon of frequency Ω. After scattering the phonon has a frequency ω_d. Conservation of energy and momentum lead to

$$\hbar\omega_d = \hbar\omega \pm \hbar\Omega \approx \hbar\omega \tag{2.2}$$

$$\hbar\vec{k}_d = \hbar\vec{k}_i \pm \hbar\vec{k}_a \tag{2.3}$$

Here $\vec{k}_d$, $\vec{k}_i$ and $\vec{k}_a$ are the wavevectors of diffracted light, incident light and acoustic wave respectively. Using the approximation given in eq. 2.2 one can extend eq. 2.3 to

$$|\vec{k}_d| = |\vec{k}_i \pm \vec{k}_a| = |\vec{k}|$$

Thus the wavevector of the diffracted light lies on a sphere with radius $|\vec{k}|$, depicted in Figure 2.3 on the left where the three wavevectors form an isosceles triangle. It can be seen that eq. 2.1 can also be derived from this phasematching condition. The scheme can now be used to understand the diffraction of a non planar light wave e.g. a Gaussian beam from an acoustic beam. As depicted in Figure 2.2 on the left side light and sound beam have an angular divergence of $\delta\theta$ and $\delta\theta_s$ respectively.
Both beams can be imagined as composed of multiple planar waves with travel directions governed by $\delta\theta$ and $\delta\theta_s$. If now the central wavevectors of both incident light and sound beam fulfill eq. 2.1 and $\delta\theta_s \gg \delta\theta$ then the light beam is reflected in its full shape. This is because for every light wavevector $\vec{k}_i$ there is a sound wavevector $\vec{k}_a$ such that the resulting $\vec{k}_d$ lies on the sphere with radius $|\vec{k}|$ as it is sketched in Figure 2.2 on the right side.
In the scheme above effectively only one diffraction angle can be used and the laser beam manipulation concentrates on the modulation of the RF amplitude. Here a change in RF amplitude alters the diffraction efficiency and hence the diffracted light intensity. To extend this to both angle and intensity manipulation acousto-optical deflectors are used. The difference to the process above is that a deflector needs an anisotropic material. In our case TeO_2 is used which is actually a common crystal choice. The basic working principle can be explained with Figure 2.3 right side. Here a wavevector diagram for

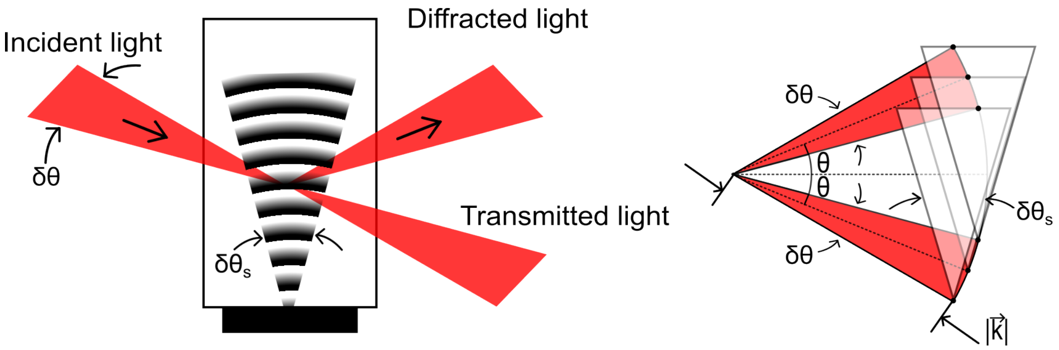


Figure 2.2: Bragg diffraction beam scheme adapted from [17]: an incident laser beam has an angular width $\delta\theta$ and the sound wave an angular width $\delta\theta_s$, the full beam is then diffracted if $\delta\theta_s \gg \delta\theta$ such that phase matching is possible for every wavevector of the incident laser beam

a positive uniaxial crystal is shown where the incident beam as an extraordinary wave with wavevector $\vec{k}_i$ is diffracted into an ordinary wave with wavevector $\vec{k}_d$ by an acoustic wave with wavevector $\vec{k}_a$. Due to the anisotropy of the material the wavevector triangle for phasematching is not isosceles anymore and the bandwidth for this diffraction process can be maximized if the acoustic wavevector is tangential to the inner curve of the diffracted wave [18].

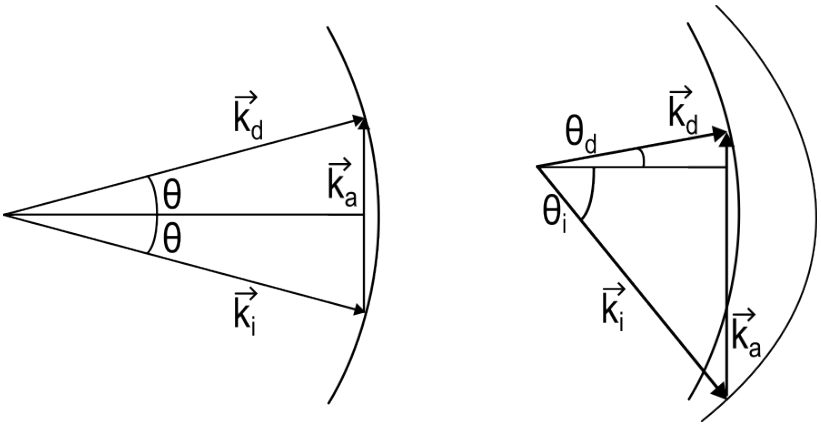


Figure 2.3: Phase matching condition for Isotropic diffraction (left) and diffraction in a positive uniaxial crystal (right) adapted from [19]

Due to energy and momentum conservation the incident and diffracted angles are now described by Dixons equations [20]:

$$\sin(\theta_i) = \frac{\lambda_0}{2n_i v}\left[f_{\mathrm{RF}} + \frac{v^2}{f_{\mathrm{RF}}\lambda_0^2}\left(n_i^2 - n_d^2\right)\right] \tag{2.4}$$

$$\sin(\theta_d) = \frac{\lambda_0}{2n_d v}\left[f_{\mathrm{RF}} - \frac{v^2}{f_{\mathrm{RF}}\lambda_0^2}\left(n_i^2 - n_d^2\right)\right] \tag{2.5}$$

Here n_i and n_d are the refractive index for incident and diffracted light beam respectively, v the speed of sound in the medium and f_{RF} the frequency of the RF signal and hence of the acoustic wave. An important characteristic behavior can be seen in vicinity of the central frequency [20]

$$f_{\mathrm{cen}} = \frac{v}{\lambda_0}\sqrt{\left|n_i^2 - n_d^2\right|}$$

Defining $\Delta n = n_i^2 - n_d^2$ and writing eqs. 2.4-2.5 in terms of the central frequency ($x = f_{\mathrm{RF}}/f_{\mathrm{cen}}$) one yields the following expressions by Taylor expanding at $x = 1$ to first order:

$$\begin{aligned}
\sin\left(\theta_i\right) &= \frac{\sqrt{\Delta n}}{2n_i}\left(x + \frac{1}{x}\right) \approx \frac{\sqrt{\Delta n}}{n_i} \\
\sin\left(\theta_d\right) &= \frac{\sqrt{\Delta n}}{2n_d}\left(x - \frac{1}{x}\right) \approx \frac{\sqrt{\Delta n}}{n_d}(x-1)
\end{aligned}$$

It was assumed that we have a positive uniaxial crystal thus $n_i > n_d$. One can see that in vicinity of the central frequency the diffraction angle shows a linear behavior while the incident angle is constant. This is the working regime of our AOD with high deflection efficiency where the angle of the diffracted beam can be chosen depending on the applied radio frequency in a linear regime. To extend this a RF signal consisting of many different frequency components can be applied such that a single laser beam is deflected into multiple angles at the same time.

2.2 Optical tweezer

In order to trap an atom it has to experience a potential with minimum at the desired trapping spot. An optical tweezer utilizes the dipole potential from laser light to create a deep localized trapping spot. To this end a Gaussian beam is used. Its most important properties for this thesis are explained in the following lines [17]: The intensity of a Gaussian beam is described by

$$I(\rho, z) = I_0\left[\frac{W_0}{W(z)}\right]^2 \exp\left[-\frac{2\rho^2}{W^2(z)}\right] \tag{2.6}$$

with $\rho = \sqrt{\left(x^2 + y^2\right)}$ and z as polar coordinates where the beam propagates along the z-axis. I_0 is the intensity at $z = \rho = 0$. The other parameters are given as

$$\begin{aligned}
W_0 &= \left(\frac{\lambda z_0}{\pi}\right)^{\frac{1}{2}} \\
z_0 &= \frac{\pi W_0^2}{\lambda} \\
W(z) &= W_0\left[1 + \left(\frac{z}{z_0}\right)^2\right]^{\frac{1}{2}}
\end{aligned}$$

with λ as the wavelength of the laser. $W(z)$ is a measure for the beam width by marking the radius at which the peak intensity at $\rho = 0$ has dropped by a factor of $\exp(-2)$. z_0 is called Rayleigh range, at this point $W(z) = W_0\sqrt{2}$. Figure 2.4 shows two plots regarding the Gaussian beam. On the left side the beam width is shown in units of the beam waist W_0. It can be seen that the Rayleigh range is a good measure to characterize how good a Gaussian beam is collimated: The longer the Rayleigh range the longer it takes the beam to diverge significantly. On the right side a heat map of the intensity profile normalized by maximum intensity at distance z is shown with the coordinate axis in units of the beam width W. The red circle marks the region of $\rho = W$.

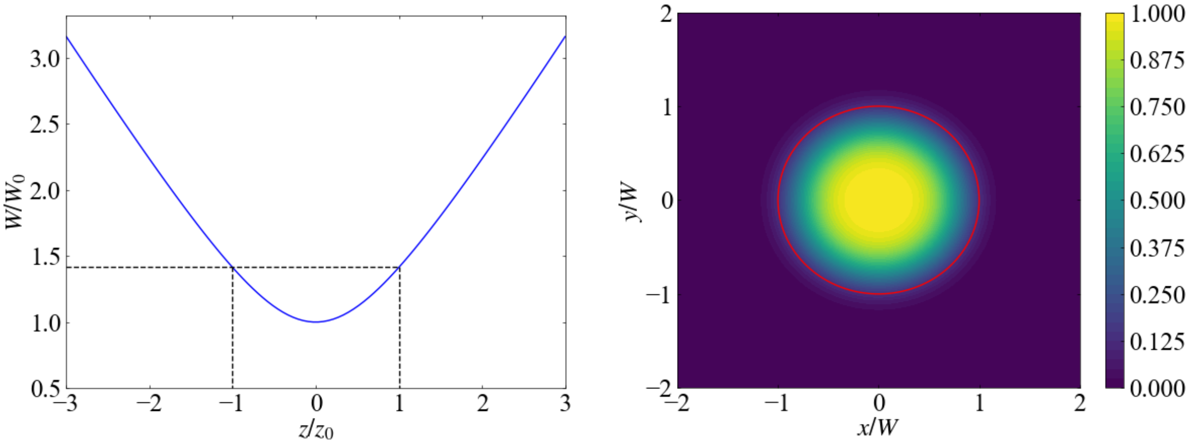


Figure 2.4: Left: Beam waist W of a Gaussian laser beam depending on the distance z from the waist W_0
Right: Beam intensity profile of a Gaussian laser beam, the 1/e^2 drop in intensity is marked with the red circle

Since W increases in both directions from the beam waist at $z = 0$ one can see together with Figure 2.4 that there is always an intensity gradient pointing to the beam center at the waist. This behavior will cause atoms to be attracted to this point as it will be shown now [21]:
If an atom is placed in the beam the electric field of the laser induces an electric dipole moment which is given by

$$\tilde{p} = \alpha \tilde{E}$$

where $\tilde{p}$ and $\tilde{E}$ are the complex amplitudes of the dipole and electric field respectively. α is the complex polarizability of the atom which is given by

$$\alpha = 6\pi\epsilon_0 c^3 \frac{\Gamma/\omega_0^2}{\omega_0^2 - \omega^2 - i\left(\omega^3/\omega_0^2\right)\Gamma} \tag{2.7}$$

Here ω is the laser frequency, ω_0 the atoms resonance frequency and Γ a scaled classical damping rate due to the radiative energy loss. The interaction energy of dipole and electric field is now given by

$$U_{\text{dip}} = -\frac{1}{2}\langle \vec{p}\vec{E} \rangle = -\frac{1}{2\epsilon_0 c}\text{Re}\left[\alpha\right] I$$

Here I is the light intensity. The factor 1/2 in the interaction potential is needed because the atoms electric dipole moment is not permanent but induced. We can now calculate the force on the atom by

$$\vec{F}_{\text{dip}}\left(\vec{r}\right) = -\nabla U_{\text{dip}} = \frac{1}{2\epsilon_0 c}\text{Re}\left[\alpha\right]\nabla I\left(\vec{r}\right)$$

Note that $\text{Re}\left[\alpha\right]$ is positive for $\omega < \omega_0$ and thus due to the intensity gradient the force points toward the intensity maximum. In case for the Gaussian Beam this is the waist center.

CHAPTER 3

RF signal generation

In order to control the AODs a signal source is needed which provides MHz output for each AOD separately. For our needs an output up to 120 MHz should be possible. The device must be programmable in the millisecond regime for implementation into the experimental cycle. In addition the source should be able to generate arbitrarily formed signal patterns in order to allow multi-tone output for realizing multiple simultaneous tweezers. To meet these requirements an M4i.6621-x8 AWG from Spectrum Instrumentation is used as a RF source. This AWG provides two analog outputs with 625 MS/s maximum sampling rate generated with 16 bit digital signal resolution. The output amplitude can go up to 2.5 V into 50 Ω. The device has two trigger channels and communicates over a PCI x8 Gen2 interface which is also connectable to x16 slots as one can see at the retention hook in Figure 3.1.

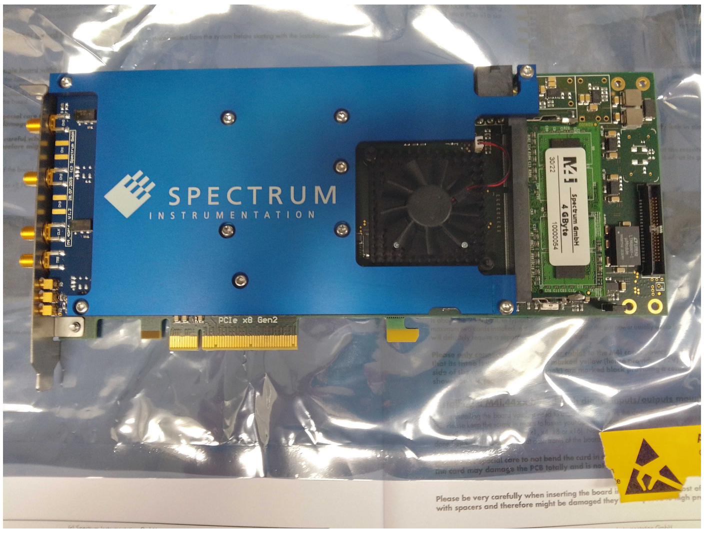


Figure 3.1: Front picture of the AWG placed on the protection foil before installed into the lab PC

In the following section the theory of programming the AWG will be shown. After that methods of signal computation for generating arbitrary output patterns will be explained. At the end the programmed card output is shown and characterized.

3.1 AWG programming

The SI card can be programmed with different programming languages including Python which is used in this thesis. The communication for commands and signal transfer between PC and SI card is managed by the driver provided from Spectrum Instrumentation. The device is controlled by writing values into memory registers of the AWG. This includes features like amplitudes, trigger conditions as well as signal memory management.
In general 16-bit numerical data represented in a buffer object is written into the memory. Each number corresponds to a voltage output. The maximum voltage for each channel is written into additional registers. The output with the memory data is then scaled with respect to the set voltage level. The SI signal is given by the memory data with a speed determined by the sampling rate. Formally the data is saved in a multiplexed way e.g. for two channels the memory scheme looks like this

D11	D21	D12	D22	...

where Dij correspond to data point j for channel i. However, the data is read in parallel for both channels such that the overall output time is given by one channel data length divided by the sampling rate.
For this thesis two modes provided by the device are used: Standard Single Replay and Sequence Replay Mode. The Standard Single Replay Mode only uses one sequence of data with external triggering options. After the data is read from beginning to end it can be repeated instantaneously. Thus the Single Replay Mode provides a fast and easy implementation of a continuous arbitrary signal. A programming example of this method for generating two independent sinus signals is given in appendix A.
The Sequence Replay Mode can be seen as an extension of the Standard Single Replay Mode. Here the memory is divided into multiple segments filled with data. Then it can be defined at which segment the output starts and what the order of sequences with additional conditions is. An example scheme is shown in Figure 3.2.

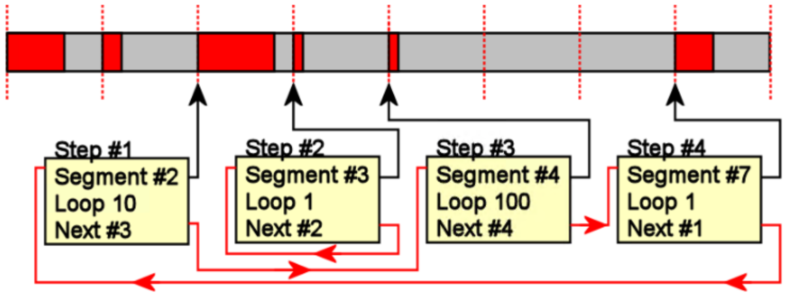


Figure 3.2: Example sequence scheme for Sequence Replay Mode of the Spectrum Instrumentation AWG [22]

In this example the memory is divided into eight segments from which six are loaded with data marked in red. As it can be seen, it is not mandatory to fill the whole segment memory with data. The sequence contains of four steps with each of their own loops and next step settings. There are so called feature flags which determine the behavior of the card after one step is finished. Following options exists:

Feature flag	Description
ENDLOOPALWAYS	Continue with next step unconditionally
ENDLOOPONTRIG	Continue with next step if trigger condition is met
END	Stop card

The feature flag takes place when the loop cycle for the corresponding step is finished. For example in the experiment the trigger feature flag is used to define precisely when the SI card starts its output in the experiment cycle.

