# Contents

1.	1. Introduction			
2.	Theory         2.1. Hyperfine Structure         2.2. Lasers         2.3. Spectroscopy Methods         2.4. Elements in the optical path	<b>4</b> 4 8 8 10		
3.	Setup of the Experiment	12		
4. 5.	Procedure         4.1. Determination of the Lasing Threshold	<ol> <li>13</li> <li>13</li> <li>15</li> <li>15</li> <li>16</li> <li>17</li> <li>19</li> </ol>		
6.	5.1. Determination of the Lasing Threshold         5.2. Frequency Calibration         5.3. Analysis of the Rb Spectra         5.4. Evaluation of the Spectroscopy Results         Summary and Discussion         6.1. Results	19 20 20 25 <b>31</b> 31		
	6.2. Discussion	$31 \\ 32$		
Re	eferences	34		
А.	Appendix         A.1. Setup	<b>35</b> 35 36 37 40 41		
в.	Python Code	46		

# List of Figures

1.	Term Schema	6
2.	Saturation Spectroscopy	10
3.	Setup	14
4.	Mode-Hop-Free Range	16
5.	Absorption Spectrum	17
6.	Saturation Spectrum	18
7.	Fluorescence Spectrum	18
8.	Determination of the Lasing Threshold	19
9.	FPI Calibration	21
10	. FPI Calibration on Frequency Axis	21
11	. Saturation Spectrum - Running Average	22
12	. Saturation Spectrum - Underground Subtraction	23
13	. Saturation Spectrum - Peak Positions	25
14	Absorption Spectrum - Peak Positions	26
15	Fluorescence Spectrum - Peak Positions	27
16	Spectroscopy Results	28
17	. Setup	35
18	. Saturation Spectrum - Large Range	36
19	Saturation Spectrum - Underground Considerations	37
20	Absorption Spectrum - Underground Considerations	38
21	. Fluorescence Spectrum - Underground Considerations	39
	* <del>`</del>	

# List of Tables

1.	Transition Frequencies	7
2.	Spectroscopy Results	29
3.	Percentage of 87-Rb in the Mixture	30
4.	Fit Results of the Saturation Spectrum	40
5.	Fit Results of the Absorption Spectrum	40
6.	Fit Results of the Fluorescence Spectrum	40

# 1. Introduction

Coupling atoms or molecules to laser light can lead to quantum interference in the amplitudes of optical transitions. This can drastically modify the optical properties of a medium. Thus, the medium may become transparent to another laser whose frequency is one that a non-prepared medium would absorb. This effect is called *Electromagnetically induced transparency (EIT)* [3, 8]. The preparation of a medium via a strong pump beam not only influences the absorption spectrum of the medium but also its refractive index. The modified optical properties of the medium can lead to extremely low group velocities for the second laser propagating through the medium. In fact, EIT makes it possible to produce stopped light. In the context of EIT, this refers to "trapping" a light pulse inside the prepared medium by switching off the beam that prepares the medium. There is experimental evidence of such trapped beams [4]. Another application of EIT is laser cooling of long strings of atoms [11], and slowing down light has various potential applications such as the construction of highly sensitive interferometers [12] or optical quantum information processing [13].

The one-week version of the EIT experiment is dedicated to the characterization and optimization of the laser source. The main goal is investigating the spectrum of the isotopes <sup>85</sup>Rb and <sup>87</sup>Rb and determining their hyperfine structure constants. This is done via three different spectroscopy methods, namely absorption spectroscopy, saturation spectroscopy and fluorescence spectroscopy. This report only covers the one-week version of the experiment and focuses on the Rubidium isotope spectra and the setup used to record the spectra.

This report is structured as follows: In Section 2, the theoretical concepts needed to understand the measurements done in the experiment as well as its results will be provided. Then, Sections 3 and 4 will detail the setup of the experiment and the steps taken to perform the measurements. In Section 5, the results of the measurements as well as the analysis of the data will be presented. Section 6 concludes the report by summarizing the results obtained by the experiment and discussing their quality as well providing an assessment of the experiment as a whole.

### 2. Theory

In this section, the theoretical background needed to understand the setup and structure of this experiment is summarized. If not stated otherwise, this section is based on *Atomic Physics* by Foot [9] and *Laser Spectroscopy* by Demtröder [6] as well as Ref. [5].

#### 2.1. Hyperfine Structure

Electrons orbiting a nucleus move in a coulomb potential generated by the electric charge of the nucleus. When considering the orbits of the electrons classically, only discrete orbits are "allowed". The energy difference of these orbits can be seen in form of spectral lines. Due to magnetic interaction between the electrons orbital angular momentum and their spin, the energy levels are slightly shifted. This resulting splitting of energy levels is called fine structure. In general, the energy shift of this effect is small compared to the energy difference of the original levels. Due to interaction of the nuclear spin with the total angular momentum of the orbiting electrons, the energy levels are further split up. This is called hyperfine structure.

Total Electronic Angular Momentum In a quantum system consisting of several electrons (such as an atom), their orbital angular momentum can be summed to obtain a total orbital angular momentum L. Similarly, summing their spin angular momenta results in a total spin angular momentum S. In LS-coupling, the total angular momentum of the system is given by J = L + S. The corresponding quantum number J can take integer values bounded by

$$|L - S| \le J \le |L + S|. \tag{1}$$

The total magnetic moment of the electrons is given by

$$\boldsymbol{\mu}_J = -\frac{\mu_B}{\hbar} g_J \boldsymbol{J} \,, \tag{2}$$

where  $\mu_B = e\hbar/(2m_e)$  is called Bohr magneton and  $g_J$  is the Landé factor

$$g_J = 1 + \frac{J(J+1) + S(S+1) - L(L+1)}{2J(J+1)}.$$
(3)

**Term Symbols** The electron configuration of an atom using LS-coupling can be described using term symbols. The general form of a term symbol is  $n^{2(S+1)}L_J$ , where *n* is the principal quantum number of the outer electron and *L* is the total orbital momentum written in letters using spectroscopic notation. An example for *L* in spectroscopic notation is *S* for L = 0 or *P* for L = 1.

**Hyperfine Structure** The hyperfine structure is the result of the interaction of the total nuclear angular momentum I with the total angular momentum of the electrons J. The total atomic angular momentum F is then given by

$$\boldsymbol{F} = \boldsymbol{I} + \boldsymbol{J} \,, \tag{4}$$

where the corresponding quantum number F can take integer values bounded by

$$|I - J| \le F \le |I + J|. \tag{5}$$

In this experiment, the transition lines of interest are the hyperfine transitions between the  $5^2 S_{1/2}$  and  $5^2 P_{1/2}$  states of <sup>85</sup>Rb and <sup>87</sup>Rb. Expressed in quantum numbers, both states have S = 1/2 and J = 1/2 and the transitions of interest are those between



Figure 1: Term schema of both the <sup>85</sup>Rb and <sup>87</sup>Rb Rubidium isotopes, adapted from Ref. [5]. The energy shifts are not up to scale. The values are taken from Refs. [14, 15].

L = 0 and L = 1. The ground state of <sup>85</sup>Rb has a total nuclear angular momentum of I = 5/2. According to Eq. (5), this means that F can take the values F = 2 or F = 3 for both the  $5^2S_{1/2}$  and  $5^2P_{1/2}$  states. The total nuclear angular momentum of the ground state of <sup>87</sup>Rb is I = 3/2, resulting in F = 1 or F = 2. The hyperfine splitting of <sup>85</sup>Rb and <sup>87</sup>Rb is pictured in Fig. 1. The  $D_1$ -line pictured in Fig. 1 corresponds to the fine-structure transition between the  ${}^2S_{1/2}$  and  ${}^2P_{1/2}$  states and has (within the corresponding uncertainties) the same frequency for both Rubidium isotopes [14, 15].