3.2 Signal computation

In order to use the SI card to drive the AODs it has to provide sinusoidal signals. The Standard Single Replay Mode can be used but the finite sampling rate and the finite sample length impose a restriction on the signal quality. The problem is that one needs a segment length such that the signal from last to first step is continuous because it repeats at the end. This is illustrated in Figure 3.3. Here an arbitrary sample length of fourteen samples was chosen. In theory, after the SI card processed these data points it would re-loop. This is depicted by continuing the signal with exactly the same data. The red square marks the region where the loop occurs. It is visible that the signal cannot be continued smoothly, a jump appears. In practice this would lead to additional frequency components in the power spectrum of the SI output. The goal is then to find a sample length such that last and first signal point would lead to a continuous output.

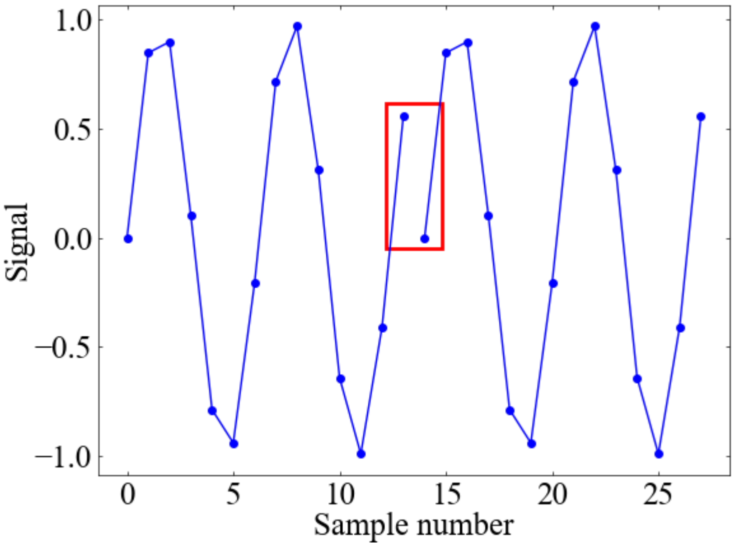


Figure 3.3: Signal jump of a simulated sinus signal due to finite sampling rate, the sample points are connected with straight lines, the red rectangle marks the region where the jump appears

One method to realize this is taken from Ref. [23] and will be explained in the first part of this section. The minimum sequence length is given by the card as $L_{\text{min}} = 384$ samples. If the segment length is longer then the additional length has to be a multiple of 32 due to internal AWG restrictions. In the end the goal is to create a signal $S_\nu(i)$ with frequency $f = \nu f_{\text{cen}}$ with f_{cen} as a central frequency (not the same as for the AOD) such that $S_\nu(L_{\text{min}}) = S_\nu(0)$. We can define the following parameters:

$$N_{\text{loop}} = \left\lceil \frac{L_{\text{min}} \cdot f_{\text{cen}}}{625\ \text{MS/s}} \right\rceil \quad f_{\text{SR}} = \frac{L_{\text{min}} \cdot f_{\text{cen}}}{N_{\text{loop}}} \tag{3.1}$$

Here N_{loop} is the minimum number of loops needed for a signal with central frequency when sampled with sampling rate f_{SR}. $\lceil x \rceil$ is the ceiling function such that the sampling rate is adjusted to f_{SR}. The signal is then given by

$$S_\nu(i) = \sin\left(2\pi \cdot N_{\text{loop}} \cdot \frac{i}{L_{\text{min}}} \cdot \nu + \phi_0\right) \tag{3.2}$$

where ϕ_0 is an optional phase shift. The signal can be scaled by a 16 bit integer. In order to meet the boundary condition mentioned above one can see that ν has to be a rational number with denominator of N_{loop}.

In order to be able to apply a finer frequency spacing one then needs to extend this data sample with length L_{min} to a multiple of it:

$$L_{\text{min}} \longrightarrow N_{\text{length}} L_{\text{min}}$$

Because of the above mentioned sample length restriction N_{length} needs to fulfill the following condition:

$$\begin{aligned} &N_{\text{length}} L_{\text{min}} = 32 \cdot m \quad m \in \mathbb{N} \\ \Leftrightarrow &12 \cdot N_{\text{length}} = m \end{aligned} \tag{3.3}$$

The frequency precision p can be defined as

$$p = f_{n+1} - f_n = \frac{f_{\text{cen}}}{N_{\text{loop}} N_{\text{length}}}$$

The sample length can then be expressed as

$$N_{\text{length}} = \frac{f_{\text{cen}}}{N_{\text{loop}} p} \tag{3.4}$$

Combining eq. 3.3 and eq. 3.4 we get a an expression for all possible frequency precisions:

$$p = \frac{12 \cdot f_{\text{cen}}}{N_{\text{loop}} \cdot m} \tag{3.5}$$

In total eqs. 3.1, 3.2, 3.4 and 3.5 fully define the signal data for a limited segment. This technique can already be utilized for tweezer array generation: Using the Sequence Replay mode each segment would then contain one frequency pair (one frequency for each output channel) corresponding to one tweezer. If the phase shift ϕ_0 is equal for every segment the output signal switches continuously from one frequency component to the next one. This fast tweezer ’jumping’ then creates an averaged potential for every tweezer [24]. One restriction of this method is the fact that every tweezer is only active for finite time and inactive as long as the signal cares for the other frequency components. Thus the signal must drive back to the original place in a short amount of time that depends on the atoms reaction when no tweezing force appears. In conclusion the signal length N_{length} imposes then restrictions on the frequency precision p.

A more robust but less flexible method is to provide continuous signal output with amplitude and frequency depending on time. It can be realized with the following formula:

$$S(t) = A(t) \cdot \sin\left(2\pi \cdot \int_0^t f\left(t'\right) dt' + \phi_0\right) \tag{3.6}$$

Here $A(t)$ and $f(t)$ are the time dependent amplitude and frequency respectively. ϕ_0 is an optional phase shift. The integral is needed to take previously collected phases from other frequencies into account. On top of this the time dependent amplitude can be simply multiplied since it is not phase dependent.

In discrete form for computational implementation eq. 3.6 becomes

$$S(t_i) = A(t_i) \cdot \sin\left(2\pi \cdot \sum_{j=0}^{i-1} f\left(t_j\right) \Delta t + \phi_0\right) \tag{3.7}$$

where $\Delta t = f_{\text{SR}}^{-1}$.

In order to apply multi-tone signals onto the AODs the data for different frequencies (tweezers) can be summed up:

$$S_{\text{total}}(t_i) = \sum_{d=1}^{N} S_d(t_i)$$

The downside of this superposition method is that for a 2D AOD combination every frequency component of x-diffraction axis couples with every other from the y-axis. Thus only 1D or quadratic patterns are possible in contrast to the former method where every tweezer has its own time sequence. Due to this separation only one frequency at each AOD appear at a certain point in time and thus only these frequencies couple together. Two example configurations in 2D are shown in Figure 3.4 which were created with the ’jumping’ method and recorded with a camera. The potential averaging mentioned above can be seen here effectively in the overall lower brightness for the spots in the left picture. Here much more spots are generated and hence the off-time is longer than for the triangle shape in the right picture. In the later part of this thesis we will only focus on the 1D part with the superposition method because it is used in the main experiment.

While computations for the ’jumping’ method can be run on CPU with short calculation times, the latter method can consume times of multiple seconds. For our experimental purposes this is not fast enough, the experiment cycle time is two seconds [25] and we want the SI card to be ready in under one second. Therefore the computations for these long samples are performed on a *Nvidia GeForce RTX 3060TI* graphics card. The reason why GPU calculation speed is faster than that of CPU is parallelization. Usually, commercial CPUs contain only below hundred cores which are the computation units. Arithmetic operations can be distributed among them such that computation time

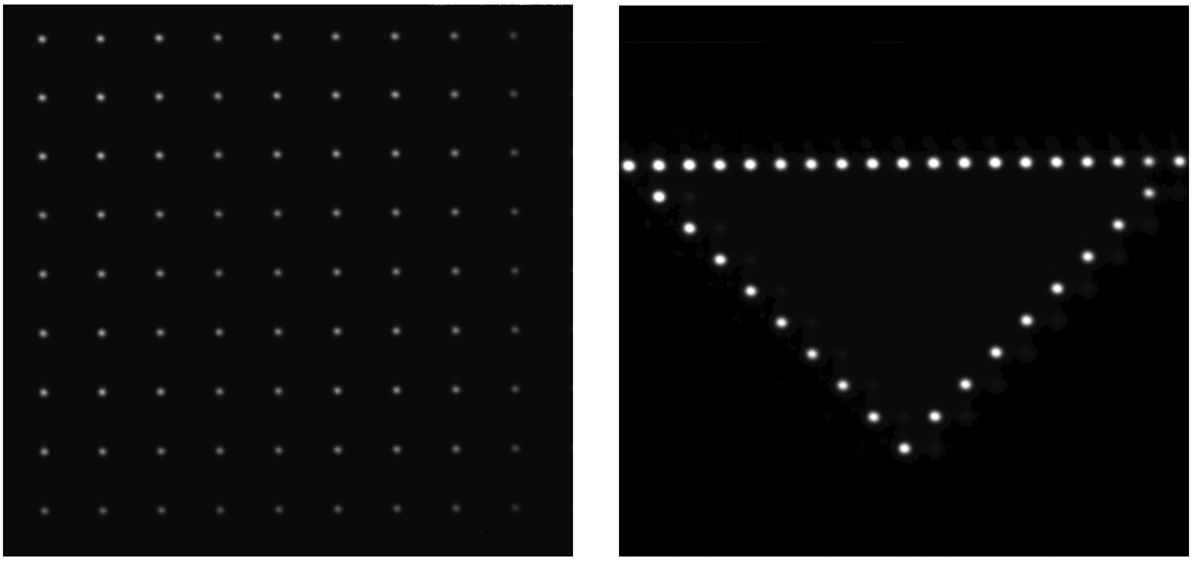


Figure 3.4: 2D square and triangle light pattern generated with tweezer 'jumping' method recorced with a camera

decreases because sub tasks are solved in parallel. For complicated and long calculations like graphics calculations for modern computer games CPU power is not enough to finish these tasks in a short amount of time. This also holds for numeric scientific problems like our case. A GPU has thousands of cores thus computations are faster. The computation scheme in our case can be summarized in Figure 3.5 shown below. Three instances are used: GPU, CPU and SI AWG card. The information transfer is governed by PCIe lines indicated with black arrows. The central part is the processor which coordinates the information stream with the graphics card as well as setting the spectrum card over registers and final signal stream. Possible feedback from the spectrum card is handled in terms of error queries. As already explained, heavy calculation for eq. 3.7 are done via graphics card. Necessary information about what to compute is streamed from CPU to GPU. On the other side, fully calculated data is given back from GPU to CPU.

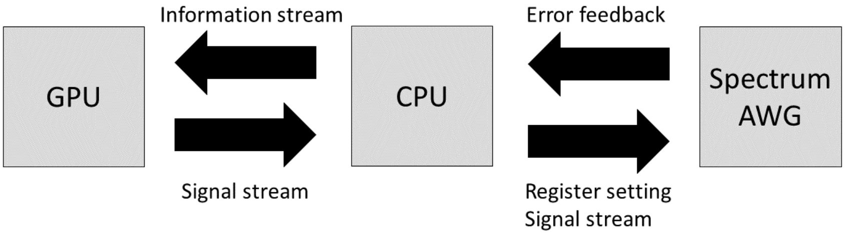


Figure 3.5: Scheme of computational signal flow between PC and Spectrum card

3.3 Output characterization

For creating well-defined tweezers one needs to know the technical possibilities and limitation of the SI card as well as the software performance. The interesting parameters here are amplitude and frequency behavior as well as computation times of the RF signals. To characterize the SI ouput a Agilent Technologies EXA N9010A spectrum analyzer was used. In Figure 3.6 the upper plots show the measured output power with respect to the software set amplitude for channel 0 and channel 1 respectively. For better visualization the same data is shown with logarithmic x-scales in the bottom line in addition with a reference line which shows the theoretically expected output.

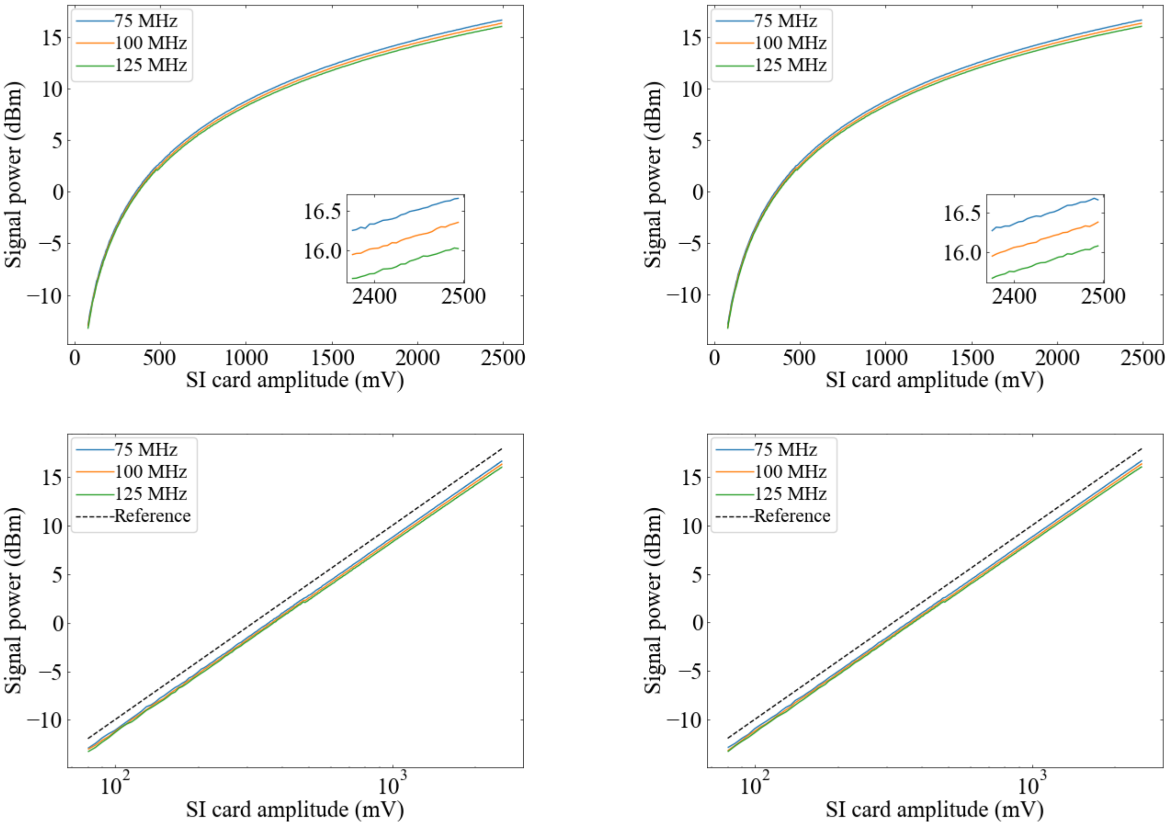


Figure 3.6: Power output of SI card with respect to theoretically set amplitude
left side: ch0 with/without reference; right side: ch1 with/without reference
bottom plots are shown with logarithmic x-scale

Two things are apparent: the output power varies slightly with frequency and is lower than 1 dB apart from the expected output. The reason for this is the finite sampling rate of the SI card. The peak values of the sinus signal lies with low probability on a sample point. In addition since we work with fixed sampling frequency the location of maximum value sample points varies with desired signal frequency. Therefore the real output has to lie always below the reference line. This effect is illustrated in Figure 3.7. Here a snippet of simulated sinus samples for three frequencies are shown. One can see that the sample points cover the ideal signals only roughly which is the reason why the detailed calculation scheme described in section 3.2 is needed.