The goal of this experiment is to observe the spectrum of the Rubidium isotopes caused by hyperfine transitions. The selection rules allow only for transitions with  $\Delta L = \pm 1$ . The expected transition lines between the  ${}^{2}S_{1/2}$  and  ${}^{2}P_{1/2}$  states of the Rubidium isotopes along with their frequency differences to the  $D_{1}$ -line of  $3.77 \cdot 10^{14}$  Hz are listed in Table 1. All transitions that are expected to be observed in this experiment are transitions between the  ${}^{2}S_{1/2}$  and  ${}^{2}P_{1/2}$  states. In the following, the transitions are denoted by

$${}^{A}\mathrm{Rb}_{F_{L=S}}^{F_{L=P}},$$
(6)

where A is the mass number of the isotope and  $F_{L=P}$  and  $F_{L=S}$  mark the hyperfine states of the upper  ${}^{2}P_{1/2}$ -state and the lower  ${}^{2}S_{1/2}$ -state, respectively. An example for this notation is  ${}^{87}\text{Rb}_{F=2}^{F=1}$ , which refers to transition from the  ${}^{87}\text{Rb}$  state  ${}^{2}P_{1/2}$  with F = 1

 $\mathbf{6}$ 

$\operatorname{to}$	the	$^{2}S_{1/2}$	$\operatorname{state}$	with	F	=	2
---------------------	-----	---------------	------------------------	------	---	---	---

Transition	Frequency offset
${}^{87}\mathrm{Rb}_{F=2}^{F=1}$	$-3.07\mathrm{GHz}$
${}^{87}\mathrm{Rb}_{F=2}^{F=2}$	$-2.25\mathrm{GHz}$
${}^{85}\mathrm{Rb}_{F=3}^{F=2}$	$-1.48\mathrm{GHz}$
${}^{85}\text{Rb}_{F=3}^{F=3}$	$-1.22\mathrm{GHz}$
${}^{85}\mathrm{Rb}_{F=2}^{F=2}$	$1.56\mathrm{GHz}$
${}^{85}\mathrm{Rb}_{F=2}^{F=3}$	$1.92\mathrm{GHz}$
${}^{87}\mathrm{Rb}_{F=1}^{F=1}$	$3.76\mathrm{GHz}$
${}^{87}\mathrm{Rb}_{F=1}^{F=2}$	$4.58\mathrm{GHz}$

Table 1: Hyperfine transitions between the  ${}^{2}P_{1/2}$  and  ${}^{2}S_{1/2}$  states of both Rubidium isotopes. The notation used for the transitions is explained in Eq. (6). The frequency differences of the transition lines to the  $D_1$ -line of  $3.77 \cdot 10^{14}$  Hz are also listed. The values are calculated from Refs. [14, 15].

**Hyperfine Structure Constants** The energy shift caused by hyperfine splitting is given by [9]

$$\Delta E_{\rm HFS} = \frac{A}{2} [F(F+1) - I(I+1) - J(J+1)],$$

where A is the hyperfine structure constant. The energy difference of states with the same quantum numbers I and J and neighboring F, that is  $F_2 = F_1 + 1$ , is then given by

$$\Delta E_{F_2=F_1+1} = A(F_1+1), \qquad (7)$$

where  $F_1$  is the lower of the two quantum numbers F. This energy difference can be rewritten in terms of the frequency  $\nu$  emitted by the corresponding transition via

$$\Delta E = h \Delta \nu \,, \tag{8}$$

where h is the Planck constant. Thus, by combining Eqs. (7) and (8), the hyperfine structure constant can be expressed in terms of the frequency of transitions between atomic states with the same I and J and neighboring F by

$$A = \frac{h\delta\nu}{F_1 + 1}\,,\tag{9}$$

where as before  $F_1$  refers to the lower of the quantum numbers for the total atomic momentum.

#### 2.2. Lasers

Lasers are a source of coherent light, which means that the emitted light is monochromatic and in phase. They consist of three main components: An active laser medium, a pumping mechanism and a resonating cavity.

When an atom in the gain medium is excited, it can relax into a lower energy state, and the excess energy is emitted in form of a photon. This is called *spontaneous emission*. This photon will have a specific wave length due to the discrete energy levels in the atom. Interaction with a photon whose energy matches the excitation energy can cause the de-excitation of an excited state. The photon emitted in the de-excitation has the same frequency, polarization, phase and direction of travel as the interaction photon. This process is called *stimulated emission*. If the two photons then interact with other excited atoms, stimulated emission occurs again and the light is amplified.

In a laser, this effect is used to produce coherent light. The active laser medium is excited by the pumping mechanism. Then, a spontaneous emission causes stimulated emission in the active laser medium. The resonating cavity provides optical feedback to single out a specific frequency which is predominantly amplified. This can for example be realized by placing the active laser medium into an arrangement of mirrors of which one is partially transparent. The reflection at the mirrors causes frequencies that form a standing wave to be amplified due to constructive interference, while other frequencies experience destructive interference. The light passes through the active laser medium several times, causing stimulated emission. It can then leave the cavity through the partially transparent mirror, forming a laser beam.

However, in order to consistently emit light, a population inversion must be maintained. This means that there must be more atoms in an excited state than in the ground state. Population inversion is a requirement for laser operation, as the light emitted by relaxing atoms are of exactly the frequency needed to excited the atoms. If there are more atoms in ground state, the light emitted by one atom will excite another atom. Only if population inversion is achieved, the rate of stimulated emission is higher than the rate of absorption, and there is a net light amplification. The minimal pump power at which this is the case is called *lasing threshold*. In general, the output power of a laser rises with increasing pump power. Above the lasing threshold, the slope of output power to pump power is orders of magnitude higher than below the threshold.

#### 2.3. Spectroscopy Methods

Spectroscopic investigations allow insights into the structure of atoms and molecules. The frequency at which spectral lines of an atom occur corresponds to the difference between energy levels, and the line intensity is proportional to the corresponding transition probability. In this experiment, the spectrum of <sup>85</sup>Rb and <sup>87</sup>Rb is recorded by monitoring the absorption of laser photons.

When the frequency of the laser photons is tuned to the energy difference of an atomic transition, the laser photons can excite an atom to a higher atomic state. This excited state will spontaneously decay, emitting a photon of the same frequency as the laser photon. In contrast to the laser photons, a photon originating in spontaneous decay is not emitted in a preferred direction. If however a laser photon interacts with an already excited atom, the atom can de-excite via stimulated emission. In this case, the photon emitted in the de-excitation is identical to the laser photons.

In the following paragraphs, different spectroscopy methods employed in this experiments are briefly discussed. If not stated otherwise, this subsection is based on *Laser Spectroscopy* by Demtröder [6].

**Fluorescence Spectroscopy** Fluorescence spectroscopy detects spontaneously emitted photons stemming from a de-excitation of an atomic state excited by the laser. This is done by placing a photo diode next to the sample, perpendicular to the propagation of the incident laser beam. The recorded spectrum will show peaks at the spectral lines of the sample.

**Absorption Spectroscopy** Absorption spectroscopy measures the laser beam after passing through the sample. Photons at wavelengths that do not correspond to spectral lines of the sample will pass through the sample. However, photons originating in the spontaneous de-excitation of an atom are emitted in an arbitrary direction. Thus, at wavelengths corresponding to spectral lines, less photons will be recorded behind the sample. Consequently, the absorption spectrum will show dips at the spectral lines of the sample.

**Saturation Spectroscopy** Spectral lines are subject to broadening effects, such as Doppler broadening. Doppler broadening is caused by thermal movement of the atoms in the sample. For an atom moving parallel to the propagation of laser photons, the frequency of the laser photons will be Doppler shifted. Thus, even if the frequency of the laser photon is slightly below or above the frequency needed to excite an atom, the laser may excite atoms which are moving with respect to the propagation of the laser photons. As a consequence, the peak or a dip corresponding to the excitation energy can already be seen at these slightly shifted energies. In other words, the peaks or dips of the spectrum broaden. As only the velocity component in direction of the incident laser beam is relevant for Doppler broadening, the distribution of the intensity caused by Doppler broadening is Gaussian.

Saturation spectroscopy is a method to overcome Doppler broadening without having to cool the sample to millikelvin temperatures. The main idea is the following: A strong pump beam, that is a laser beam with high intensity, is directed into the sample. A weaker probe beam at the exact same wavelength is directed into the sample from the opposite direction. The probe beam is then recorded by a photo diode. If the photon energy is slightly below the excitation energy of the sample, the pump beam will excite Doppler shifted atoms moving towards the source of the pump beam. The probe beam, coming from the opposite direction, will excite Doppler shifted atoms moving away from the source of the pump beam. Thus, the beams do not excite the same atoms. Just as in the absorption spectrum, this will cause a lower intensity in the recorded probe beam. The same principle applies to photon energies slightly above the excitation energy.

If however the photon energy matches the excitation energy of the sample, the atoms that can be excited by the probe beam can also be excited by the pump beam and vice versa. The stronger pump beam will cause a lot of the atoms to be in an excited state. Then, the probe beam will encounter these excited atoms, causing stimulated emission. Thus, the intensity of the probe beam at this energy is slightly higher than for energies slightly above or below the excitation energy: The spectrum will show a small peak inside the absorption dip. This peak is not affected by Doppler broadening. Its line width results from the natural line width influenced by the much narrower Lamb dip. An example for a spectral line observed using saturation spectroscopy is pictured in Fig. 2.



Figure 2: Example for an absorption line observed using saturation spectroscopy. The figure is taken from Ref. [10].

#### 2.4. Elements in the optical path

In this subsection, the properties of some elements that are part of the optical path are discussed.