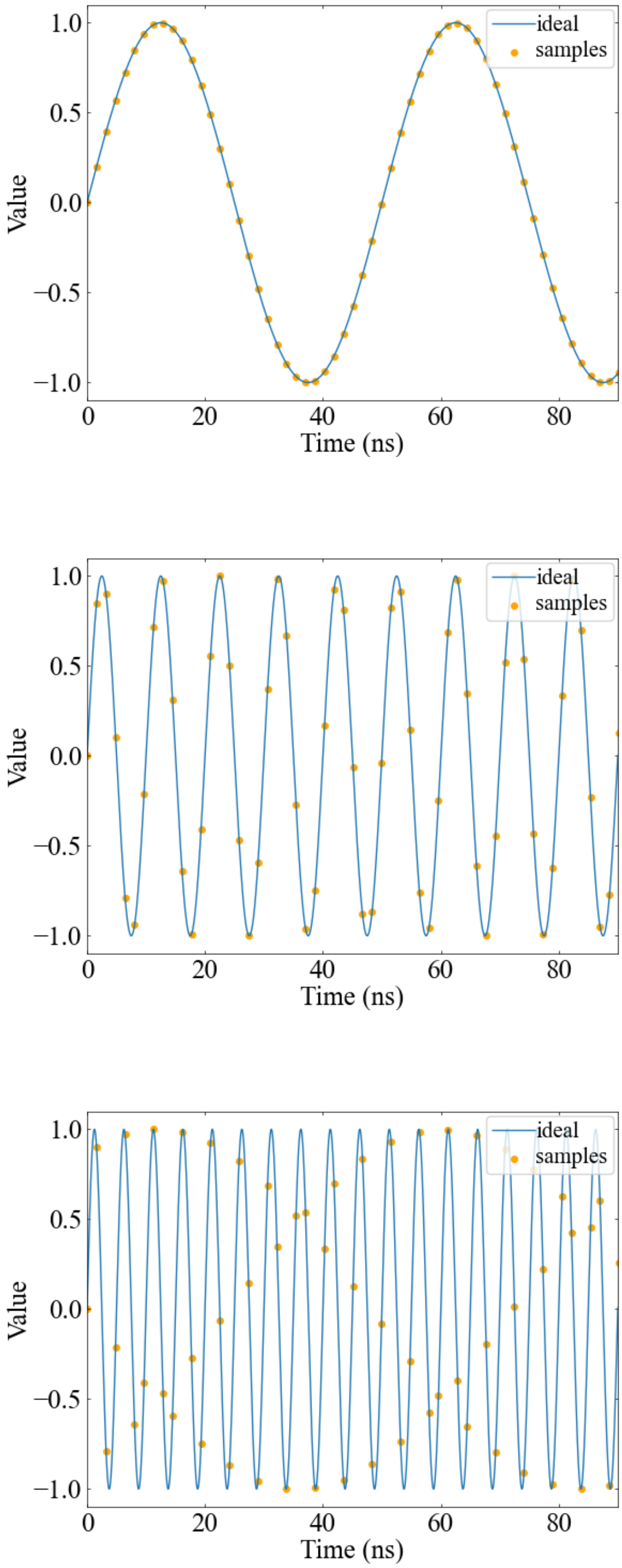


Figure 3.7: Simulated sample data in comparison to ideal sinus signal for 20, 100, 200 MHz respectively

The change of amplitude in dependence of the programmed frequency is shown in Figure 3.8 for different programmed amplitudes. As expected for higher frequencies the amplitude decreases because of poorer sampling. For lower frequencies the amplitude curves reach a maximum and then fall off again. Just from a standpoint of sampling a plateau should appear because at some point the wavelength is long enough to cover the slopes sufficiently for the finite sampling rate. I assume that the drop off to the left comes from bandwidth limitations of electronic components inside the SI card.

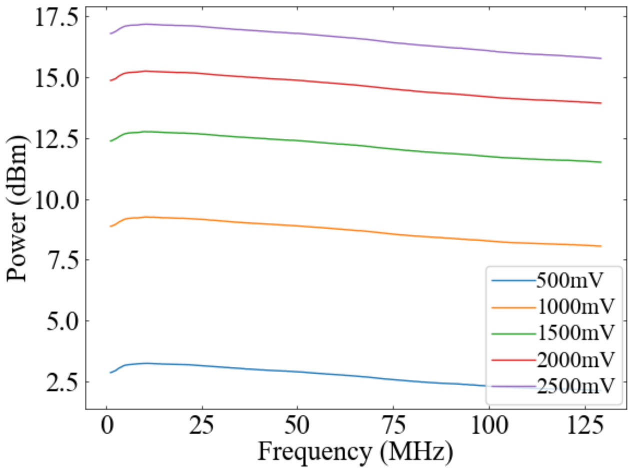


Figure 3.8: SI output power for different set amplitudes in dependence of frequency

For the test setup the SI signal is attenuated and amplified to enhance the deflection efficiency of the AODs. This happens because with increasing RF power the amplitude of the sound wave inside the AOD crystal increases. This corresponds to a higher amount of phonons which increases the probability of deflection. However, it is also important that the maximum power used is below the damage threshold of the AODs specified by the manufacturer. In our case the threshold is 2 W which corresponds to 33 dBm. At the same time we want to use the whole amplitude range of the SI card, namely 2500 mV, thus a combination of attenuator and amplifier was selected such that the output power is as high as possible but below 2 W. The output result is shown in Figure 3.9. The overall shape stays the same like in Figure 3.6 as expected. Small variations can appear since attenuators and amplifiers are frequency and amplitude dependent.

It can be seen in the insets of Figure 3.9 that the maximum output was measured for 75 MHz at 2500 mV programmed SI amplitude. The power is roughly at 32.8 dBm but still below 33 dBm. Therefore the lowest frequency which can be applied with confidence in meanings of safety is 75 MHz for this setup.

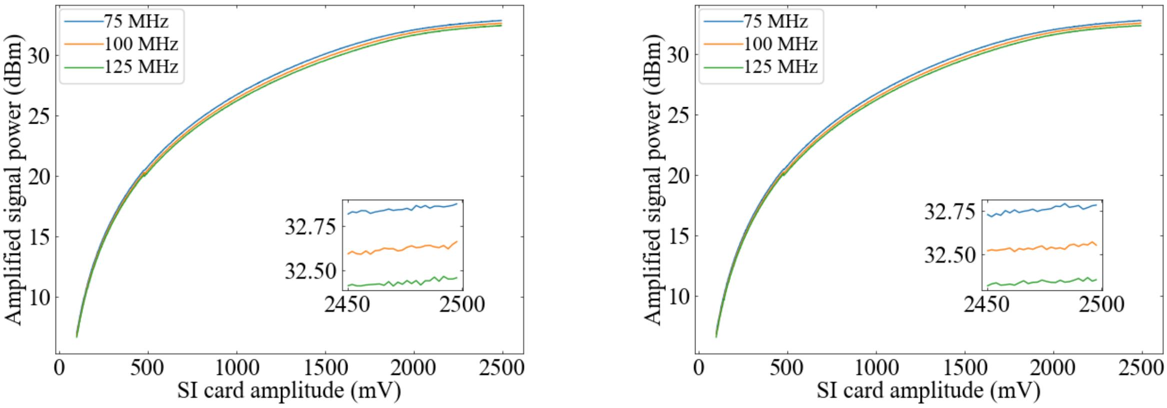


Figure 3.9: Power output of SI card with amplification for channel 0 and channel 1 respectively

Please note that the output is also temperature dependent. With the setup used in this thesis all amplifiers are cooled. If for example the amplifiers for the SI output would not be cooled the maximum amplified output could exceed 2 W.

The other important quantity to check is the match of the output frequency with the theoretically programmed one. To this end the output was measured with the spectrum analyzer at high resolution bandwidth. The result is shown in Figure 3.10 where on the left side the measured frequency output is shown with respect to the programmed value. The linear relationship is clearly visible. A closer look can be taken by subtracting the theory frequency from the measured value to get the frequency deviation Δf. It is shown with respect to the theory frequency in the plot on the right side. The deviation increases with higher set frequency. This is again expected because of poorer sampling quality as for the amplitude case. The frequency deviation increases step wise which I assume is caused by finite digitization. However, in the range we are interested in the deviation is in the sub kHz regime which is perfectly fine since the tweezer resolution in the main experiment is roughly 1 MHz.

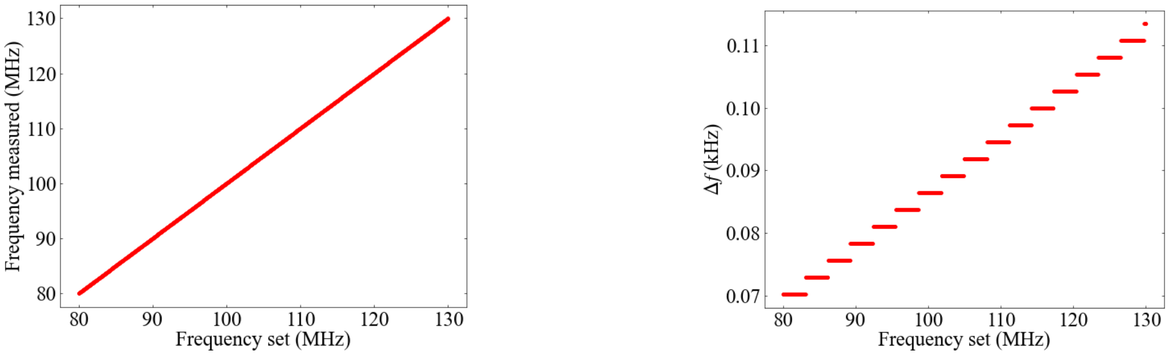


Figure 3.10: Frequency output vs expectation of the SI card, the deviation of real and expected output Δf lies in the kHz regime and increases with increasing set frequency due to poorer sampling

One disadvantage of the use of amplifiers is signal distortion when applying multi-tone input. In Figure 3.11 the power spectrum of a signal programmed originally with frequencies of 75, 80 and 85 MHz are shown in blue. One can clearly see that side peaks are apparent. These appear due to nonlinearities in the amplifier circuits and result in frequency mixing [11]. It is possible to reduce this effect by changing the phases of individual signals. In the plot the spectrum for random phases is shown in orange. One can see that the main amplitudes stay the same while the side-peaks amplitudes are getting damped by several dBs. This damping is important because otherwise unwanted tweezers would be generated. Diminishing the RF amplitude of these signals will then result in less energy contributing into the optical power of these spots. In the end the side peaks have less intensity by several orders of magnitude such that these side effects become negligible. In the main experiment random phases are then used for the RF signals to create the corresponding tweezers for the atom clouds.

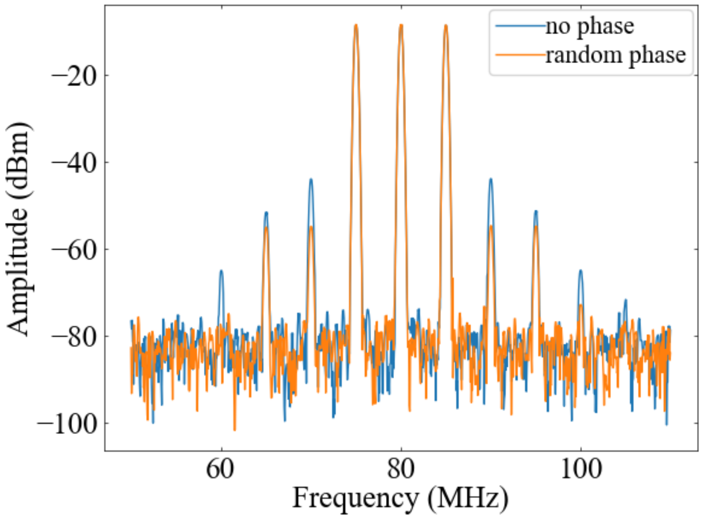


Figure 3.11: Example of frequency mixing due to amplifiers for an input signal originally containing 75, 80, 85 MHz components, the amplitudes of the side peaks can be damped with phase shuffling the original RF signals

One other critical part of the RF generation characterization is the time needed from starting signal computation until the SI card starts its output. For this the performance of a 1D sequence for the main experiment is considered. The (technical) details of this sequence are explained in chapter 5 when the actual implementation into the main experiment is described. For now it is enough to know that a self written software controls signal generation and uploading to SI card. The software enables a user to specify initial tweezer pattern, movement specifications and end pattern. Start and end output are implement with eq. 3.2 whereas the moving is realized with eq. 3.7. The number of tweezers is arbitrary and the movement takes place in the millisecond regime. Thus the user is able to generate an arbitrary tweezer configuration in time and space. The concept is depicted in Figure 3.12 where an arbitrary sequence with four tweezers of different frequency and amplitude behavior was run. The voltage signal was measured by a LeCroy WavePro 7Zi oscilloscope. The data was then split into small time bins and Fourier transformed to get the frequency spectrum over time.
The intensive part of calculating the movement data is done by the GPU. However, at some point for number of tweezers and/or moving time the calculation time can be very long even with GPU

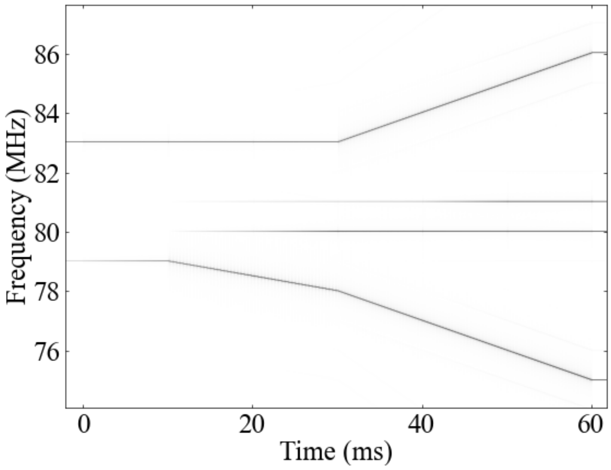


Figure 3.12: Fourier spectrum of example 1D sequence

computing. For benchmark purposes a characteristic pattern was calculated: For simplicity all tweezers start at $t_0 = 0$ with the same initial frequency and end at $t = t_{\text{end}}$ with the same end frequency. In between the frequency increases linearly to the end. The time for movement (simulation) $t_{\text{end}} - t_0$ and number of tweezers were then increased to test the computing performance. The actual computing time until the card is output ready was measured and the mean of ten iterations (of a certain simulation time and tweezer number setting) was taken. The result can be seen in Figure 3.13 in the first plot.
The white region on the first plot marks configurations where the GPU memory had an overflow meaning the data to calculate didn't fit into the GPU memory. The red lines mark the boundary of one second computation time and transition from valid to abandoned data which is the limit for integration into the main experiment. It can be seen that with more tweezers the maximum moving time gets more restricted. However, for application purposes long times are not needed because the moving part is only used for loading and distributing the tweezers effectively which can be done in a few milliseconds.
The bottom plot of Figure 3.13 shows the same benchmark data but here the time values above one second are also discarded such that the color scaling for the time below one second is better visible. It is apparent that increasing tweezer number and/or simulation time increases computation duration. However, what is visible in both plots is that there are sometimes fluctuations where this rule does not apply, e.g. with increasing simulation time sometimes the computation time becomes smaller for a certain setting and then increases again as normally expected. During the process of developing the benchmark scripts I noticed that computation time fluctuates when using the GPU. I assume that different efficiencies in internal memory structure building can cause this behavior. However, for practical implementation this is not a problem for the computing performance we are interested in.

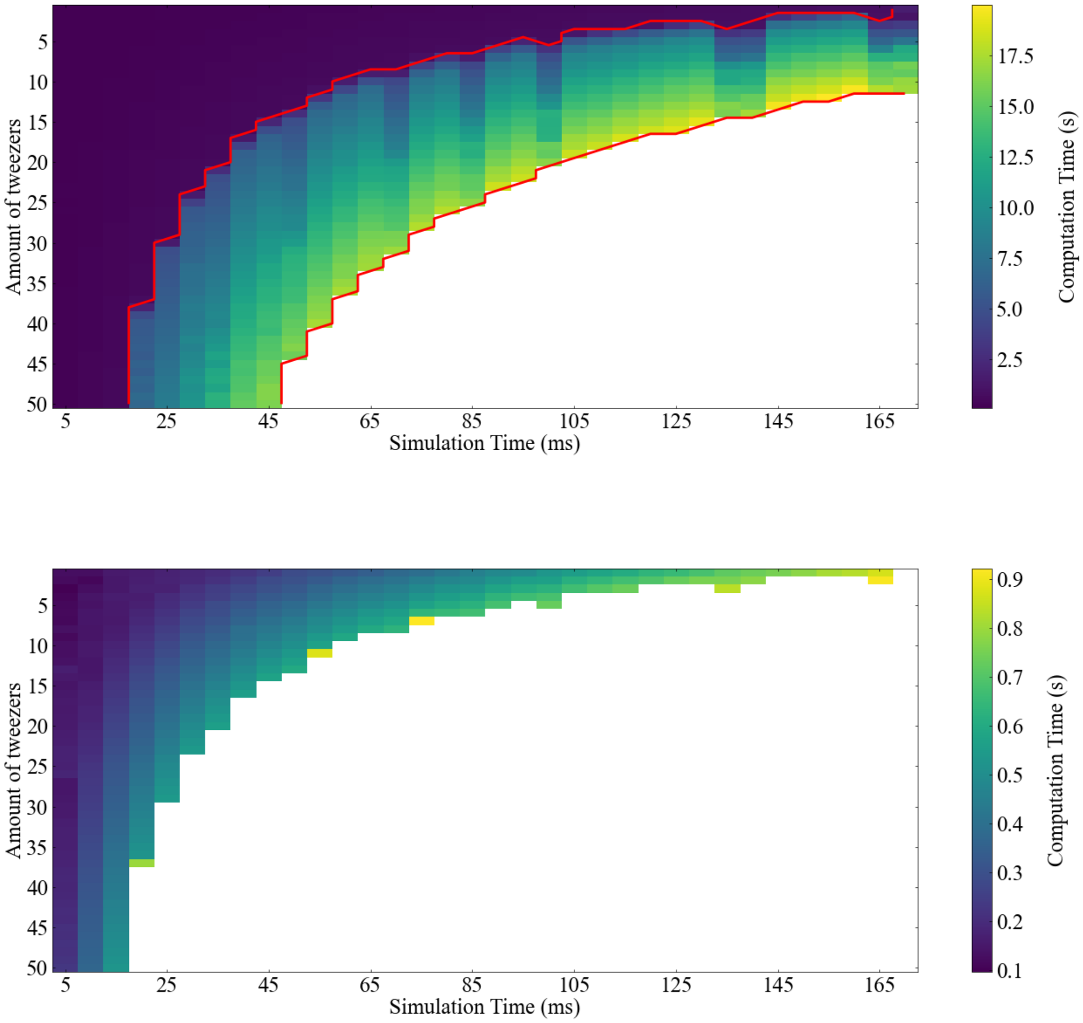


Figure 3.13: 1D tweezer sequence computation time benchmark, the computation time was averaged ten times for each setting, white region in upper plot has no data due to GPU memory overflow, in the lower plot time values above one second were additionally abandoned to the white region such that the color scale in the allowed region is better visible

CHAPTER 4

Creation & optimization of AOD deflections

For testing tweezer generation and characterization an optical test setup was built. In addition it was used to experiment with methods of optimizing AOD deflections in terms of generating equal deflections efficiencies for different frequency settings. In the first part of this chapter the whole setup and its details will be explained. The second part then discusses the deflection efficiencies. The goal of this part of the thesis is to point out technical limitations when working with AOD deflections and to find RF manipulation mechanisms that could be utilized in the main experiment.

4.1 Test setup

In Figure 4.1 a sketch of the optical test setup is shown. The flow of explanation goes from right to left in correspondence with laser propagation direction. Before the laser light can be used for generating tweezer patterns two points must be assured:

- Intensity stability of the laser light source is a necessary condition for producing intensity stable tweezers. Therefore an AOM is used where the intensity of the diffracted laser light is stabilized with a PID loop (discussed in section 4.1.2)
- For producing well defined single tweezer points the light has to be in Gaussian mode. This can be assured by leading the light though a polarization maintaining single mode fiber.

The RF power of the AOM is adjusted by a control box which will be explained subsequently. After the fiber a polarizer is used to ensure vertical polarization (perpendicular to the optics table). This has two reasons. Firstly the two axis AODs are optimized for this polarization and secondly any polarization drifts from the fiber coupling are changed into intensity drifts by the polarizer which can be adjusted by the intensity stabilization.

After the polarizer the laser light is divided into two parts by a beam splitter. One part is further used for tweezer generation, the other one is used for intensity stabilization. The intensity is detected by a photo diode (PD) in terms of photo current. The corresponding voltage signal is fed into a feedback loop that controls the AOM RF amplitude. With this feedback loop the laser light intensity going into the PD is held at a desired point. Therefore also the transmitted light of the beam splitter stays constant. This is then a reliable input for the AODs. The SI card inside the computer produces the desired RF signal which is amplified and fed into the AODs. The diffracted laser light is then focused

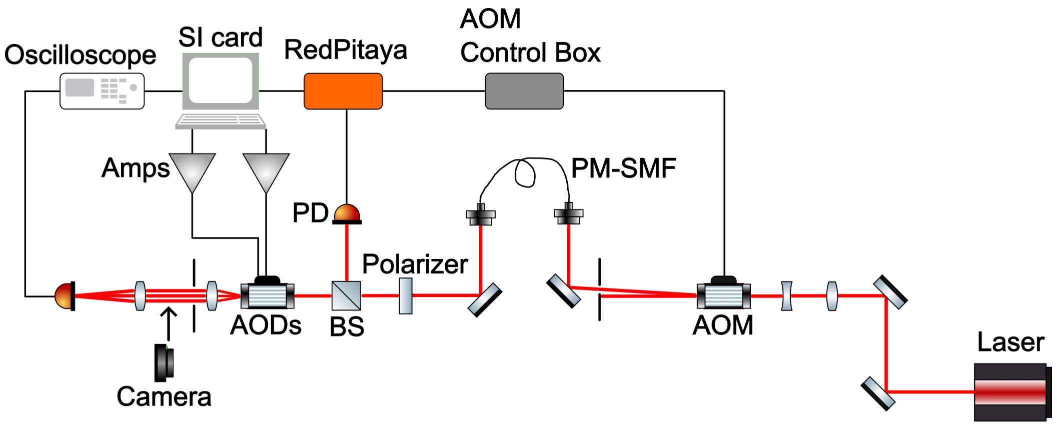


Figure 4.1: Optical test setup for generating and manipulating tweezer arrays

either on a PD or on a camera depending on the kind of measurement. In this thesis intensity related measurements were done with the PD and position depended ones with the camera because it turned out that camera intensity values where too unreliable. This will be discussed in more detail in section 4.2.3. For PD measurements the laser light intensity is turned into a proportional voltage which is measured with an oscilloscope. This value can than be digitized and read by the computer.

4.1.1 AOM control box

Two devices are used for control of the AOM output to produce intensity stabilized laser light: RedPitaya and AOM Control Box. This subsection concentrates on the latter one while the actual intensity control with the RedPitaya will be discussed in the following chapter. The control box is used to generate an RF signal suitable for the AOM and to provide a possibility for manipulation of the amplitude of this signal to change the AOMs diffraction efficiency. These boxes are in-house built for standardizing AOM control. Figure 4.2 shows a scheme of the box.

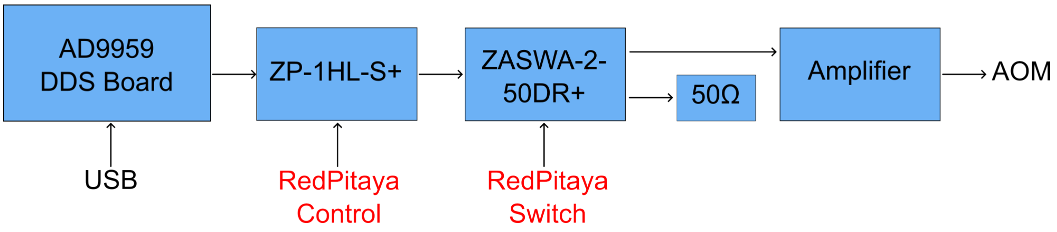


Figure 4.2: Scheme of the AOM control box flow

The RF source for driving the AOM is a AD9959 DDS board from ANALOG DEVICES which is programmed to provide an output of 80 MHz. The programming can be done by a computer with the manufacturers software via USB connection. In order to be able to modify the amplitude of this signal it is mixed with an analog input from the RedPitaya with a Mini-Circuits ZP-1LH frequency mixer. The mixed signal is then passed though a ZASWA-2-50DR+ coaxial switch. The TTL control input is also provided by the RedPitaya. Usually this stage is used for reliable shut-off service and debugging purpose. In this thesis it was also used in the process of intensity stabilization as shown in chapter

4.1.2. If the RF signal is shut off (TTL=0) it is just terminated on a 50 Ω load. Otherwise (TTL=1) the signal is amplified and then fed into the AOM. Practically this scheme after the DDS in Figure 4.2 can be done four times because the DDS board provides four independent output channels. However, in the setup of this thesis only one channel is used since only one AOM is needed. Figure 4.3 shows the inside of the box.

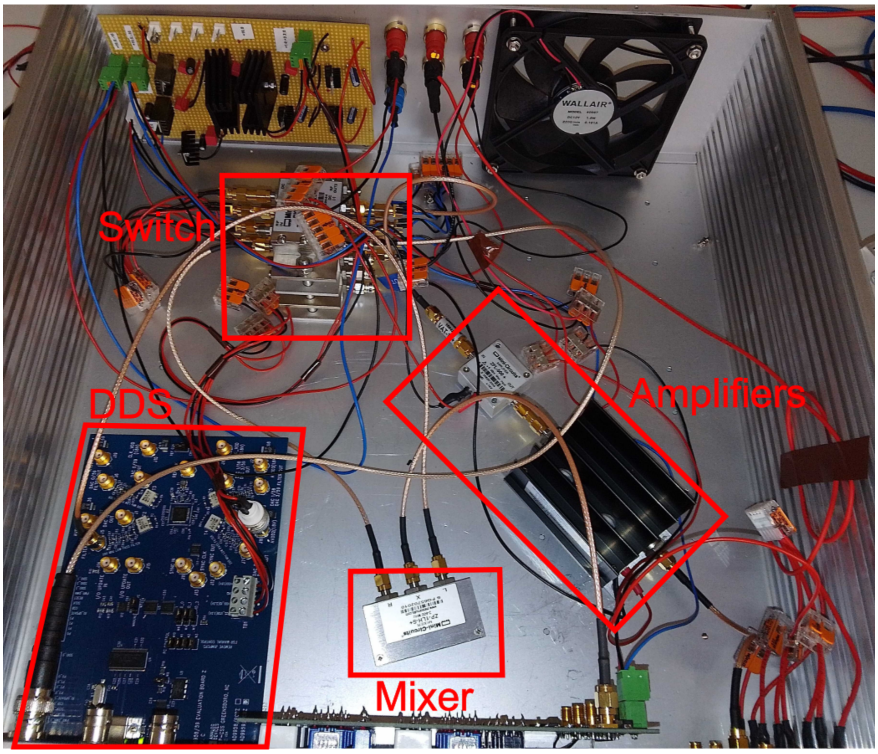


Figure 4.3: Inside of the AOM control box

As one can see on the right side in the middle of the box a pre- and a main amplifier (small gray and big black box) are used to provide suitable output for the AOMs. They need to be cooled during operation and thus a fan is integrated in the wall of the box. Cooling is also necessary for the amplifiers of the SI card signal. Because of the fans vibrations these components are placed on racks isolated from the optical table to not disturb the experiment.
In Figure 4.4 a scan of the mixer response is shown. The RedPitaya output is scanned from 0 to 1 volt and amplified to meet the analog input range. In parallel the laser input signal to the RedPitaya is measured. Because the first order diffraction of the AOM is used it is possible to completely shut down the laser intensity by the mixer.
The purpose of intensity stabilization will be to keep this laser signal constant as accurate as possible by controlling the RedPitaya output.

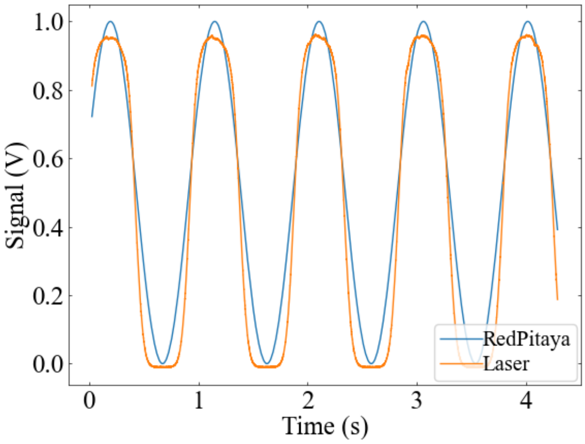


Figure 4.4: Control box scan: The RedPitaya signal modulates the PD laser signal by altering the AOM RF amplitude in the mixer

4.1.2 Laser intensity stabilization

Before diving into the details of intensity stabilization, the RedPitaya device will be discussed in the first part of this section. After that the ideas of feedback-based signal manipulation and PID stabilization is discussed. At the end the results implementing this scheme in the experimental setup are shown.

The company RedPitaya produces FPGA based multifunctional compact devices with the aim to combine multiple laboratory instruments, like oscilloscopes or signal generators, in one. The product used for this thesis is the STEMlab 125-14 which is shown in Figure 4.5. The NQO group also has a standardized PID solution with self manufactured PID boards but here a new solution shall be discussed.

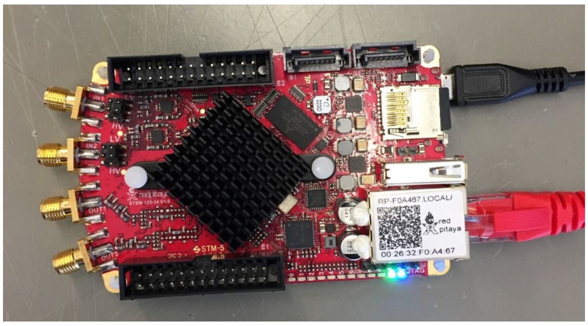


Figure 4.5: Photo of the STEMlab 125-14 device from RedPitaya

In the test setup the device is directly connected to the PC via Ethernet connection. It provides two analog input and output channels. There are different ways to communicate with and set up the RedPitaya for different purposes. In this thesis a python based open-source software package called PyRPL is used [26]. It offers a graphical user interface to set up different functionalities for the device. For intensity stabilization the PI-module is used which will be explained subsequently.

In order to be able to stabilize the used laser light after the AOM one needs a feedback mechanism.

The abstract layer is depicted in 4.6. One can see that a system provides an output y that can be altered actively with some input. The idea is now to feed this signal back to a controller which considers the error value 'e' which is given by the deviation of the output 'y' from a user desired set value 's'. The controller then performs a calculation based on the error and provides an input 'c' to the system. The function that determines the correction signal will depend on a certain amount of parameters which can be tuned. The assumption is now that for certain parameter values the controller is able to efficiently alter the systems behavior such that the output converges to the desired set value.

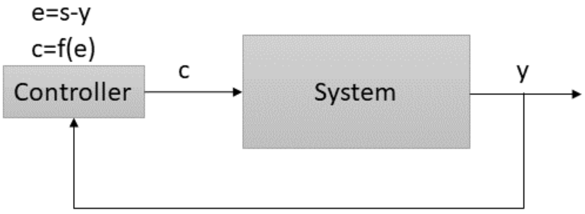


Figure 4.6: Abstract sketch of stabilizing a systems signal with feedback based on the deviation 'e' of the output 'y' from a set value 's'

The controller used in this thesis is the RedPitaya with PID feedback. PID stands for 'Proportional', 'Integral' and 'Derivative' and refers to the mathematical feedback components the algorithm is based on. The formula which produces the controller signal is given by [27]

$$c(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

As one can see the output consists of a proportional, integral and derivative part of the error signal. The proportional constants K_i define the strength of individual terms. Usually proportional and derivative terms are used to stabilize fast fluctuations of the signal and the latter one is meant to be more sensitive since it reacts on the derivative of the error signal. The integral part is essential for letting the output signal reaching the target value in terms of an offset.

In this thesis the derivative part is not used but this is also not necessary for a comparable slow time-scale task as laser intensity stabilization. The proportional constants and the target value can be set individually in the PyRPL GUI. As already mentioned, the PD signal from one laser beam of the beam splitter is used as system output (see Figure 4.1 in section 4.1). It is then compared with the target value and further processed with the PID controller. The RedPitaya then forwards the control signal to the AOM box mixer. The main goal of the intensity stabilization section was then to find optimal values of the proportional constants such that recovering to the target value is (i) as short and (ii) as stable as possible.

The parameters were found with the following process: system and controller output are both monitored with the PyRPL oscilloscope function. The TTL signal which is provided by the second RedPitaya output is varied periodically with a square signal. In parallel the PI controller is running. When the TTL is off, the controller will automatically jump to the maximum value due to a finite integrator part. When the TTL is then activated the controller will try immediately to adjust its output to restore the set value. The oscilloscope can be triggered on the TTL signal such that the effect of altering P and I parameter can be immediately seen. At first sight it may seem overambitious to artificially saturate the controller output when the TTL is off but the advantage is that the temporal efficiency of the setting can be judged well. This behavior is shown in Figure 4.7 for two exemplary settings. The controller is told to stabilize the system signal at 0.8 V. Both settings have the same I parameter but on the left side the P part is higher than on the right. Lowering the latter parameter reduces overshoots of the control which is clear from the PID formula since the P part is proportional to the error. With this interface the parameter adjustment can be quickly found by just varying two parameters and to see the shape change of controller and system curve.