**Polarizing Beam Splitter** A beam splitter splits the light of a beam into a reflected and a transmitted beam. A polarizing beam splitter achieves this by means of a birefringent material: The incident light beam is split into two beams of orthogonal polarization, one of which gets transmitted while the other is reflected.

**Waveplates** A waveplate changes the polarization state of light passing through. This is done using birefringent materials. Birefringent materials are materials for which the

index of refraction differs for linearly polarized light depending on the polarization axis of the light. Thus, the different polarizations of the incoming laser beam will experience a phase shift with respect to each other. By carefully choosing the thickness of the waveplate, the angle by which the phase shifts can be controlled. Two common types of wave plates are quarter-wave plates and half-wave plates. A half-wave plate will shift the phase by  $\pi$ , which corresponds to half a wavelength. This rotates the polarization axis of linear polarized light and changes the handedness of circularly polarized light. A quarter-wave plate will shift the phase by  $\pi/2$ , corresponding to a quarter of the wavelength, and will convert linearly polarized light into circularly polarized light and vice versa. As the phase shift also depends on the wave length, waveplates typically only work for a particular range of wavelengths.

**Fabry-Pérot Etalon** A Fabry-Pérot interferometer (FPI) is an optical resonator consisting of two partially transmitting mirrors or a material that partially reflects light at the layers through which the light enters or leaves. If the distance between the mirrors can not be adjusted, it is also called etalon. When a light beam enters the etalon from the left side, it is partially reflected at the right side of the etalon. A portion of this reflected light is once again reflected at the mirror through which it enters the etalon and interferes with the light that was initially transmitted through the right side. Whether this interference is constructive or destructive depends on the thickness L of the etalon and the wave length of the light: If the wave length of the light is an integer multiple of half the thickness,

$$\lambda = \frac{mL}{2} \,, \qquad m \in \mathbb{N} \,,$$

the interference is constructive. Thus, the transmission spectrum of etalon consists of thin peaks at the resonating wave lengths and is almost zero otherwise. The frequency distance  $\delta\nu$  between two peaks

$$\delta\nu = \frac{c}{2L}\,,\tag{10}$$

does not depend on the wavelength. It is referred to as *free spectral range (FSR)*. In the above equation, c refers to the speed of light in the etalon material. Thus, the transmission spectrum of etalon in frequencies shows equidistant thin peaks. The width of the peaks is characterized by the finesse, which is defined as

$$\mathcal{F} \coloneqq \frac{\Delta\lambda}{\delta\lambda} \,, \tag{11}$$

where  $\Delta\lambda$  is the free spectral range in wavelengths and  $\delta\lambda$  is the full-width-half-maximum (FWHM) of the peak. The FPI used in this experiment has a FSR of 700 MHz and a finesse of 300.

**Neutral density filter** A neutral density filter (ND filter) is a filter that reduces the intensity of all wavelengths equally. This means that it can for example be used to prevent saturation in a photo diode without distorting the spectrum. The ND filter

used in this experiment is a reflective continuous variable ND filter, which means that it has a continuous range of optical densities onto which an incoming laser beam can be directed.

**Diode Laser** In this experiment, the grating stabilized laser head DL pro 020011 by Toptica is used as a source for the laser photons used for spectroscopy. It is controlled by the digital laser controller DLC pro 020012 by Toptica. The following explanation is based on the manuals for the laser head [2] and for the laser controller [1].

The laser head uses a laser diode as a source of light. Such a laser diode has a linewidth of almost 100 MHz. The emission frequency of the laser diode can be tuned by adjusting the current and temperature. However, when adjusting the current at constant temperature or vice versa, large, discontinuous hops in frequency can occur. These hops are called mode hops. The range in which the frequency can be adjusted continuously is called mode-hop-free range.

For the purpose of spectroscopy, the emitted linewidth should be as narrow as possible and the spectrum has to be continuous. A grating laser head improves both the linewidth and the mode-hop free range. The function principle is as follows: The light emitted by the front facet of the laser diode is collimated and then strikes a reflection grating. The grating is angled such that the first diffraction order of the grating is reflected back towards the photo diode. This setup is called Littrow-setup [6]. As the light passes the collimator again, it is focused back into the laser diode resonator. The feedback from the grating is higher than the feedback from the front facet. Thus, the light begins to oscillate between the grating and the back facet, forming a new resonator. This new resonator typically has a FSR of a few centimeters and a linewidth of about 1 MHz, which is much smaller than the linewidth of the laser diode. The frequency of the light that is reflected back into the collimator can be continuously adjusted by tilting the grating. The angle of the grating is controlled by a piezo element.

The laser controller offers a scan mode, in which a scan over a range of frequencies is performed. The grating is moved by applying voltage to the piezo element. In scan mode, a variable voltage is added to a constant piezo voltage referred to as scan offset. The scan amplitude determines the maximum voltage added to the scan offset. The piezo voltage is then continuously adjusted between the minimum and maximum voltage. The shape of the piezo voltage can be chosen to be a triangle shape or a sine shape. The frequency of the scan is set by the scan frequency.

To improve the mode-hop free range, a ramp proportional to the piezo scan ramp can be applied to the laser diode current. The slope of the ramp can be adjusted by setting feed forward.

## 3. Setup of the Experiment

The aim of this experiment is to measure the spectrum of <sup>85</sup>Rb and <sup>87</sup>Rb using absorption spectroscopy, saturation spectroscopy and fluorescence spectroscopy. The basic setup for

all three measurements differs only slightly.

First, the light beam from a diode laser is passed through an isolator. An isolator is an optical device through which light can only pass in one direction. This prevents any reflections from re-entering the diode laser. Next, it is passed through a polarizing half-wave plate. This ensures light polarized in the correct angle before entering the polarizing beam splitter. After passing a series of mirrors, the beam enters a polarizing beam splitter. The polarization that passes the beam splitter is sent to the FPI. The other polarization is passed through the Rubidium cell and a quarter wave plate. Then, it is reflected at a mirror positioned perpendicular to the beam. This ensures that the beam passes the Rubidium cell at the same spot, which is important for the saturation measurement. This beam, having passed a quarter-wave plate twice, is now polarized perpendicular to its original orientation after the beam splitter. Thus, when entering the beam splitter again, it does not reflect but is instead transmitted.

For the absorption measurement, the beam is then sent on to a photo diode, which is used to record the absorption spectrum. For the saturation measurement, a ND filter is added between the quarter-wave plate and the mirror. The photo diode is in the same position as for the absorption measurement. This means that the beam returning through the Rubidium cell will be strongly attenuated, which makes it possible to use the returning beam as the probe beam needed for saturation spectroscopy. For the fluorescence measurement, the ND filter is removed and the photo diode is placed next to the Rubidium cell. The basic setup with the variations for each of the spectroscopy methods is sketched in Fig. 3.

### 4. Procedure

#### 4.1. Determination of the Lasing Threshold

In order to determine the lasing threshold, the setup was modified such that the laser beam was directed onto the photo diode without passing the rubidium cell. The photo diode was connected to the oscilloscope. The photo diode current was read off the oscilloscope for laser **set current** in a range of 60 mA to 140 mA in steps of 1 mA. As the lasing threshold was observed at a current of about 87 mA, the photo diode current was additionally recorded between 86 mA and 88 mA in steps of 0.1 mA.

#### 4.2. Configuration of the Main Setup

The configuration of the main setup used for spectroscopy can in general be separated into two parts: The setup and adjustment of the optical devices pictured in Fig. 3 and fine tuning the diode laser controller settings that control the current scan and the feed forward. They will be briefly explained in the following.



Figure 3: Schematic setup of the experiment. The optical elements pictured above are the diode laser (DL), isolator (ISO), half-wave plate (HWP), quarter-wave plate (QWP), polarizing beam splitter (PBS), rubidium cell (RB), photo diode (PD), ND filter (ND), Fabry-Pérot interferometer (FPI) and mirrors (M1) – (M7). Optical devices that are present only for saturation spectroscopy are marked in orange (dotted), those for absorption and saturation are marked in blue (dashed) and those only used for fluorescence spectroscopy are marked in green (dotted and dashed).

#### 4.2.1. Setup of the Optical Devices

First, all optical devices except for the Rubidium cell and the ND filter were positioned as pictured in Fig. 3. The laser was turned on and a scan over a range of wavelengths at a scan frequency of 10 Hz was started. A laser viewing card was used to check the height of the beam throughout the setup. Then, all mirrors were adjusted until the height was constant. Both the photo diode and the Fabry-Pérot interferometer were connected to the oscilloscope. Then, the height of the photo diode and the FPI was adjusted until the signal on the oscilloscope was maximized.

To ensure that the beam splitter transmits about half the incoming light, the half-wave plate was turned until the intensity of transmitted and reflected light was approximately equal. This was checked by observing the beam through a smartphone camera on a piece of paper.