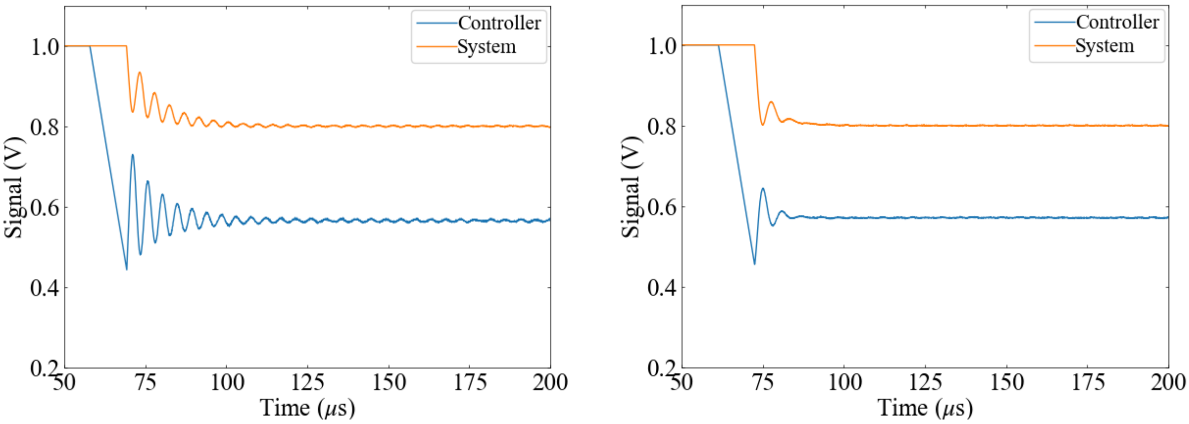
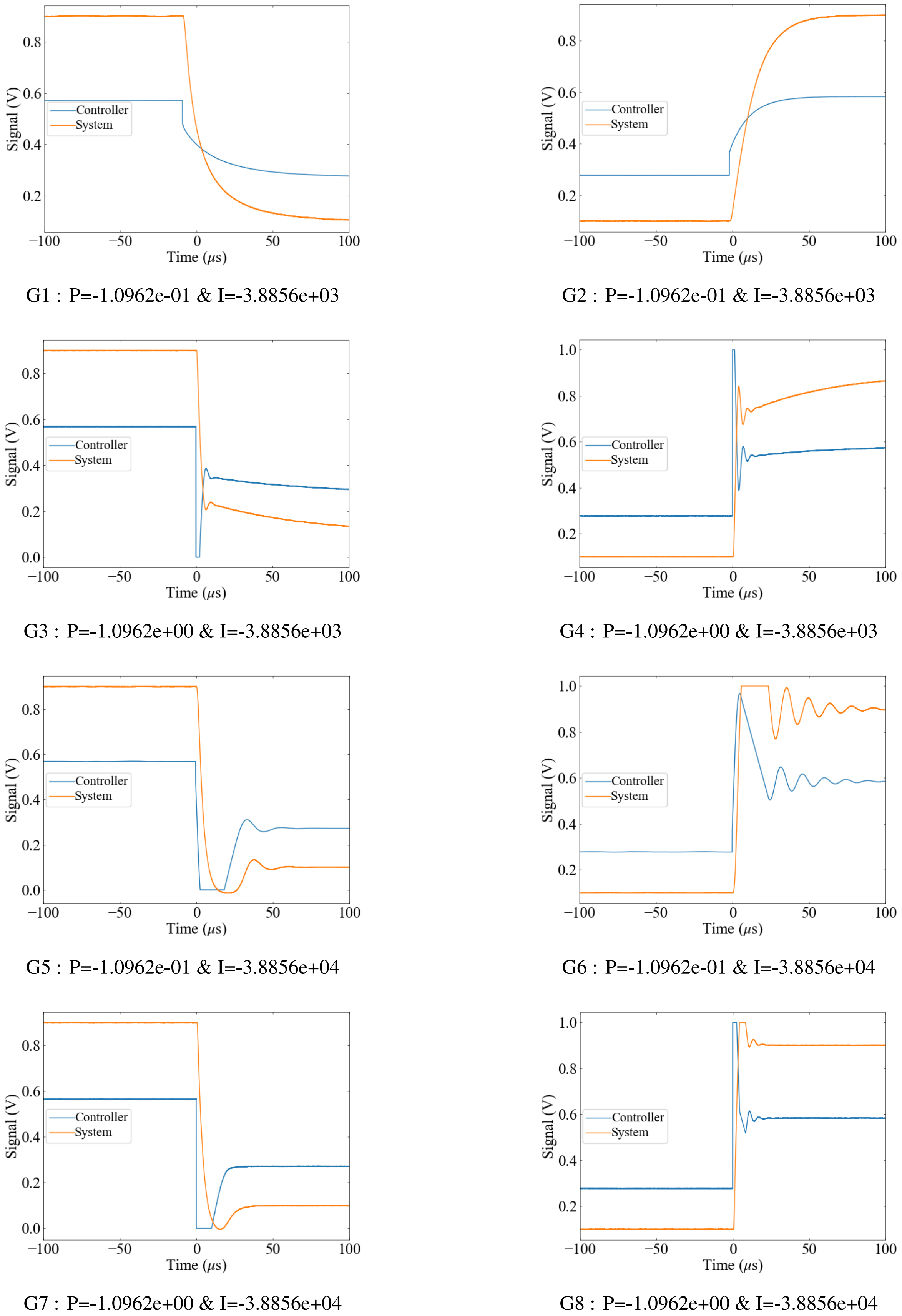


Figure 4.7: Controller (RedPitaya) and System response for two settings with different P parameters. For the right measurement the P parameter was decreased to reduce overshoots

In Figure 4.8 the controller/system responses for different P and I settings are shown. Here the set value was changed from 0.9 to 0.1 V and vice versa to put high stress on the controller. In the first row the lowest P and I settings (in absolute units) are chosen and one can see that the set value can be reached but the change of the system state is very slow. Increasing the parameters will increase the response strengths but unwanted effects can appear. If one parameter is too high in comparison to the other one the lower one is not able to compensate misleading behavior of the strongest one. This can be seen in the graphs G3 - G6. However, there are values where the parameters balance out each other leading to a more stable behavior. The magnitude of both values then determines the total strength of the controller and hence the speed of correction. The best values of the system in this thesis are shown in G7 and G8 where the signal can be stabilized in roughly 25 μs. It is worth to notice that this result was limited due to practical implementations. First of all one can see that the controller voltage sometimes saturates because the maximum RedPitaya analog output is 1 V limiting the overall manipulation strength of the controller. Secondly the I parameter value was lower bound to -3.8856e+04 by the PyRPL software. It is imaginable that with even lower values the controller response could be further optimized.



G1 : P=-1.0962e-01 & I=-3.8856e+03

G2 : P=-1.0962e-01 & I=-3.8856e+03

G3 : P=-1.0962e+00 & I=-3.8856e+03

G4 : P=-1.0962e+00 & I=-3.8856e+03

G5 : P=-1.0962e-01 & I=-3.8856e+04

G6 : P=-1.0962e-01 & I=-3.8856e+04

G7 : P=-1.0962e+00 & I=-3.8856e+04

G8 : P=-1.0962e+00 & I=-3.8856e+04

Figure 4.8: Controller/System response for different P and I settings when set value is changed from 0.9 to 0.1 V (left column) and vice versa (right column)

So far only the local response of intensity stabilization due to induced set value changes were shown. The main goal is to have a stable signal over long time. To see this the output was monitored on a separate oscilloscope to record a high resolution longtime measurement both for active and inactive controller. In the latter case only a DC signal was applied to the AOM to observe the bare intensity fluctuations. The results are shown in Figure 4.9. One can see that without stabilization the signal is heavily fluctuating including overall jumps. The controller mechanism diminishes the fluctuations and no jumps appear. It is also apparent why the time optimization of the controller response is important. Without optimization the controller might be able to damp the long term drift but not the fast fluctuations underlying. For quantifying the degree of stabilization the following formula can be useful:

$$\eta = \frac{S_{\text{max}} - S_{\text{min}}}{S_{\text{mean}}} \tag{4.1}$$

Here S_{max} and S_{min} are the maximum/minimum signal value respectively whereas S_{mean} is the mean of the whole signal time trace. Thus we set the signal width in contrast to the total offset. Applying the formula to the recoreded data we get the following values:

	not stabilized	stabilized
η	0.169	0.006

The controller is able to stabilize the signal from 16.9% down to 0.6%. This is an important result because not only the stability of tweezer light depends on the initial laser stability but also calculated deflection efficiencies for further characterizations (see later in section 4.2) will use an initial laser intensity measurement for reference. Hence the reliability of this signal must be guaranteed.

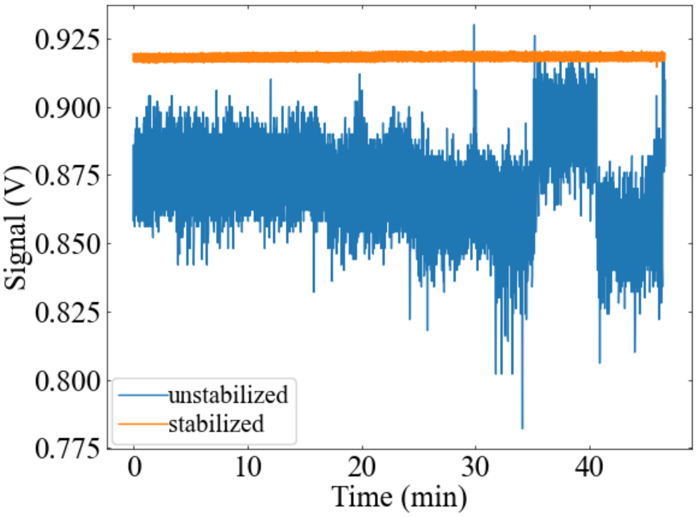


Figure 4.9: Long term laser signal on controller PD both for stabilized and not stabilzed case

4.1.3 AOD

The central part of the optical setup is the crossed AOD configuration. A picture of the device is shown in Figure 4.10 on the left side. The RF signals to drive the crystal as described in section 2 are applied via SMA connections. The AODs are attached on a flat metal plate which is screwed on a kinematic mount which allows to turn the configuration both horizontally and vertically. With these two degrees of freedom it is possible to align the AODs for maximum deflection efficiency which was done in the following way: Firstly only the AOD for y-deflection gets a RF signal at the central frequency of 104.5 MHz and the mounting is aligned until the maximum possible intensity for the first diffraction order is reached. This then provides a good starting point for turning on the AOD for x-deflection because the final diffraction order already gets some energy of the former one and it is easier to detect a signal. Finally the fully deflected 1,1 order beam is aligned until it reaches the maximum intensity.

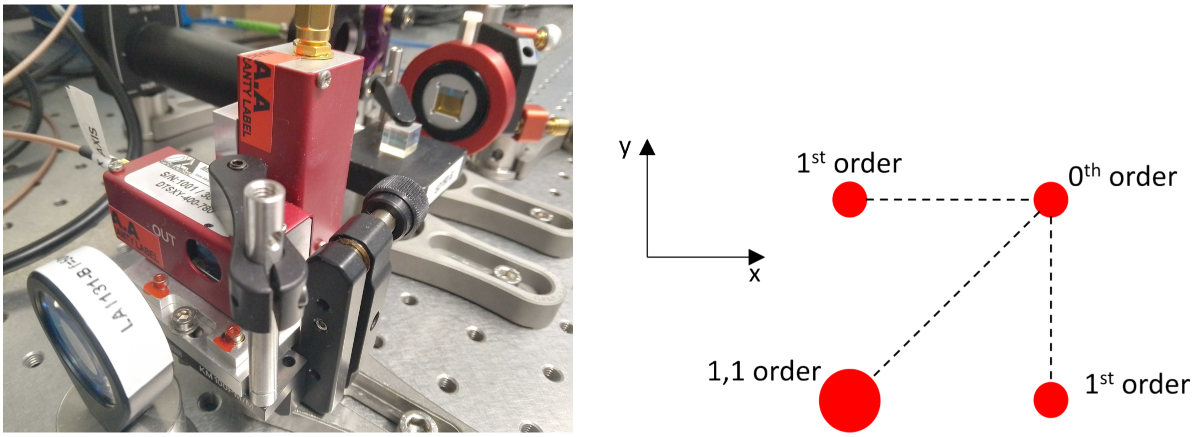


Figure 4.10: Left: Picture of the double AOD configuration screwed on a kinematic mount, the RF signals are provided via SMA connections; Right: Sketch of AOD deflection orders

Note that there are four beams coming out of the second AOD in total. They are the

- 0th order beam which is the part of the incident laser light which has not been deflected
- 1st order beam in y-direction which is the first deflection produced by the first AOD and which has not been deflected by the second AOD
- 1st order beam in x-direction which is the deflection of the 0th order beam going through the second AOD
- 1,1st order beam which is produced by both deflection in x- and y- direction

This is also illustrated in Figure 4.10 in the right sketch. Since the first three beams are arranged at the edges of the deflection window they can be easily blocked with an iris.
So far the processing of electronics and optics for producing stable deflections were discussed. In the next chapter the detection of the beams and their intensity control will be explained.

4.2 Deflection efficiency

A relevant part of generating tweezer patterns is not only frequency management but also adjustment of amplitudes. This is important because the deflection efficiency depends on frequency and RF power. This section discusses the frequency dependence of the double AOD system and how to compensate for frequency depended intensity variations. To this end we first look at the natural AODs behavior when running it with maximum RF power for a scan of frequencies. Because there are two frequency options (one for each AOD) a 2D heat map can be created where the deflection efficiency is color coded. We can define the efficiency by the following equations:

$$\eta = \frac{P_{\text{def}}}{P_0} = \alpha \cdot \frac{V_{\text{osci}}}{P_0}$$
$$\alpha = \frac{P_{\text{ref}}}{V_{\text{ref}}}$$

Here P_{def} and P_0 are the deflected power and total power after AODs without RF signal respectively. The latter one can be measured beforehand with a power meter. Now it also becomes clear why we paid much attention on the intensity stabilization because we would like to have a reliable quantity to reference the deflected power to. Because the deflection power is measured with a PD and oscilloscope readout, we need to get a relation between oscilloscope voltage V_{osci} and power. This is done by introducing the proportionality constant α which is calculated beforehand at an arbitrary frequency setting by measuring the oscilloscope signal V_{ref} and the corresponding deflected laser power P_{ref}. Dividing these values gives a reference such that any other measured oscilloscope voltage can be converted into power.
Applying these equations to the above mentioned 2D-scan yields the result shown in Figure 4.11:

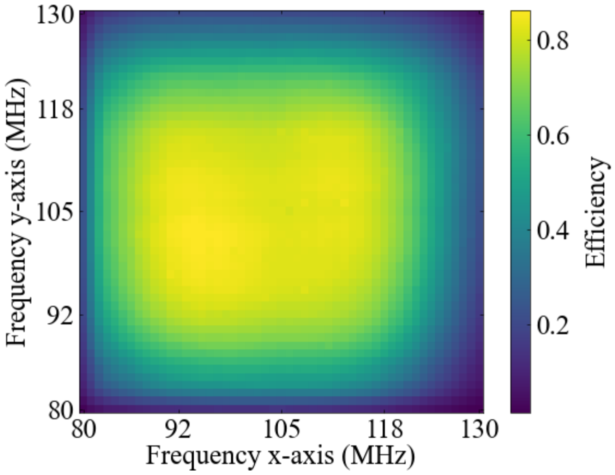


Figure 4.11: Efficiency map of the crossed AOD system with RF power close to 2 W

One can see that over some range the system produces a roughly homogeneous deflection output with efficiencies over 80%.
When operating on this space one is in general interested to have tweezers with controlled efficiency,

ideally the same. This can be done by utilizing the free range of RF amplitude. Subsequently two methods will be discussed how this could be achieved.

4.2.1 Feed-forward efficiency optimization

A straightforward method is to measure a stack of efficiency values for different amplitudes and frequencies. Then one can interpolate between these values to get an estimation of the right amplitude for a desired efficiency. Here a linear interpolation method implemented by the *RegularGridInterpolator* from SciPy [28] is used which needs a regular data grid. Here a cubic grid based on efficiency measurements for different x-, y- frequencies and amplitudes was measured. The goal is to predict an efficiency value for an arbitrary frequency and amplitude combination. To this end the algorithm uses triliniear interpolation. This method approximates the target value based on cubic grid points which surround the target point by using linear interpolation. For a formula the idea is to multiply the values at the grid points by factors such that if the approximation function is evaluated at such a grid coordinate only the prefactor for the target value remains as 1, the rest vanishes. In between there should be then a linear combination. This can be realized by using the following formula (for 3D):

$$\begin{aligned} f(x,y,z) =&(1-a)(1-b)(1-c)f(x_0,y_0,z_0) + a(1-b)(1-c)f(x_1,y_0,z_0)+\\ &(1-a)b(1-c)f(x_0,y_1,z_0) + ab(1-c)f(x_1,y_1,z_0)+\\ &(1-a)(1-b)cf(x_0,y_0,z_1) + a(1-b)cf(x_1,y_0,z_1)+\\ &(1-a)bcf(x_0,y_1,z_1) + abcf(x_1,y_1,z_1) \end{aligned}$$

Here x_i, y_j, z_k with $i, j, k \in \{0, 1\}$ are the corner grid points of the grid cube surrounding the target point and $f(x_i, y_j, z_k)$ the known value at the grid points. The evaluation restriction mentioned above then gives the following values for the prefactors:

$$\begin{aligned} a &= \frac{x-x_0}{x_1-x_0}\\ b &= \frac{y-y_0}{y_1-y_0}\\ c &= \frac{z-z_0}{z_1-z_0} \end{aligned}$$

In order to predict the right amplitude at a certain frequency combination the interpolator is asked to give efficiency estimations for a close space of amplitudes on this frequency combination. Then the best amplitude can be chosen.

It is important to note that the range of reachable frequencies has two limitations. Firstly there will be certain frequencies at the edges of the shown efficiency map that can not reach the desired value at all, because their values are too low even at maximum applied amplitude. Secondly a limitation is given by the lower and upper boundary of the amplitude measurement space. On the lower side we have the highest efficiency value for the lowest recorded amplitude as the bottom boundary. On the upper side we have the lowest efficiency value for the highest recorded amplitude as the top boundary. Thus one needs to choose a desired efficiency between these two limitations for the best performance on the whole frequency space.

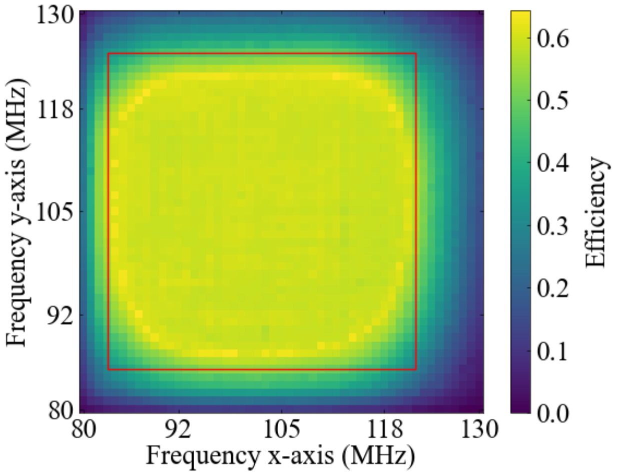


Figure 4.12: Manipulated efficiency map by trilinear interpolation based on lookup-table, the red rectangle marks the performance evaluation region

In Figure 4.12 a goal efficiency of 60 % was chosen for the whole efficiency map. As a measure for the performance, one can compute the mean absolute deviation in percent from the efficiency goal in a defined evaluation window since it does not make sense to include frequency points at the very edge. Here we choose $83 \le f_x \le 122$ and $85 \le f_x \le 125$ shown as the red rectangle in the plot. The absolute deviation mean for this window is $4.55\% \pm 0.04\%$. It is clear that the accuracy of this approach depends on the density of measurement points of the lookup table as well as the amount of measurement points since it is possible that the system behavior changes slightly due to environmental influences while measuring.

4.2.2 Iterative efficiency optimization

A method that does not need any prerecorded measurements is iterative amplitude adjustment in real time. To this end we start with an arbitrary amplitude for each frequency tuple. Here for simplicity every amplitude begins at maximum value and the efficiencies for the whole frequency map is measured. The deviations of these values from the target efficiency is computed. Then the amplitudes are adjusted (upwards or downwards) based on the efficiency deviations multiplied by a weight, here called α. The advantage of this method is that it can adjust the amplitudes in real time as long as needed. The concept is depicted in Figure 4.13 below. Basically it is the same approach as the P part for the PID controller. It can be expected that the system should converge to the desired efficiencies since the efficiency deviation $\Delta\eta$ becomes smaller iteratively.

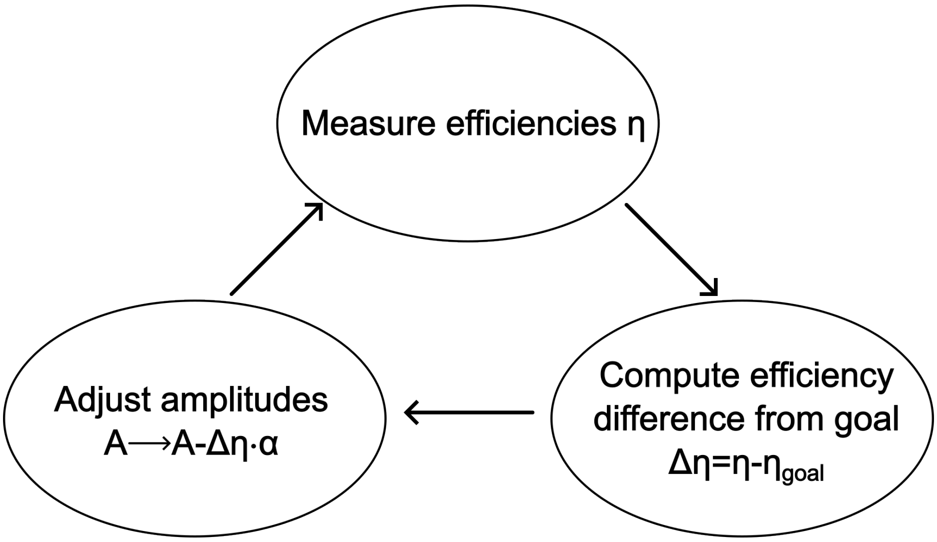


Figure 4.13: Sketch of iterative efficiency optimization by amplitude adjustment in real time

The key point of this method is a good choice of the adjustment weight α. There are two regimes we would like to avoid:

- α is too small: here the strength of amplitude adjustment is damped by α. The adjustment over time will become very slow which could then make the system unable to properly react to environment changes or just be too slow for experimental time scales
- α is too large: here the change in amplitude will become so strong that the efficiency change overshoots the goal. Then it is possible that it needs long time for the system to converge or the goal will never be reached

The practical effect of these point is depicted in Figure 4.14. Here the efficiency map evolution for an efficiency goal of 50 % is shown at three stages of iteration.
There is no general solution which value to pick for α because it depends both on electronic and optic components used in the system. However, it is possible to search for a reasonable value beforehand. For the values shown here a range of values for α were tested and for illustrative purposes three scenarios are presented. In the first row (α = 400), where a small α was chosen, it is apparent that

there is still a large location which is far apart from the goal. In the second row ($\alpha = 3000$) one sees effectively an inversion of the efficiency map which comes from the overshoot. In the third row ($\alpha = 2000$) the evolution with a weight between the two extremes is shown. In the last example the evolution does not jump and converges quickly. This is also shown in Figure 4.15. Here again the performance is quantified by the mean of the absolute efficiency deviation from the goal Δd in the frequency window $83 \leq f_x \leq 122$ and $85 \leq f_x \leq 125$. In contrast to the former feed-forward method the evaluation point could be arbitrarily chosen. The values after 19 iterations are:

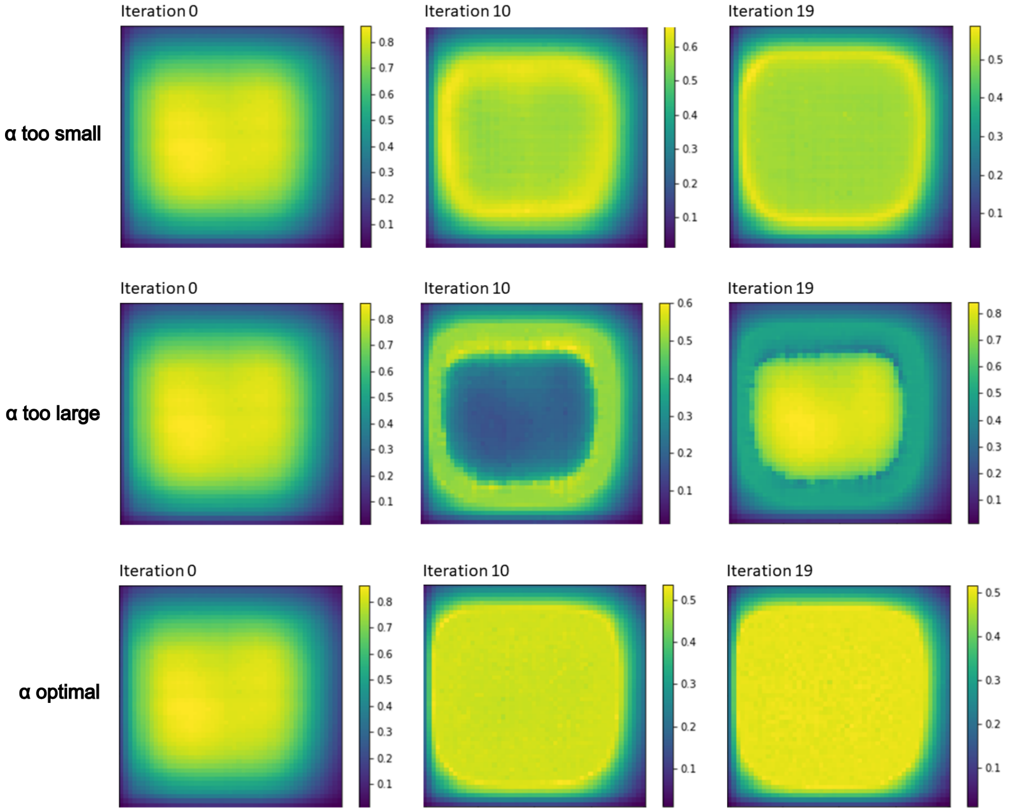


Figure 4.14: Example efficiency evolution for small (first row) and large (second row) α regime; third row shows optimal evolution

α	Mean absolute efficiency deviation (%)
400	4.01 ± 0.04
2000	1.15 ± 0.05
3000	31.44 ± 0.04

We can see that the performance of the iterative method is better than the feed-forward one. This has the following reasons: firstly it does not assume any specific mathematical form of the efficiency landscape other than a global minimum for $\Delta\eta$. The feed-forward method does it by taking linear approximation and the accuracy also depends on the distance of measurement values. Secondly this method automatically adapts to potential fluctuations while operating. In contrast the former method takes false assumption if the system changes due to environmental influences during recording the lookup table. Another advantage of this approach is that it is not limited which efficiencies to reach due to measurement points. We can conclude that this method is more reliable to reach its goal, however it is not capable to give predictions beforehand for short timescale experiments since it stabilizes on the run. It is also important to note that the actual time of convergence depends on the number of tweezers and the measurement speed of the experimental system.

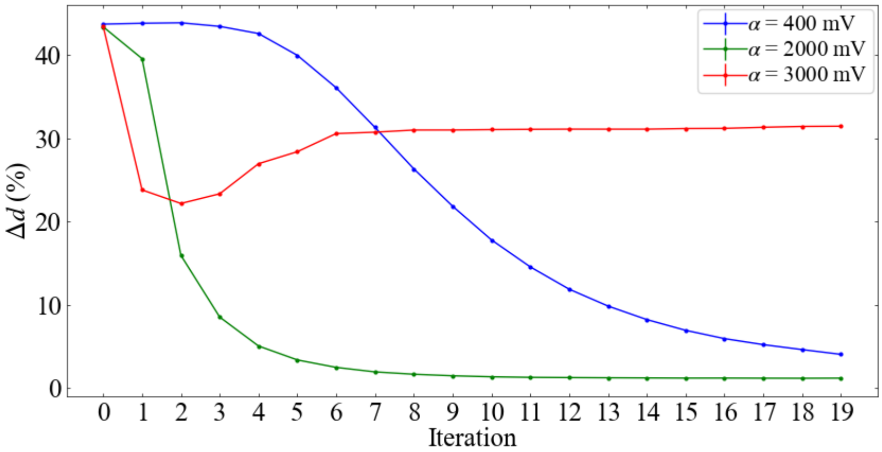


Figure 4.15: Trace of efficiency evolution performance Δd for three α configurations: when α is too low (blue) the convergence is slow, when α is too high (red) the zero line can to be reached, in between there is an optimal region for α (green) such that convergence to the zero line happens in a short amount of iterations

An additional important point to consider is that here the method shows how to reach tweezers with equal intensity. Later the trapping of atom clouds in the tweezers are demonstrated and then the atomic background needs to be taken into account. This was outside the scope of this thesis but for practical implementation the mathematical mechanism is the same.

4.2.3 ArduCam

In order to see the spatial distribution of generated tweezers in the test setup a MT9J001 ArduCam was used. It is a monochromatic CMOS camera that can be interfaced with python to read out pixel values proportional to the impinging intensity. A picture of the device is shown in Figure 4.16. One can see the colorful CMOS chip in the middle. The ArduCam is embedded into a lens tube for practical mounting on the optics table. In addition it is possible to attach optical filters on it to protect the chip and ovoid saturation when laser light falls onto it. The full camera actually consists of two stages: camera chip (upside) and breakout board (downside). Both are connected with a ribbon cable which is not visible in the picture. The USB connection for data streaming is provided by the breakout board.

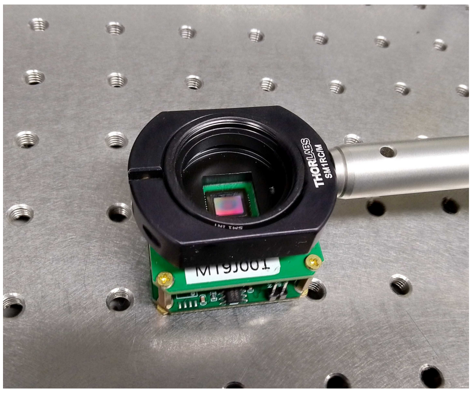
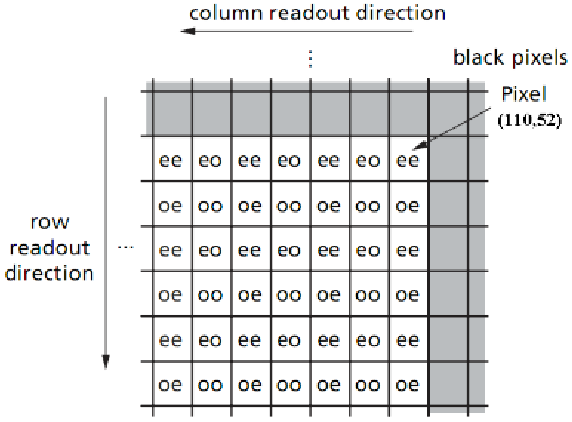


Figure 4.16: Photo of the ArduCam surrounded by a lens tube

Unfortunately the sensor chip is not fully monochromatic, a Bayer monochrome pattern is implemented onto it. A scheme of this pattern is shown in 4.17.



ee	eo	ee	eo	ee	eo	ee
oe	oo	oe	oo	oe	oo	oe
ee	eo	ee	eo	ee	eo	ee
oe	oo	oe	oo	oe	oo	oe
ee	eo	ee	eo	ee	eo	ee
oe	oo	oe	oo	oe	oo	oe

Figure 4.17: Bayer monochrome pattern scheme of ArduCam MT9J001 [29]

The different ee, eo, oe, oo pixels will lead to different values due to wavelength dependence. However, it is possible to reduce this effect because one has access to control registers. With these registers one can for example change the pixels gains such that the input of the patterned pixels are differently weighted such that the output gets effectively balanced. Software routines to find the right register values were developed by Samuel Germer and his results for the settings are used in this thesis [30]:

gain register in decimal code	gain value in decimal code
12374	4162
12376	4176
12378	4173
12380	4161

The effect of this register change can bee seen in Figure 4.18. A picture of a Gaussian laser beam was taken with the ArduCam without/with register value change according to the table above. On the left one can clearly see the checkerboard pattern induced by the Bayer filter. The projections of the data both for horizontal and vertical axis show that the Gaussian shape is clearly distorted. On the right it can be seen that with register changes the Bayer pattern can be diminished.

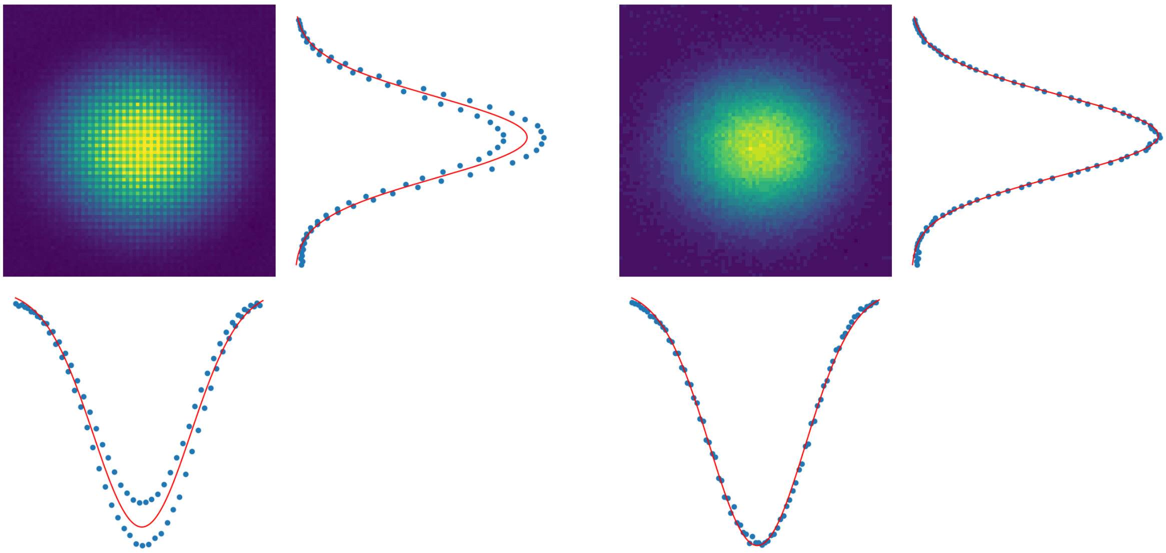


Figure 4.18: Data of deflected Gaussian beam recorded with ArduCam: left without register change; right with register change; the horizontal and vertical pixel sums (projections) are fitted with a Gaussian function

At this point it is possible to reliably monitor the deflection position over time. This is shown in Figure 4.19. Here the location of a deflected beam on the camera was traced over time. It is apparent that the beam moves over some pixels in both horizontal and vertical direction for more than one hour until it roughly converges to a final spot which also effects the deflection efficiency. This is a temperature effect and will be discussed in section 4.2.4. As it is apparent in the pixel projections in Figure 4.18, the register change does not cancel the pixel pattern fully and in addition interferences still can occur due to a camera protection layer. This is the reason why a PD in combination with an oscilloscope was used for this kind of measurement since the deflected beams are all focused on the same spot of the PD. It turned out that the efficiency values are more reliable then. However, the ArduCam is still a useful tool for spatial depended measurements.

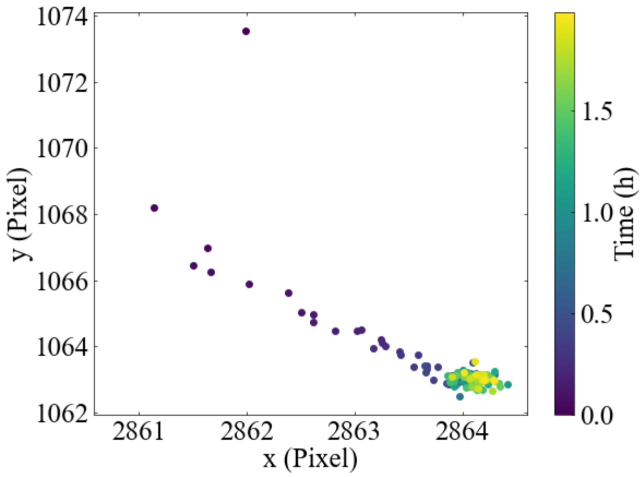


Figure 4.19: Deflection spot trace recorded with ArduCam; the spots are the centers from the fits of the deflected Gaussian laser beams

4.2.4 Temperature dependency

As already mentioned previously, the AODs have a non-constant efficiency behavior over time after turn-on. To investigate this, an array of points of the efficiency map were selected. Their deflection efficiency was then measured for multiple hours with the PD. This is shown in the plot below. The measurement was started with cold AODs at laboratory temperature. On the left side the selected frequency combinations for the longtime measurements are depicted. On the right side the corresponding efficiency evolution is shown. One can see that there is an oscillatory drift in efficiency for the first half hour. The amount of oscillation and total change of drift is different for the frequency settings. After thirty minutes to one hour a stationary stage is reached.