The light that is originally reflected at the beam splitter should optimally be completely transmitted when once again being directed through the beam splitter. To set the QWP such that the laser beam experiences the desired change in polarization of  $\pi/2$ , the intensity of the beam after the beam splitter (this corresponds to the left side of the beam splitter when referring to Fig. 3) was observed again through a smartphone camera on a piece of paper. Then, the axis of the QWP was adjusted such that the observed intensity was maximal.

The FPI can only record the light if it enters it from the front. To ensure this and to optimize the signal, mirrors (M6) and (M7) were turned until the amplitude observed at the oscilloscope was maximized and the expected equidistant peak-structure was visible.

#### 4.2.2. Fine Tuning of the Diode Laser

The main goal in tuning the diode laser is finding a mode-hop free range in which measurements can be performed. The wavelength range the diode laser scans over depends on the temperature, the set current, the scan offset, the scan amplitude and the feed forward setting. The range in which there are no mode hops is influenced by the combination of all these settings. An example of a mode-hop free range and a range containing mode hops can be found in Fig. 4.

In order to perform spectroscopy measurements, the Rubidium cell was inserted into the optical path as pictured in Fig. 3. Then, the wavelength range at which absorption can be observed was found by scanning through a range of diode laser currents. After having observed absorption, the mode-hop free range was improved by trying different combinations of set current, scan offset, scan amplitude and feed forward. The signal quality was also strongly dependent on the angle of the Rubidium cell relative to the optical path. Because the FPI peaks were very clear and had a comparatively high amplitude, it was possible to reduce the intensity of the beam that was directed into the FPI. This in turn made it possible to increase the intensity of the beam measuring the Rubidium spectrum and thus improving the data used for analysis. This was done by



Figure 4: Example for a mode-hop free range and a range with mode hops. The photo diode current can be seen in yellow and the FPI current in cyan. The blue signal was used as trigger. The settings for the range with mode hops in Fig. 4a are a scan offset of 27.3, a scan amplitude of 10.3, a set current of 166.6 mA and a feed forward of 1.10. The settings for the mode-hop free range in Fig. 4b are a scan offset of 21.1, a scan amplitude of 10.3, a set current of 172.2 mA and a feed forward of -0.81.

slightly adjusting the half-wave plate, which changes the polarization axis of the beam before entering the beam splitter. The oscilloscope was used to ensure that the FPI signal was still clear enough to be used for calibration.

Most of the steps in setting up the optical path and fine tuning the diode laser detailed in Sections 4.2.1 and 4.2.2 were not performed only once, but repeated over and over again in order to iteratively optimize the signal.

#### 4.3. Absorption Spectroscopy

After carrying out the steps detailed in Section 4.2, the spectrum showing absorption peaks was exported from the oscilloscope. It was confirmed that the range in which the spectrum was recorded was mode-hop free by removing the Rubidium cell after the measurement. An example for the absorption spectrum and the corresponding spectrum after removing the Rubidium cell can be seen in Fig. 5.

#### 4.4. Saturation Spectroscopy

As pictured in Fig. 3, the setup for saturation spectroscopy differs from that for absorption spectroscopy only by an additional ND filter. After finding suitable diode laser settings for absorption spectroscopy, the filter was inserted. The ND filter caused some of the incoming beam to be reflected. To prevent the reflection from reentering the Rubidium cell at the point and contributing to the signal recorded by the photo diode, the filter was inserted at a slight angle with respect to the optical path. The strength of the filter was varied until the saturation peaks were visible. The laser diode settings as well





- Figure 5: Example for the Rubidium spectrum and the corresponding spectrum without the Rubidium cell. The photo diode current can be seen in yellow and the FPI current in cyan. The blue signal was used as trigger. It can clearly be seen that the mode hops to the left and the very right are present in both recordings. In contrast to this, the dips in the middle vanish when the cell is removed an can thus be identified as
  - the dips in the middle vanish when the cell is removed an can thus be identified as absorption dips caused by the Rubidium. The laser diode settings were a scan offset of 11.38, a scan amplitude of 23.8, a set current of 155.58 mA and a feed forward of -1.122.

as the positions of all elements in the optical path were carefully adjusted until the peaks were as clear as possible. An example for the saturation peaks and the corresponding spectrum without the ND filter can be seen in Fig. 6.

#### 4.5. Fluorescence Spectroscopy

As a last step, the fluorescence spectrum was recorded. The photo diode was moved to be approximately perpendicular to the Rubidium cell as pictured in Fig. 3. As before, setup steps were once again repeated to improve the signal and the mode-hop free range. An example for the fluorescence spectrum and the corresponding spectrum without the Rubidium cell is pictured in Fig. 7.



Figure 6: Example for the saturation spectrum and the corresponding absorption spectrum without ND filter. The photo diode current can be seen in yellow and the FPI current in cyan. The blue signal was used as trigger. The saturation peaks not present in the absorption spectrum are clearly visible. The laser diode settings were a scan offset of 11.88, a scan amplitude of 19.8, a set current of 156.10 mA and a feed forward of -1.002.



Figure 7: Example for the fluorescence spectrum and the corresponding spectrum without the Rubidium cell. The photo diode current can be seen in yellow and the FPI current in cyan. The blue signal was used as trigger. The laser diode settings were a scan offset of 10.28, a scan amplitude of 20.10, a set current of 157.70 mA and a feed forward of -0.902.

# FP-II

### 5. Data Analysis

The data analysis of the whole experiment was performed in python. If not mentioned otherwise, the fits were conducted performing a least squares minimization using scipy.optimize.curve\_fit. The python code used for the analysis is shown in Appendix B.

#### 5.1. Determination of the Lasing Threshold

Before any spectroscopy measurements could be started, some measurements and observations had to be made to get a better understanding of the laser.

The measurements to determine the lasing threshold were performed as described in Section 4.1. The resulting photo diode intensity for each set current can be seen in Fig. 8.



Figure 8: Plot of the measured photo diode intensity against the corresponding set current of the laser. The inset shows a range around the set current of the lasing threshold where additional data has been recorded in 0.1 mA steps. The axis of the inset are the same as in the large plot. An estimation for the lasing threshold and its uncertainties are shown in green.

The lasing threshold corresponds to the set current at which the laser output is dominated by stimulated emission. This can be seen in a spontaneous increase in the slope of the output intensity. Using the zoom in, the lasing threshold could be determined to

$$I_T = (87.0 \pm 0.1) \,\mathrm{mA} \,.$$
 (12)

It can be mentioned that the measured intensity is not very linear at larger set currents, as it would have been expected. This is probably an artifact of the photo diode which has been used in the amplified setting and thus saturates at about 10 V. For the later measurements, the photo diode was used in the non amplified setting. Thus, the laser could be operated at high set currents between 150 mA to 170 mA without saturating the photo diode. The high set currents were chosen because they correspond to high intensities and therefore large absorption and fluorescence effects.

#### 5.2. Frequency Calibration

After further fine tuning of the laser and the setup as described in Section 4, large mode-hop free ranges could be identified in which the Rb measurements could be made. For each measurement, the corresponding FPI spectrum was recorded to perform a calibration from channels on the oscilloscope to frequency-differences.

An example for the recorded FPI spectrum can be seen in Fig. 9. The peak positions were estimated automatically in python by identifying the channels with the largest amplitude. The uncertainties were estimated to  $\pm 5$  channels.

The distances in channel of the peaks in the FPI spectrum during a frequency-sweep of the laser correspond exactly to the frequency of the free spectral range of the FPI. In the instructions [7], a FSR of 700 MHz and a finesse of 300 were given. Unfortunately, it can be seen that the frequency-sweep of the laser was not as consistent as required. This resulted in varying channel-differences of  $\Delta ch = 178$  channels to 209 channels, as also pictured in Fig. 9.

To get a more accurate channel to frequency conversion for the analysis, it was decided not to perform a linear fit but instead to use linear interpolation between two adjacent FSR peaks. This makes it possible to correct for the observed systematic deviations. The linear interpolation was performed using the numpy.interp method.

With this interpolation, the FPI spectrum can be shown on a frequency-axis, which is pictured in Fig. 10.

This analysis of the FPI spectrum and the interpolation were performed for each Rb spectrum separately, as between measurements minor changes to the laser settings were made in order to optimize the mode-hop free range. The FPI spectrum used for frequency calibration is therefore always shown together with the corresponding spectra.

#### 5.3. Analysis of the Rb Spectra

For the final analysis of the Rb spectra, multiple preparation steps were required to perform the necessary fits and determine the relevant quantities. These are shown exemplarily for a recorded saturation spectrum. The steps were performed in the same manner in the analysis of the fluorescence and the absorption spectrum.



Figure 9: Example of an FPI spectrum taken with the oscilloscope during a frequencysweep of the laser. Considering the properties of an FPI, the channel-differences of the peak positions correspond exactly to the free spectral range of the FPI. Thus, a calibration from channel-differences to frequency-differences is possible. The FPI spectrum on a frequency axis is visible in Fig. 10. The channel-differences between two adjacent peaks ( $\Delta$  ch) are listed in the plot as well.



Figure 10: FPI spectrum on a frequency axis.

**Preparation of the Oscilloscope Data** A recorded oscilloscope spectrum of the photo diode with the corresponding calibration spectrum of the FPI can be seen in Fig. 11. The FPI calibration from channels to frequency-differences has already been performed, as explained in Section 5.2.

When plotting the data points individually, it is evident that the raw data on the oscilloscope consists of a double-line. This effect is common to oscilloscopes and probably results from internal reflections. To be able to perform fits to the spectrum later on, a running average over 11 values was applied to the data to obtain one single line. It is shown in Fig. 11 as well.



Figure 11: Recorded oscilloscope spectrum of the photo diode during a frequency-sweep of the laser. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows a saturation spectrum recorded with the photo diode (orange). To correct for the double-line, a running average over 11 values has been performed (green).

During a frequency-sweep of the laser not only the frequency of the emitted beam but also its intensity varies. This results in a linear background to the photo diode spectrum. As this linear decrease of intensity would shift and distort the applied fits, it has to be corrected. To subtract the linear background of the spectrum, first the data points not influenced by the Rb absorption were selected. In Fig. 12, they are shown in red. A

$$A = m \cdot \delta \nu + c \tag{13}$$

was performed using the selected data points. The fit results were then used to subtract the linear background from the spectrum. The result of the fit as well as the corrected spectrum are also pictured in Fig. 12.



Figure 12: Recorded oscilloscope spectrum of the photo diode during a frequency-sweep of the laser. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows a saturation spectrum recorded by the photo diode after having applied a running average (green). To subtract the linear background, a linear fit of form Eq. (13) was performed to the linear data points indicated in red. The data after subtracting the linear background is pictured in blue.

With these considerations to all Rb spectra, it was possible to obtain clean spectra to which Gaussian fits could be performed in order to find the Rb peak positions and amplitudes. The applied changes to the underground are shown also for the other spectra in Figs. 19 to 21.

**Saturation Spectroscopy** After applying the corrections as described above, the saturation spectrum could be analyzed. As explained in Section 2.3, saturation spectroscopy is a special kind of absorption spectroscopy. Each absorption peak can be approximated by a Gaussian function due to the limited resolution of the photo diode. In addition, each absorption peak (with negative amplitude) will show an additional saturation peak with positive amplitude. The saturation peak will be much smaller and narrower due than the absorption peak since it is not affected by Doppler broadening. Thus, a full transition peak in the saturation spectrum can be approximated by the following sum of two Gaussian curves,

$$S(\delta\nu) = A_1 \cdot \exp\left(-\frac{(\delta\nu - \mu_1)^2}{2\sigma_1^2}\right) + A_2 \cdot \exp\left(-\frac{(\delta\nu - \mu_2)^2}{2\sigma_2^2}\right),$$
 (14)

where the constraints  $A_1 < 0$ ,  $A_2 > 0$ ,  $|A_1| > |A_2|$ ,  $\mu_1 \approx \mu_2$  and  $\sigma_1 > \sigma_2$  hold from the previous considerations. The mean value corresponding to the saturation peak,  $\mu_2$ , can then be used as an estimate for the frequency of the transition line. Fits of this form have been applied to each of the saturation peaks in the spectrum and can be seen in Fig. 13. The fit results are listed in Table 4.

**Absorption Spectroscopy** The absorption spectrum of the Rb cell was measured in the same setup after removing the additional ND-filter. To be able to perform the necessary fits, the same background considerations as in Section 5.3 were performed. The spectrum after having applied a running average and subtracted the linear underground is pictured in Fig. 14.

As argued in the analysis of the saturation spectrum, absorption peaks can be approximated by Gaussian functions with negative amplitude. Thus, fits of the form

$$A(\delta\nu) = A \cdot \exp\left(-\frac{(\delta\nu - \mu)^2}{2\sigma^2}\right)$$
(15)

with A < 0 were performed to the spectrum and are pictured Fig. 14 as well. The fit results are listed in Table 5. Compared to the peaks seen in the saturation spectrum, it can be seen that one peak could not properly be extracted from the data. Its position can be estimated to lie between the two leftmost peaks. This observation is qualitatively important in the identification of the transitions but is not used in the further analysis.

**Fluorescence Spectroscopy** For the fluorescence spectroscopy, the photo diode was placed directly next to the Rubidium cell. As a result, only photons emitted from the cell are recorded. Thus, the fluorescence peaks are expected to resemble positive Gaussian functions with a positive amplitude. The recorded spectrum together with the performed fits are shown in Fig. 15.

As one one of the peaks was very closely merged with the two adjacent fluorescence peaks, the peaks were not fitted separately to take this influence into account. Instead,



Figure 13: Recorded saturation spectrum of the photo diode during a frequency-sweep of the laser in the saturation spectroscopy setup. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows the saturation spectrum after subtracting a linear background as explained in Section 5.3. Saturation fits of the form Eq. (14) have been performed and can be seen in red. The single contributions of the absorption peaks and the saturation peaks are shown in blue and orange respectively. The fit results are listed in Table 4.

a sum of four individual Gaussian peaks was fitted to the spectrum. It can be described by

$$A(\delta\nu) = \sum_{i=1}^{4} A_i \cdot \exp\left(-\frac{(\delta\nu - \mu_i)^2}{2\sigma_i^2}\right)$$
(16)

where  $A_i > 0$  for  $i \in \{1 \dots 4\}$ . The result of the fit is shown in Fig. 15 and Table 6.

#### 5.4. Evaluation of the Spectroscopy Results

Accuracy of the Saturation Spectroscopy Measurement To estimate the improvement in accuracy of saturation spectroscopy over the other spectroscopy methods,



Figure 14: Recorded absorption spectrum of the photo diode during a frequency-sweep of the laser in the absorption spectroscopy setup. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows the absorption spectrum after subtracting a linear background as explained in Section 5.3. Gaussian fits of the form Eq. (15) have been performed and can be seen in orange. The fit results are listed in Table 5.

the widths of the saturation peaks can be compared to the widths of the absorption peaks. The widths  $\mu_2$  (compare Eq. (14)) of the saturation peaks are  $(68 \pm 3)$ % to  $(78.9 \pm 1.2)$ % smaller compared to the widths  $\mu_1$  of the corresponding absorption peaks, which are pictured in Fig. 13. This qualifies the saturation-spectroscopy method as a much more precise spectroscopy method compared to common absorption spectroscopy.

**Identification of the Observed Transitions** With the results of the three spectroscopy methods, it is possible to identify the relevant transitions of the Rubidium isotopes  ${}^{85}$ Rb and  ${}^{87}$ Rb. As only frequency-differences can be deduced from the recorded data, the frequency-differences from each peak to the lowest peak, which was identified as the lowest  ${}^{87}$ Rb peak, were calculated. The calculation of uncertainties using Gaussian



Figure 15: Recorded fluorescence spectrum of the photo diode during a frequency-sweep of the laser in the fluorescence spectroscopy setup. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows the fluorescence spectrum after subtracting a linear background as explained in Section 5.3. Gaussian fits of the form Eq. (16) have been performed and are pictured in orange. The fit results are listed in Table 6.

error propagation is give as follows:

$$\delta\nu\left(\mathrm{Rb}_{F=j}^{F=i}\right) = \delta\nu\left(\mathrm{Rb}_{F=j}^{F=i}\right) - \delta\nu\left(^{87}\mathrm{Rb}_{F=2}^{F=1}\right)$$
(17)

$$s_{\delta\nu\left(\mathrm{Rb}_{F=j}^{F=i}\right)} = \sqrt{\sigma\left(\delta\nu\left(\mathrm{Rb}_{F=j}^{F=i}\right)\right)^2 + \sigma\left(\delta\nu\left(^{87}\mathrm{Rb}_{F=2}^{F=1}\right)\right)^2}.$$
(18)

The notation used for the relevant Rubidium transitions is explained in Eq. (6). The uncertainties of each frequency are taken from the respective standard deviation  $\sigma$ , as the fit-uncertainty of  $\mu$  would overestimate the accuracy of the fit. The determined frequency-differences are shown in Fig. 16 alongside with the frequency-differences calculated as described in Section 2.1. The gray, dashed peak in the absorption spectrum could only be guessed by hand but not fitted (compare Fig. 14). Thus, it was not considered for the later analysis.