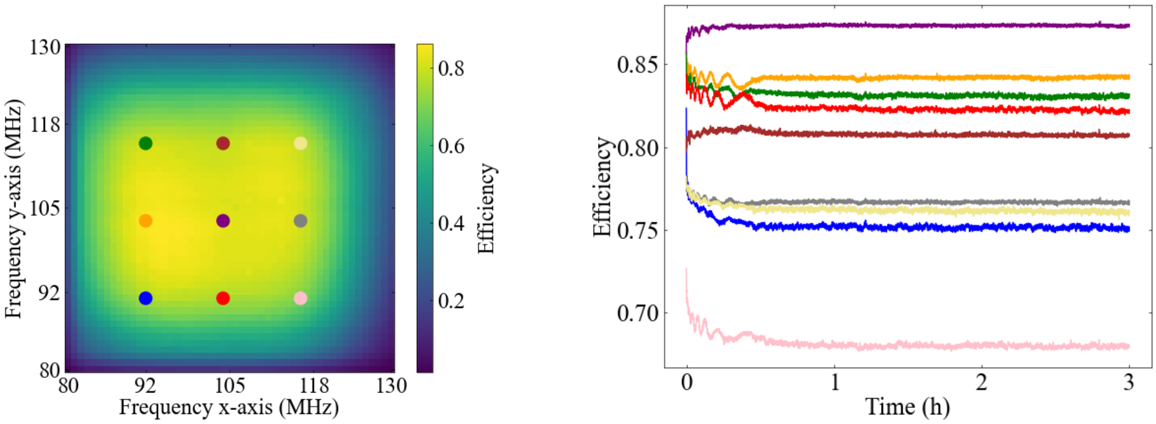


Figure 4.20: Left: selected points from efficiency map for longtime efficiency measurement
Right: longtime efficiency measurement for selected frequency combinations

The reason for these drifts is the temperature change of the AODs (see Figure 4.21). Applying RF signals to the AODs heat the crystals up and they expand. Effectively the end sides of the crystals provide reflective surfaces and interference drifts can be seen while expanding. As soon as the temperature change stops the efficiency drift also ends. This effect also explains the beam location change on the Arducam shown in section 4.2.3. The deflection angle is altered by the changing crystal geometry. In the picture the curves saturate at different temperature levels because the temperature sensors were placed on different spots of the AODs because of access restrictions.

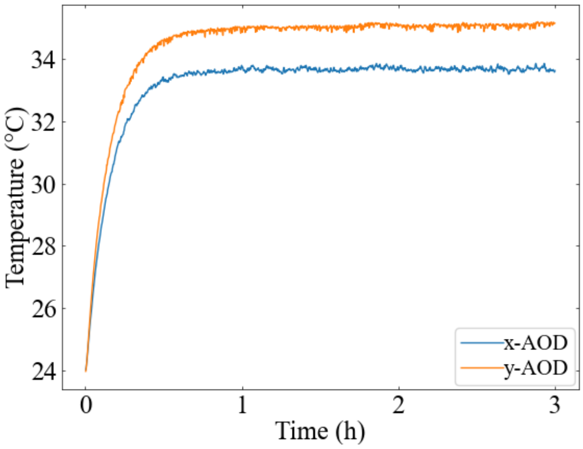


Figure 4.21: Temperature trace at AOD surfaces while operating

Thus for a reliable application the AODs need to be held in a stationary state e.g. applying a low efficiency RF signal while being unused to hold the temperature.

CHAPTER 5

Implementation into main experiment

The previous sections discussed generation and control of optical tweezers. So far the tweezers were investigated as distributed light fields detected with camera or PD. The next part is to check the functionality of the SI card/software system in the main experiment. In this chapter firstly the experiment is discussed in the detail that is relevant for this project. Then the technical infrastructure and the implementation of the new SI card/software system will be explained. Currently the experiment uses a single AOD and the second part of this chapter shows the successful generation of arbitrary 1D tweezer patterns.

5.1 Overview

The experiment uses ultracold rubidium atoms in Rydberg states for quantum optics. The strong interaction between Rydberg atoms can be connected to photons flying through the atom cloud and for example control the photon statistics as mentioned in the introduction. For this multiple atom clouds can be placed with optical tweezers in a 1D array. To this end the RQO setup releases rubidium atoms from heated dispensers in a vacuum chamber and then cools and traps them. The steps to achieve this are the following:

1. Firstly a magneto-optical trap (MOT) is loaded with the atoms for 1.3 s. Then the MOT is compressed by increasing the magnetic field gradient and the detuning of the cooler laser. In parallel the intensity of cooler and repumper laser beams is reduced. By this the atoms are transported into a crossed optical dipole trap. This process takes 100 ms.

2. The atoms in the optical dipole trap are further cooled with a combination of evaporative and Raman sideband cooling. This is discussed in Ref. [31]. This procedure needs 40 ms. At the end the atoms have a temperature on the order of 4 μK [31].

3. The trapping geometry is then altered by an additional trapping beam (here called magic beam) along the long axis of the crossed dipole trap. The trap provides a magic wavelength for trapping both the ground state and the relevant Rydberg state of rubidium.

4. At the end of cooling the optical dipole trap is turned off and the tweezer laser beam can be activated to localize the atoms at well-defined spots.

A sketch of the final beam configuration is shown in Figure 5.1. The tweezer beams are created with a one-dimensional AOD offering deflection in the y-direction. The z-axis of the tweezer beam is fixed. To analyze the trapped atoms absorption imaging is used. With this technique it is possible to measure for example the number of atoms in a cloud or the atoms temperature. The atoms are imaged along the z-axis in Figure 5.1.

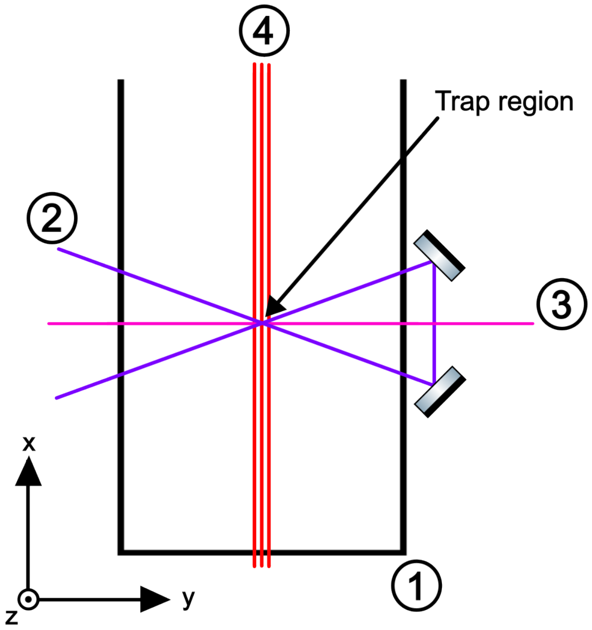


Figure 5.1: Sketch of laser geometry inside the vacuum chamber of the main experiment (the MOT beams are not shown)
1. Vacuum chamber; 2. Crossed dipole trap
3. Magic beam; 4. Tweezer beams

For quantum optics experiments photon probe pulses go through the atom clouds and the Rydberg states are applied with a control laser beam. Both beams overlap with the magic beam.

5.2 Digital infrastructure

As mentioned before the old system used three DDS boards as RF control for the tweezers. These are now replaced with the single SI AWG. Physically this is simply done by connecting the SMA ports of AOD and SI card. The next step is to embed the PC with SI card into the experimental control system. To this end consider Figure 5.2 which shows the digital communication infrastructure of the main experiment. For easier understanding not all instances are fully shown in detail. In general the infrastructure consists of computers (blue), electronic devices (gray) for measuring or signal generation and software (green) that enables communication with other instances. The central part of the experiment is the CONTROL PC which controls the action of the other devices through the script *cycle.py*. This cycle script sends pings to other machines in a defined order. Two other important instances are VAULT PC and CAMERA PC. The former one hosts a SQL server which provides numeric variables that are frequently used in the experiment. In addition on VAULT PC measurements are saved. CAMERA PC runs the camera which measures the photon absorption.

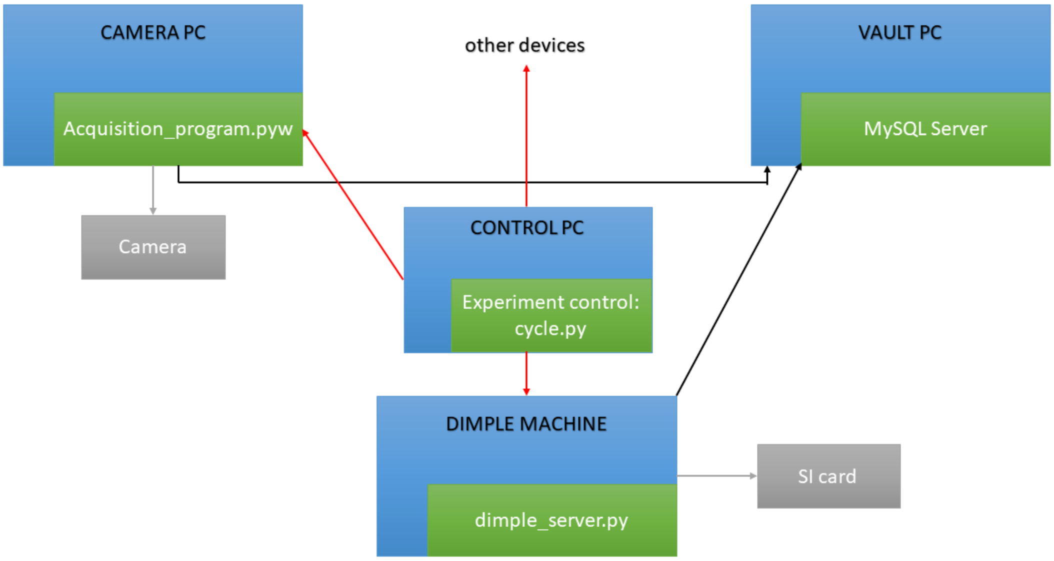


Figure 5.2: Communication scheme of the main experiment
Computers (blue) run scripts (green) to control laboratory devices (gray), the scripts run when receiving a ping from the CONTROL PC

The new device is the computer DIMPLE MACHINE which runs the SI card. The *dimple_server.py* script waits for a ping from the control. If activated it reads sequence data from the vault, computes the RF data and starts the card which actually starts the RF sequence when triggered from control. The sequence data contains the time, frequency and amplitude properties of the tweezer signals at user defined timestamps. As mentioned earlier in this thesis the data between these timestamps is then generated in a linear manner. The data is saved in a txt-file and is build up as follows: the first three lines consists of one matrix each which specify

1. timestamps (ms) at which the the tweezers should have certain properties. Each matrix row represents one tweezer and the numerical values are the time values in milliseconds
2. frequencies (MHz) of each tweezer that correspond to the timestamps
3. amplitudes (mV) of each tweezer that correspond to the timestamps

These are the data for the y-axis. The amplitudes are scaled if the final generated RF data exceeds a predefined allowed value. The z-axis has fixed values of frequency and amplitude for the whole sequence which are given in the next two lines. The last value in the txt-file is the trigger level of the SI card in mV. An example file could look like the data in Figure 5.3 on the left side which is actually the sequence data which generated the output presented in Figure 3.12 which is shown here again.

With this method the tweezer configuration can be simply changed be rewriting the data in the txt-file and the tweezers are then changed accordingly in the next cycle. In addition to explicit user input the software is able to read values from the vault database and include them into the sequence data. With this different sequence configurations can be tested automatically.

```
[[0, 10, 20, 30, 60], [0, 10, 20, 30, 60], [0, 10, 40, 50, 60], [0, 30, 60]]
[[79, 79, 78.5, 78, 75], [80, 80, 80, 80, 80], [81, 81, 81, 81, 81], [83, 83, 86]]
[[200, 500, 600, 600, 760], [0, 0, 300, 760, 760], [0, 0, 300, 760, 760], [600, 600, 760]]
80
2450
100
```

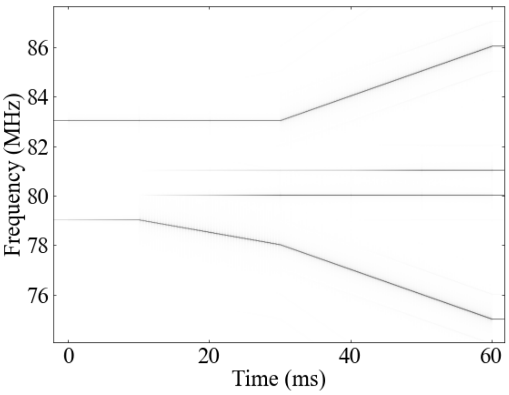


Figure 5.3: The sequence data for every tweezer on the left creates the Fourier signal data on the right. The first three matrices describe timestamps (ms), frequencies (MHz) and amplitudes (mV) of channel one, every entry corresponds to one tweezer each. The last three lines specify amplitude, frequency for channel two and trigger level.

5.3 Trapped atom clouds

The first part of the verification process was to check that the new RF generator can produce three tweezers as possible with the old system. Different configurations were tested and four absorption images are shown in Figure 5.4. The atom density is color coded in arbitrary units. The images are not single shot pictures, each configuration is an average of several experimental iterations. This is needed because the loading result is not the same for every shot which is caused by the complex trapping architecture inside the chamber and atom movement. The whole sequence until the tweezers are in their final positions takes 15 ms.

The first picture shown in Figure 5.4 is the bare atom cloud trapped in the crossed optical dipole trap from which the actual tweezers catch their atoms. The next two images show the three tweezers with equal distances but in the second configuration the RF amplitude for the third tweezer was higher than the other ones. By doing this the third tweezer gets more diffraction light intensity and hence the trap depth is deeper and the tweezer collects more atoms. The third configuration shows the three tweezers with unequal distances.

The next part verifies that the system can go beyond the old one by generating more tweezers. This is shown in Figure 5.5. Different configurations are shown from four to seven tweezers. It is important to note that the more tweezers are created the less power they receive each from the RF source because the total power is distributed over more tweezers. Therefore during the experiment the tweezer laser intensity was increased to make the traps as deep as for the few tweezer configuration. From picture three to four the third tweezer was split into two close ones as indicated with the red arrows. It is apparent that for some tweezers atoms leak out of the trap spots leading to vertical strips going through the tweezers in the absorption images. The reason for this is the finite temperature of the atoms where the hottest ones can overcome the trap potential and align then themselves along the Gaussian tweezer beam. However, this is not a problem for the experiment since probe and control beam come from the perpendicular direction and only have to apply on the atoms in the tweezer centers.

Figure 5.4 and 5.5 show the proof of principle of the new tweezer generation to trap atoms. Further optimization like balancing atom numbers in the tweezers is necessary, but this lies beyond the time-frame of this thesis.