ы

**Fluorescence Peaks** 

Absorption Peaks

Saturation Peaks

2000

1000

×

×

×

0

Figure 16: Frequency-differences of the Rb peaks to the lowest-frequency-peak identified as  ${}^{87}\text{Rb}_{F=2}^{F=1}$ . The uncertainties were taken from the respective fit parameter  $\sigma$ . The expected frequency-differences were calculated, as explained in Section 2.1. The gray, dashed peak in the absorption spectrum could only be guessed by hand but not fitted. Comparing the experimental results to the expected frequency-differences, the transitions can be assigned to the peaks as listed in Table 2.

Frequency-Differences  $\delta v$  to <sup>87</sup>Rb F=1 [MHz]

\_\_\_

3000

4000

Expected 87-Rb Peak Positions

Expected 85-Rb Peak Positions

Expected D<sub>1</sub>-Line

5000

By comparing the experimental results to the calculated frequency-differences, the transitions can be assigned to the observed peaks as listed in Table 2.

For simplicity, the expected transitions are enumerated with  $T_1$  to  $T_6$  from left to right as pictured in Fig. 16. From Fig. 16, small systematic deviations to smaller calculated frequency-differences are visible. This raises the question whether the free spectral range of the FPI of 700 MHz is correct or whether it is larger or even inconsistent between the measurements.

Estimation of the Rb Composition of the Cell Additionally, the Rb composition of the cell can be estimated from the amplitude-ratio of two corresponding <sup>85</sup>Rb and <sup>87</sup>Rb peaks. These transitions with the same transition probabilities between <sup>85</sup>Rb and <sup>87</sup>Rb are ( $T_1$  and  $T_3$ ) and ( $T_2$  and  $T_4$ ) (compare Fig. 16 and Table 2. These are the ones

	Expected [MHz]	Saturation [MHz]	Absorption [MHz]	Fluorescence [MHz]
$T_1: {}^{87}\mathrm{Rb}_{F=2}^{F=1}$	0.0	$0 \pm 40$	$\int_{0+130}$	$0 \pm 400$
$T_2: {}^{87}\mathrm{Rb}_{F=2}^{F=2}$	814.5	$720\pm60$		$700\pm300$
$T_3: {}^{85}\mathrm{Rb}_{F=3}^{F=2}$	1518.6	$1410\pm30$	$1080\pm190$	$\begin{cases} 1500 + 500 \end{cases}$
$T_4: {}^{85}\mathrm{Rb}_{F=3}^{F=3}$	1880.2	$1740\pm40$	$1370\pm100$	
$T_5: {}^{85}\mathrm{Rb}_{F=2}^{F=2}$	4554.4	-	-	$\begin{cases} 4500 + 400 \end{cases}$
$T_6: {}^{85}\mathrm{Rb}_{F=2}^{F=3}$	4916.0	-	-	
$T_7: {}^{87}\mathrm{Rb}_{F=1}^{F=1}$	6834.7	-	-	-
$T_8: {}^{87}\mathrm{Rb}_{F=1}^{F=2}$	7649.2	-	-	-

Table 2: Frequency-differences of the Rb peaks to the lowest-frequency-peak  $T_1$  identified as  ${}^{87}\text{Rb}_{F=2}^{F=1}$ . The expected transitions have been calculated in Section 2.1 and Table 1 and are enumerated by  $T_1$  to  $T_8$ . The corresponding transitions are shown in the notation, introduced in Section 2.1. The results of the Rb spectra are estimated from the corresponding fit parameters  $\mu$  with the standard deviation  $\sigma$ . The data is shown in Fig. 16.

well-identified in the experiment.

Considering Fig. 16, the amplitude ratio can directly be calculated for the saturation spectroscopy results. By comparing the amplitudes, the percentage of  $^{87}$ Rb in the mixture can be calculated from

$$R_{87} = \frac{T_1}{T_1 + T_3} \tag{19}$$

and 
$$R_{87} = \frac{T_2}{T_2 + T_4}$$
, (20)

where  $T_1$  and  $T_3$  describe the amplitude of the corresponding absorption peak in the saturation spectrum. The uncertainties can be calculated by Gaussian error propagation.

For the absorption and the fluorescence spectrum, additional considerations have to be made. As visible in Figs. 14 and 16, the two peaks at  $T_2$  and  $T_3$  can be distinguished while the  $T_2$  transition can not be resolved. Looking into Fig. 14, one can estimate that the  $T_2$  transition is merged together with the  $T_1$  transition into one single peak. Thus, to get an estimate for the combined the amplitude ratio, the contributions of  $T_3$  and  $T_4$  have to be summed up as well. The percentage of <sup>87</sup>Rb in the mixture can then be calculated in this special case from

$$R_{87} = \frac{T_1 + T_2}{T_1 + T_2 + T_3 + T_4} \,. \tag{21}$$

With the same considerations one can see in the fluorescence spectrum that the contributions of  $T_3$  and  $T_4$  are merged into one single fluorescence peak. Thus, the amplitudes of  $T_1$  and  $T_2$  have to be added together to make get an estimate for the amplitude ratio. In this case, the percentage of <sup>87</sup>Rb in the mixture can be determined by

$$R_{87} = \frac{T_1 + T_2}{T_1 + T_2 + T_3 + T_4} \,. \tag{22}$$

The results for the percentage of <sup>87</sup>Rb and the corresponding uncertainties calculated using Gaussian error propagation are listed in Table 3.

Spectrum	Equation	Percentage of ${}^{87}\mathrm{Rb}$ in the Mixture [%]
Saturation	Eq. (19)	$37.9\pm0.5$
Saturation	Eq. (20)	$30.0 \pm 1.0$
Absorption	Eq. (21)	$37.7\pm0.7$
Fluorescence	Eq. (22)	$37.1 \pm 0.3$

Table 3: Percentage of  $^{87}$ Rb in the mixture, calculated with Eqs. (19) to (22) for the recorded spectra.

It can be seen that all results show good accordance with each other except for the evaluation of the  $T_2$  and  $T_4$  peaks in the saturation spectrum. It is possible that in this case some of the other peaks still influence the amplitude of the smaller absorption peaks. This could cause distortion in the fits and consequently result in a wrong value for the isotope ratio. Overall, the results are in the order of magnitude of the natural abundance of both isotopes, which is 27.2 % of <sup>87</sup>Rb [15].

**Calculation of the Hyperfine Structure Constant** Finally, the hyperfine structure constants of the  ${}^{2}P_{1/2}$ -state of  ${}^{85}$ Rb and  ${}^{87}$ Rb can be calculated from the frequencydifference between the two adjacent peaks of  $(T_1 \text{ and } T_2)$  and  $(T_3 \text{ and } T_4)$  (compare Fig. 16 and Table 2) As explained in Section 2.1, the calculation is given by

$$A = h \frac{|\delta \nu_{F_1} - \delta \nu_{F_2}|}{F_1 + 1}, \qquad (23)$$

where  $|\delta\nu_{F_1} - \delta\nu_{F_2}|$  is the frequency-difference between the two adjacent peaks from the  ${}^2P_{1/2}$ -state and  $F_1$  is the lower of the two quantum numbers F.

Unfortunately, the required frequency-differences can only be determined from the saturation spectroscopy measurement, as all transitions  $T_1$  to  $T_4$  are required. Using Eq. (23), the hyperfine structure constants of the  ${}^2P_{1/2}$ -state of  ${}^{85}$ Rb and  ${}^{87}$ Rb can be determined to

$$A_{87} = h \cdot (360 \pm 30) \text{ MHz}$$
  
 $A_{85} = h \cdot (110 \pm 13) \text{ MHz}$ .

These results are in good accordance with the literature values of  $A_{87, \text{ lit}} = h \cdot (407.25 \pm 0.63) \text{ MHz}$  and  $A_{85, \text{ lit}} = h \cdot (120.527 \pm 0.560) \text{ MHz}$ .

## 6. Summary and Discussion

#### 6.1. Results

In this experiment, the spectrum of a cell filled with the Rubidium isotopes  $^{85}$ Rb and  $^{87}$ Rb was investigated. This was done using three different spectroscopy methods: absorption spectroscopy, saturation spectroscopy and fluorescence spectroscopy. The width of the saturation peaks was about 70 % to 80 % smaller than those of the corresponding absorption peaks. This is as expected since the saturation peaks are not influenced by Doppler broadening any more. Thus, saturation spectroscopy also made it possible to separate peaks which were rather close in frequency. In contrast, in the absorption and fluorescence spectra, the widths of some of the peaks were so large that they overlapped and appeared as one peak. Even if the peaks could be separated, the peak positions could not be determined to the same precision as in the saturation measurement. Thus, in this experiment it could be confirmed that saturation spectroscopy is best suited for a precise measurement, especially since the setup has to be modified only slightly from the absorption setup. However, a lot fine-tuning of the setup is necessary before the saturation spectrum is visible.