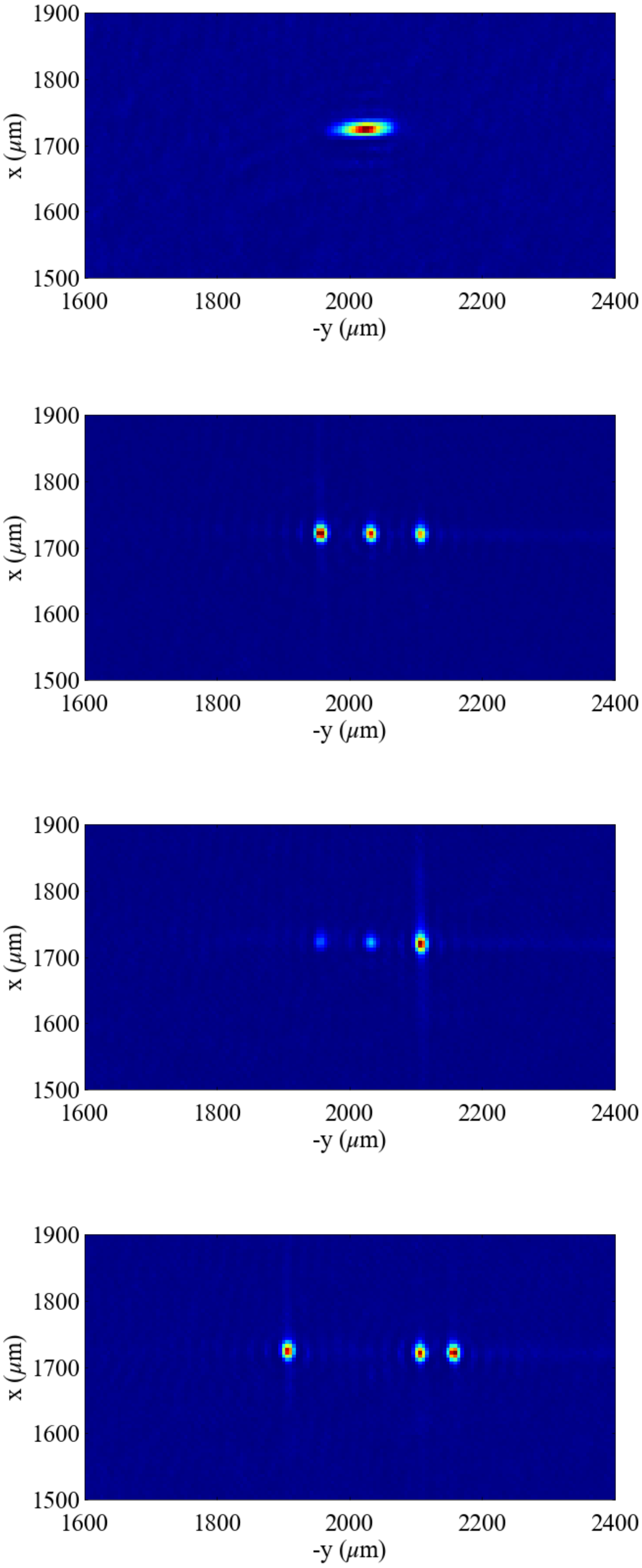


Figure 5.4: Absorption images of atoms in the crossed dipole trap and of different three-tweezer configurations

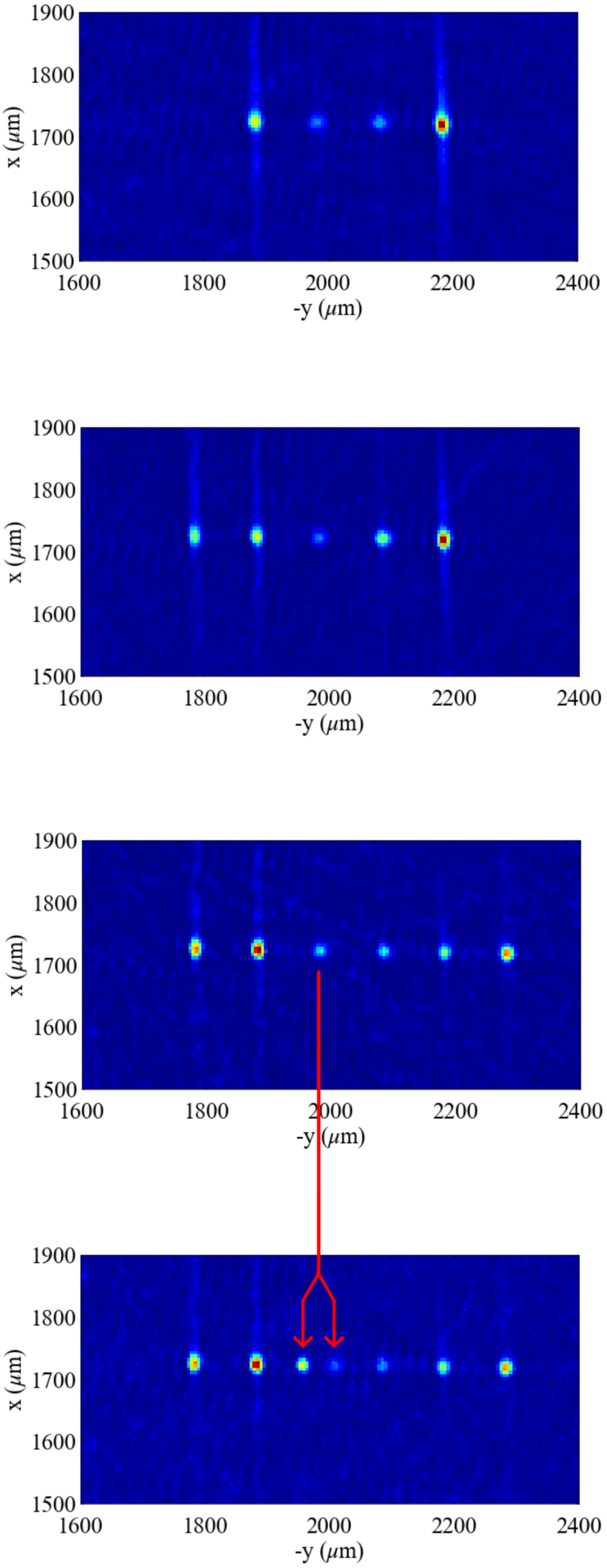


Figure 5.5: Absorption images for different amount of tweezers that go beyond the old setup, the red arrows represent how a single tweezer can be split up by shifting the corresponding frequencies up and down

CHAPTER 6

Summary & Outlook

In this thesis the properties of a crossed two-AOD system for generating tweezer patterns were investigated. The aim was to create routines for generating RF patterns in order to enable an AOD to produce arbitrary tweezer patterns. Furthermore the goal was to implement this into the RQO main experiment to create larger 1D chains for new Rydberg superatom experiments.

Firstly the theory regarding acousto-optical modulation/deflection and optical tweezers were discussed. After that a Spectrum Instrumentation AWG card was characterized for generating RF signals in the MHz regime in combination with the RF signal computation with a GPU. An optical test setup was build and characterized to present the practical properties of a two-AOD system. On this way a successful implementation of a RedPitaya device for laser intensity stabilization was shown. Furthermore two methods for controlling the deflection efficiency by software controlled RF adaption based on intensity detection feedback were demonstrated. Regarding the practical implementation of the SI card into the RQO main experiment a software infrastructure for generating arbitrary tweezer patterns was build. In case for 1D sequences a socket server script was written to embed the RF tweezer signal generator into the main experiment cycle. Lastly the successful tweezer generation was documented by absorption imaging for different tweezer patterns that extend the old amount of possible tweezers and their behavior in time and space.

So far the two-AOD system was only tested in a separate setup for this thesis. At some point in the future this device or a similar one could replace the 1D AOD system used in the main experiment currently. In combination the 2D tweezer generation with the prepared software can then be used. A further outlook considers a feedback mechanism for the main experiment. So far the tweezer generation is based on user fixed values and hence the tweezer powers is as well. In order to be able to balance the atom numbers for the tweezers a feedback infrastructure could be included. Here the feedback algorithms shown in this thesis provide a starting point which could also be utilized for another aspect: as soon as 2D patterns are generated a reliable atom loading becomes more difficult since tweezers can then cover each other. Here the full freedom of RF sequence arrangement can be utilized for testing and optimizing the tweezer loading.

In total the new tweezer generation system extends the old one by generating arbitrary RF signals instead of mixing fixed frequencies and promises to be a versatile tool for new Rydberg atom experiments. For example the new 1D capabilities promise to provide an extended infrastructure for creation of nonclassical states of light by utilizing the trapped atom clouds as chained quantum emitters [32]. Furthermore the 2D tweezer generation will enable the main experiment to follow other nonlinear optics experiments [33] to study Rydberg interactions in different spatial configurations.

APPENDIX A

Python code for generating two sinus signals with SI card

Here the code for generating two independent sinus signals with arbitrary frequencies (based on the limitations discussed in chapter 3) and amplitudes is shown to illustrate the control of the SI card. The control software for the SI card is embedded in a class and the method for generating two sinus signals 'two_sinus' uses other functions not shown here explicitly.

Different register address values or data values are predefined by SI. They are written in capital letters. Important functions predefined by Spectrum Instrumentation for interfacing the SI card are used here [22]:

The **spcm_dwSetParam** functions for 32 or 64 bit transfer are used to write values into the register memory of the SI card. It holds three arguments: the handle to the card (here called self.hCard), the numeric value of the register address, the value to write in.

The **pvAllocMemPageAligned** function is used to declare a memory buffer with certain size.

The **spcm_dwDefTransfer** defines a buffer with data that can then be uploaded subsequently. It holds the following arguments: handle to the SI card, type of buffer, transfer direction, number of bytes after which an event is sent, pointer to the data buffer, offset for transfer in board memory, buffer length.

```python
def sinus_samples(self, frequency, phase=0): #Generate sinus data for Spectrum card, frequency in MHz
  if frequency > 130:
    print("Frequencies about 130 MHz forbidden. Process stopped.")
    sys.exit()
  nu = frequency / self.f_cen
  return 32000 * np.sin(2*np.pi*self.N_loop*self.samples*nu / self.L_min + phase)

def two_sinus(self, ampl0, frequ0, ampl1, frequ1): #Compute sinus for both channels and run with SI
  if ampl0>self.max_allowed_ampl or ampl1>self.max_allowed_ampl:
    print("Amplitude above " +str(self.max_allowed_ampl)+ "mV forbidden")
    return 0
  self.stop_output()
  amplitude0 = int32(ampl0)
```

```python
amplitude1 = int32(ampl1)

frequency0 = frequ0
frequency1 = frequ1

waveform0 = self.sinus_samples(frequency0)
waveform1 = self.sinus_samples(frequency1)

spcm_dwSetParam_i32(self.hCard, SPC_CARDMODE, SPC_REP_STD_SINGLE)
spcm_dwSetParam_i32(self.hCard, SPC_CHENABLE, CHANNEL0 | CHANNEL1)
spcm_dwSetParam_i32(self.hCard, SPC_ENABLEOUT0, 1)
spcm_dwSetParam_i32(self.hCard, SPC_ENABLEOUT1, 1)
spcm_dwSetParam_i32(self.hCard, SPC_AMP0, amplitude0)
spcm_dwSetParam_i32(self.hCard, SPC_AMP1, amplitude1)

spcm_dwSetParam_i64(self.hCard, SPC_MEMSIZE, self.L_ext) #Samples per channel!
spcm_dwSetParam_i32(self.hCard, SPC_LOOPS, 0)

pvBuffer = c_void_p()
pvBuffer = pvAllocMemPageAligned(self.BufferSize * 2)
pnBuffer = cast(pvBuffer, ptr16)
merged_waveforms = self.interleave(waveform0, waveform1)
pnBufferArray = np.ctypeslib.as_array(pnBuffer, shape=(self.BufferSize*2,))
pnBufferArray[:len(merged_waveforms)] = merged_waveforms.astype(int16)

spcm_dwDefTransfer_i64(self.hCard, SPCM_BUF_DATA, SPCM_DIR_PCTOCARD, 0, pvBuffer, 0, int64(self.BufferSize * 2))
spcm_dwSetParam_i32(self.hCard, SPC_M2CMD, M2CMD_DATA_STARTDMA | M2CMD_DATA_WAITDMA)
spcm_dwSetParam_i32(self.hCard, SPC_M2CMD, M2CMD_CARD_START | M2CMD_CARD_ENABLETRIGGER | M2CMD_CARD_FORCETRIGGER)
possible_error = self.getError()
if possible_error != ERR_OK:
    print(possible_error)
```

Bibliography

[1] J. Hecht, *Short history of laser development*, Optical Engineering **49** (2010) 091002 (cit. on p. 1).

[2] A. Ashkin, *Acceleration and Trapping of Particles by Radiation Pressure*, Phys. Rev. Lett. **24** (4 1970) 156 (cit. on p. 1).

[3] A. Ashkin, J. M. Dziedzic, J. E. Bjorkholm and S. Chu, *Observation of a single-beam gradient force optical trap for dielectric particles*, Opt. Lett. **11** (1986) 288 (cit. on p. 1).

[4] A. Ashkin, J. M. Dziedzic and T. Yamane, *Optical trapping and manipulation of single cells using infrared laser beams*, Nature **330** (1987) 769 (cit. on p. 1).

[5] O. M. Marago, P. H. Jones, P. G. Gucciardi, G. Volpe and A. C. Ferrari, *Optical trapping and manipulation of nanostructures*, Nature nanotechnology **8** (2013) 807 (cit. on p. 1).

[6] Hamamatsu, *LCOS-SLM (Liquid Crystal on Silicon - Spatial Light Modulator)*, URL: `https://hamamatsu.su/media/index/?type=catalog&id=115` (visited on 13/02/2024) (cit. on p. 1).

[7] C. Peng, R. Hamerly, M. Soltani and D. R. Englund, *Design of high-speed phase-only spatial light modulators with two-dimensional tunable microcavity arrays*, Opt. Express **27** (2019) 30669 (cit. on p. 1).

[8] G. Gauthier et al., *Direct imaging of a digital-micromirror device for configurable microscopic optical potentials*, Optica **3** (2016) 1136 (cit. on p. 1).

[9] AA Opto Electronic, *DTSX - DTSXY AO DEFLECTORS 1-axis/2-axis*, URL: `http://www.aaoptoelectronic.com/wp-content/uploads/2018/08/DTSxx-ed1-18.pdf` (visited on 13/02/2024) (cit. on p. 1).

[10] K. O. Roberts et al., *Steerable optical tweezers for ultracold atom studies*, Opt. Lett. **39** (2014) 2012, URL: `https://opg.optica.org/ol/abstract.cfm?URI=ol-39-7-2012` (cit. on p. 1).

[11] M. Endres et al., *Cold Matter Assembled Atom-by-Atom*, 2016, arXiv: `1607.03044 [quant-ph]` (cit. on pp. 1, 21).

[12] K.-N. Schymik et al., *Enhanced atom-by-atom assembly of arbitrary tweezer arrays*, Phys. Rev. A **102** (6 2020) 063107 (cit. on p. 1).

[13] W. Tian et al., *Parallel Assembly of Arbitrary Defect-Free Atom Arrays with a Multitweezer Algorithm*, Phys. Rev. Appl. **19** (3 2023) 034048 (cit. on p. 1).

[14] J. E. U. Castillero, *Imaging and addressing of neutral atoms inside a fiber cavity*, Master thesis: University of Bonn, 2017 (cit. on p. 1).

[15] M. O. Scully, E. S. Fry, C. H. R. Ooi and K. Wódkiewicz, *Directed Spontaneous Emission from an Extended Ensemble of N Atoms: Timing Is Everything*, Phys. Rev. Lett. **96** (1 2006) 010501 (cit. on p. 2).

[16] N. Stiesdal et al., *Controlled multi-photon subtraction with cascaded Rydberg superatoms as single-photon absorbers*, Nature Communications **12** (2021) 4328 (cit. on p. 2).

[17] B. E. A. Saleh and M. C. Teich, *Grundlagen der Photonik*, 2nd ed., Wiley-VCH, 2008, ISBN: 978-3-527-40677-7 (cit. on pp. 4–7).

[18] T. Yano, M. Kawabuchi, A. Fukumoto and A. Watanabe, *TeO2 anisotropic Bragg light deflector without midband degeneracy*, Applied Physics Letters **26** (2008) 689 (cit. on p. 6).

[19] M. Bass, E. Van Stryland, D. Williams and W. Wolfe, *Handbook of Optics Devices, Measurements, & Properties*, 2nd ed., McGraw-Hill, 1995, ISBN: 0-07-047974-7 (cit. on p. 6).

[20] R. Dixon, *Acoustic diffraction of light in anisotropic media*, IEEE Journal of Quantum Electronics **3** (1967) 85 (cit. on pp. 6, 7).

[21] R. Grimm, M. Weidemüller and Y. B. Ovchinnikov, *Optical dipole traps for neutral atoms*, 1999, arXiv: `physics/9902072 [physics.atom-ph]` (cit. on p. 8).

[22] *M4i.66xx-x8, M4x.66xx-x4 high-speed 16 bit AWG for PCI Express bus and PXI Express bus Hardware and Software Driver Man- ual*, (2014), URL: `https://spectrum-instrumentation.com/dl/m4i_m4x_66xx_manual_english.pdf` (visited on 08/01/2024) (cit. on pp. 12, 56).

[23] V. Jonas, *Acousto-Optic Sculpturing of Optical Potential Landscapes for Ultracold Fermions*, Master thesis: University of Bonn, 2021 (cit. on p. 13).

[24] S. K. Schnelle, E. D. van Ooijen, M. J. Davis, N. R. Heckenberg and H. Rubinsztein-Dunlop, *Versatile two-dimensional potentials for ultra-cold atoms*, Opt. Express **16** (2008) 1405 (cit. on p. 15).

[25] N. Stiesdal, *Collective atom-light interactions with Rydberg superatoms*, PhD thesis: University of Southern Denmark and University of Bonn, 2022 (cit. on p. 15).

[26] L. Neuhaus and S. Deléglise, *PyRPL documentation*, URL: `https://pyrpl.readthedocs.io/en/latest/` (visited on 07/02/2024) (cit. on p. 28).

[27] F. Vatansever and E. Haciiskenderoğlu, *PID tuning with up-to-date metaheuristic algorithms*, Uludağ University Journal of The Faculty of Engineering **27** (2022) 573 (cit. on p. 29).

[28] SciPy Contributors, *SciPy - RegularGridInterpolator Documentation*, https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RegularGridInterpolator.html, (visited on 23/01/2024) (cit. on p. 35).

[29] Arducam, *CMOS MT9J001 Camera Module 1/2.3-Inch 10MP Monochrome Module Datas*, Product specification document, 2017, URL: https://www.arducam.com/downloads/modules/industrial/1Inch2.3_10_Megapixel_MT9J001_CMOS_Camera_Module_DS.pdf (visited on 08/01/2024) (cit. on p. 40).

[30] S. Germer, *Frequency stabilization of a laser and a high resolution optical setup for excitation of ultracold Rydberg atoms*, Bachelor thesis: University of Bonn, 2023 (cit. on p. 41).

[31] S. Schroers, *Raman-Seitenbandkuehlung von Rubidiumatomen*, Bachelor thesis: University of Bonn, 2022 (cit. on p. 45).

[32] K. Kleinbeck et al., *Creation of nonclassical states of light in a chiral waveguide*, Phys. Rev. A **107** (1 2023) 013717 (cit. on p. 54).

[33] H. Busche et al., *Contactless nonlinear optics mediated by long-range Rydberg interactions*, Nature Physics **13** (2017) 655 (cit. on p. 54).