The ratio of the isotopes in the Rubidium cell was calculated from each of the recorded spectra by comparing the peak amplitudes. The results are a ratio of <sup>87</sup>Rb in the mixture of  $(37.7 \pm 0.7)$ % and  $(37.1 \pm 0.3)$ % for the absorption and fluorescence measurement, respectively. Since more peaks could be identified in the saturation spectrum, it was possible to determine the ratio from two different peaks in the saturation spectrum as well. The resulting values for the ratio of <sup>87</sup>Rb in the mixture, determined from the saturation spectrum are  $(37.9 \pm 0.4)$ % and  $(29.2 \pm 1.0)$ %.

It is of note that one result from the saturation spectrum shows a large deviation from the other results. A possible reason for this is that the peaks used in this calculation had a lot lower amplitudes than the other peaks. In particular the absorption dips, which still are part of the saturation spectrum, were partially merged with absorption dips corresponding to other transitions. Since the height of the peaks is especially influenced by merging absorption dips, this could have influenced the fits and caused the observed deviation.

All other results for the ratio of  ${}^{87}$ Rb and  ${}^{85}$ Rb are within one standard deviation of each other, indicating a good agreement between spectroscopy methods. The ratio of  ${}^{87}$ Rb in naturally occurring Rubidium is 27.2 % [15]. This is in the same order of magnitude as the measured ratio in the Rubidium cell, which indicates that the result is reasonable although the exact composition of the cell is unknown.

In the second part of the analysis, the hyperfine structure constants of both Rubidium isotopes were determined. This was done using only the saturation measurement, since not enough transition peaks could be observed in the absorption and fluorescence spectra. The result for the hyperfine structure constant of  $^{85}$ Rb is

$$A_{85} = h \cdot (110 \pm 13) \,\mathrm{MHz}.$$

This shows a deviation from the literature value of  $h \cdot (120.527 \pm 0.056)$  MHz [14] by 0.78  $\sigma$  and a percentile deviation of roughly 8%. The measured hyperfine structure constant for <sup>87</sup>Rb is

$$A_{87} = h \cdot (360 \pm 30) \,\mathrm{MHz},$$

which differs from its literature value  $h \cdot (407.25 \pm 0.63)$  MHz [15] by  $1.8 \sigma$  and 12 %. Thus, both results are within good agreement to their respective literature value.

#### 6.2. Discussion

The measured ratio of the Rubidium isotopes agrees on all spectroscopy methods, which indicates a successful measurement and a correct identification of the observed transitions. Additionally, the hyperfine structure constants are in good agreement with their literature values. However, some problems were observed when executing the experiment and analyzing the data.

The most notable problem was in the frequency scan of the diode laser: The peaks in the spectrum of the Fabry-Pérot interferometer are not spaced equally in time (or channel, after being recorded on the oscilloscope) but are systematically shifted: Each peak distance is larger than the one to its right. This indicates that the diode laser did not scan over the frequencies in a linear manner. As a result, also the channel-tofrequency conversion is not linear but had to be performed by interpolation between neighboring peaks. As a result, this conversion can never be quite accurate, as it is a piecewise linear approximation of an apparently non-linear relation. If it had been possible to have a linear frequency scan, the transitions could have been determined to higher precision. Additionally, the scan was a lot faster when increasing the scan current compared to decreasing it. This is clearly visible by comparing the slope of the triangle caused by the scan (see Fig. 7b): The "up" slope is a lot shorter and flatter than the "down" slope, and the peaks in all spectra are spaced further apart in the "down" slope of the current. As a result, it was not possible to use both scans over the spectrum in the analysis, which could have improved the determination of the peak positions. However, overall, the effect of the uncertainty caused by the frequency scan is most probably negligible when compared to the uncertainties originating from the peak widths.

Another problem is evident when comparing the frequencies of the observed transitions to their literature values: All measured peaks are systematically shifted towards lower frequencies. No verifiable explanation was found for this effect. It is possible that this shift results from the frequency calibration: A systematic deviation could be explained if the actual free spectral range of the Fabry-Pérot interferometer was slightly larger than the one given by the instructions. In order to explain the deviation, more investigation is necessary. It would for example be insightful to determine the free spectral range of the Fabry-Pérot interferometer in an independent measurement, or repeat all measurements using a different Fabry-Pérot interferometer.

A further effect that was not considered in the measurement or the analysis was the influence of the angle and position of the components in the optical path. It was especially noticeable that the shape of the intensity-profile caused by the frequency scan was strongly dependent on the angle of the Rubidium cell with respect to the optical axis. It is quite possible that refraction in the glass of the cell caused frequency dependent intensity changes in the recorded spectra. This could also have influenced the results when determining the ratio of  $^{87}{\rm Rb}$ , since the ratio is calculated from the height of the peaks. However, it was not possible to further investigate this effect or consider it in the analysis.

## References

- TOPTICA Photonics AG. DL pro Digital Laser Controller Manual M-063 Version 16. Lochhamer Schlag 19, D-82166 Graefelfing, 2020.
- [2] TOPTICA Photonics AG. DL pro Grating Stabilized Diode Laser Head Manual M-30 Version 01. Lochhamer Schlag 19, D-82166 Graefelfing, 2011.
- [3] Maurizio Artoni. Electromagnetically Induced Transparency. 2016, pp. 1–9. DOI: 10.1016/B978-0-323-90800-9.00121-9.
- [4] M. Bajcsy, A. S. Zibrov, and M. D. Lukin. "Stationary pulses of light in an atomic medium" (2003). arXiv: QUANT-PH/0311092.
- [5] Clemens Baur. *Einrichtung des Versuches "Optisches Pumpen mit Laserdioden"*. Fakultät für Physik, Albert-Ludwigs-Universität Freiburg, 1997.
- [6] Wolfgang Demtröder. Laser Spectroscopy. Springer Berlin Heidelberg, 1996. DOI: 10.1007/978-3-662-08260-7.
- [7] Electromagnetically induced transparency, Master Lab. Hermann-Herder-Straße 3, 79104 Freiburg, Germany: Albert-Ludwigs-Universität Freiburg, Physikalisches Institut, 2023.
- [8] Michael Fleischhauer and Jonathan Marangos. "Fleischhauer, M., Imamolu, A. Marangos, J. P. Electromagnetically induced transparency: Optics in coherent media. Rev. Mod. Phys. 77, 633-673". *Reviews of Modern Physics REV MOD PHYS* 77 (2005), pp. 633–673. DOI: 10.1103/RevModPhys.77.633.
- [9] C. J. Foot. *Atomic Physics*. Oxford Master Series in Physics. Oxford University Press, 2005.
- [10] Wikimedia Commons Lcorman. Example of photodiode response in a saturated absorption experiment. Labels in French. Wavelength axis not perfectly calibrated. 2008.
- [11] R. Lechner et al. "Electromagnetically-induced-transparency ground-state cooling of long ion strings". *Physical Review A* 93.5 (2016). arXiv: 1603.05568.
- [12] Zhimin Shi et al. "Enhancing the spectral sensitivity of interferometers using slowlight media". Optics Letters 32.8 (2007), p. 915. DOI: 10.1364/ol.32.000915.
- [13] Marin Soljacic et al. "Electromagnetically induced transparency in microcavities" (2004). Ed. by Edward W. Taylor. DOI: 10.1117/12.562304. URL: http://dx. doi.org/10.1117/12.562304.
- [14] Daniel A. Steck. Rubidium 85 D Line Data. Revision 2.2.3, 9 July 2021. URL: http://steck.us/alkalidata.
- [15] Daniel A. Steck. Rubidium 87 D Line Data. Revision 2.2.2, 9 July 2021. URL: http://steck.us/alkalidata.

# A. Appendix

# A.1. Setup



Figure 17: Picture of the used setup in fluorescence spectroscopy mode.

Estimated Peak Positions with Uncertainties

Channels out of Mode Hop Free Range



### A.2. Saturation Spectrum - Large Range

Oscilloscope Spectrum of the FPI

Data after Underground Subtraction

. . .

Figure 18: Recorded saturation spectrum of the photo diode during a frequency-sweep of the laser in the saturation spectroscopy setup. The upper plot shows the FPI spectrum, recorded in parallel for calibration. The calibration has been performed as explained in Section 5.2 and is used to show both plots on a frequency-difference axis. The bottom plot shows the saturation spectrum after subtracting a linear background as explained in Section 5.3. It can be seen that also at higher frequencies around 5000 MHz, some peaks, caused by the rubidium are visible. Unfortunately, the data was not good enough to perform fits up to this range.



# A.3. Rb Spectra - Underground Considerations

Figure 19: Underground considerations for the saturation spectrum. The performed steps are shown to smoothen the data and subtract the linear underground as explained in Section 5.3.


Figure 20: Underground considerations for the absorption spectrum. The performed steps are shown to smoothen the data and subtract the linear underground as explained in Section 5.3.



Figure 21: Underground considerations for the fluorescence spectrum. The performed steps are shown to smoothen the data and subtract the linear underground as explained in Section 5.3.

Parameters	Peak 1	Peak 2	Peak 3	Peak 4
<i>A</i> 1 [mV]	$-2.95\pm0.02$	$-0.803 \pm 0.012$	$-4.83\pm0.04$	$-1.95\pm0.07$
$\mu_1  [\text{MHz}]$	$796.6  \pm 0.7 $	$1608 \pm 5$	$2141.7 \pm 0.6$	$2518.4 \pm 1.4$
$\sigma_1  [\text{MHz}]$	$7.5 \pm 1.0$	$76 \pm 4$	$65.1 \pm 0.6$	$96 \pm 2$
A2 [mV]	$0.65\pm0.03$	$0.653 \pm 0.017$	$0.77\pm0.05$	$0.68\pm0.07$
$\mu_2  [\text{MHz}]$	$771.7 \pm 1.2$	$1488.2 \pm 1.1$	$2178.8 \pm 1.1$	$2510 \pm 2$
$\sigma_2 [\mathrm{MHz}]$	$26.9 \pm 1.5$	$42.7 \pm 1.3$	$16.2 \pm 1.2$	$30 \pm 3$

### A.4. Fit Results

Table 4: Fit results of the saturation spectrum. The fits have been performed according to Eq. (14) and can be seen in Fig. 13.

Parameters	Peak 1	Peak 2	Peak 3
$\begin{array}{l} A \; [mV] \\ \mu \; [MHz] \\ \sigma \; [MHz] \end{array}$	$\begin{array}{r} -2.13 \pm 0.03 \\ 813.4 \ \pm 1.8 \\ 80 \ \pm 3 \end{array}$	$\begin{array}{c} -1.895 \pm 0.007 \\ 1893.8 \ \pm 0.9 \\ 52.3 \ \pm 1.5 \end{array}$	$\begin{array}{r} -1.632 \pm 0.011 \\ 2184.9 \ \pm 0.5 \\ 37.4 \ \pm 0.6 \end{array}$

Table 5: Fit results of the absorption spectrum. The fits have been performed according to Eq. (15) and can be seen in Fig. 14.

Parameters	Peak 1		Peak 2		Peak 3		Peak 4	
$A [\mathrm{mV}]$	1.23	$33 \pm 0.012$	0.70	$08 \pm 0.017$	3.28	$5\pm0.010$	1.96	$4 \pm 0.011$
$\mu$ [MHz]	715	$\pm 5$	1382	$\pm 7$	2202	$\pm 2$	5245	$\pm 2$
$\sigma$ [MHz]	254	$\pm 5$	175	$\pm 7$	52.2	$\pm 1.9$	279.2	$\pm 1.8$

Table 6: Fit results of the fluorescence spectrum. The fits have been performed according to Eq. (16) and can be seen in Fig. 15.



## A.5. Lab Notes

$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	aser set current EmA)	PD intensity oscillarcopa IVJ	03ti scale	set current [mA]	PD intersity	oszi scak
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	60	6.6 10-3 11	10 m V	88	n. 240	ZV
$\delta$ $\gamma_{0}$	4.2	7.0 11	u	49	1.32	4
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	62	7.0 //	11	90	1.80	5
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	6.7	74 4	4	31	204	4
$ \begin{array}{c} 6 & 5 & 7,8 & n & r & 3 & 3 & 7,8 & n & r \\ 6 & 6 & 7,8 & n & 6 & 7 & 5 & 7,50 & r \\ 6 & 8 & 8,6 & 8 & 6 & 9 & 6 & 9,83 & 7,8 \\ 6 & 9 & 4,6 & n & 7 & 5 & 7,50 & r \\ 6 & 9 & 4,6 & n & 7 & 5 & 7,50 & r \\ 7 & 7 & 7,0 & n & n & 3 & 2 & 5,56 & r \\ 7 & 7 & 7,0 & n & n & 3 & 8 & 3,88 & r \\ 7 & 1 & 7,4 & 8 & n & 7,0 & 1 & 4,68 & r \\ 7 & 1 & 7,4 & 8 & n & 1,0 & 1 & 4,68 & r \\ 7 & 1 & 7,4 & 8 & n & 1,0 & 1 & 4,68 & r \\ 7 & 1 & 7,4 & 1 & n & 1,0 & 1 & 4,68 & r \\ 7 & 1 & 1 & 7,4 & 1 & n & 1,0 & 1 & 4,68 & r \\ 7 & 1 & 1 & 1,0 & 1 & 1 & 4,68 & r \\ 7 & 1 & 1 & 1,0 & 1 & 1 & 1,0 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1,0 & 1 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1,0 & 1 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1,0 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1,0 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1,0 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1,0 & 1 & 1 & 0 & 1 \\ 7 & 1 & 1,0 & 1,0 & 1 & 1 & 0 & 0 \\ 8 & 1 & 1,0 & 1 & 1 & 1 & 0 & 0 \\ 8 & 1 & 1,0 & 1 & 1 & 1 & 0 & 0 \\ 8 & 1 & 1,0 & 1 & 1 & 1 & 0 & 0 \\ 8 & 1 & 1,0 & 1 & 1 & 0 & 0 & 0 \\ 8 & 1 & 1,0 & 1 & 1 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 1 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 1 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 1 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1,0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & 0 & 0 & 0 \\ 8 & 1,0 & 1 & 0 & $	6.	24 11	<u>ā</u>	92	244	
6 + 1, 5 + 1, 6 + 1, 6 + 7, 5 + 3, 10, 6 + 1, 6 + 7, 5 + 3, 10, 6 + 1, 6 + 7, 5 + 3, 10, 6 + 1, 10, 10, 10, 10, 10, 10, 10, 10, 10,	65	28 11	1	93	6,44	G
6 x       3 x       y       y       y       y       y       y       y         6 x       8 x       x       4       y <t< td=""><td>11</td><td>7,0</td><td>ti .</td><td>94</td><td>4817</td><td>. 9</td></t<>	11	7,0	ti .	94	4817	. 9
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	6 8	47 11	6	9 6	3,00	6
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	18	21	6	91	11/1/1 327	
3 - 2 $3 - 2$ $4 - 2$	6 9	8.6 11		9 2	2 -1	6
1       3       3       3       3       1         2       1	20	70 1	n .	30	200	
7, 2 $3, 4$ $n$ $4, 0, 0$ $4, 52$ $4$ $7, 3$ $5, 7$ $n$ $n$ $1, 0, 2$ $4, 52$ $4$ $7, 4$ $4, 2$ $n$ $n$ $1, 0, 2$ $4, 72$ $4$ $7, 5$ $40, 2$ $n$ $n$ $1, 0, 2$ $4, 72$ $4$ $7, 5$ $40, 6$ $a$ $n$ $1, 0, 2$ $5, 80$ $6$ $7, 8$ $40, 0, 4$ $a$ $n$ $0, 2$ $5, 82$ $n$ $7, 8$ $40, 0, 4$ $a$ $1, 0, 2$ $5, 82$ $n$ $7, 8$ $40, 0, 4$ $a$ $1, 0, 2$ $5, 82$ $n$ $8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$	3	94 11	6	99	500	6
$23$ $33$ $4$ $n$ $A = 0$ $4 > 2$ $4 > 6$ $7 > 4$ $4_0 > 2$ $4$ $n$ $1 > 2$ $4_1 > 6$ $n$ $7 > 5$ $4_0 , 6$ $n$ $n > 3$ $5_1 > 68$ $n$ $7 > 5$ $4_0 , 6$ $n$ $n > 3$ $5_1 > 68$ $n$ $7 > 5$ $4_0 , 6$ $n$ $n > 5$ $5_1 > 10$ $n$ $7 > 7$ $A_1 , 6$ $n$ $n > 7$ $5_1 > 10$ $n$ $7 > 7$ $A_1 , 6$ $n$ $n > 7$ $5_1 > 10$ $n$ $7 > 7$ $A_1 , 6$ $n$ $n > 7$ $5_1 > 10$ $n$ $7 > 7$ $A_1 , 6$ $n$ $n > 7$ $5_1 > 10$ $n$ $8 > 1 > 10 > 10 > 10 > 10 > 10 > 10 > 10$	2 2	54 0			9100	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	73	9.9 4		100	4,54	
7 $7$	7/24			101	4,68	
7       1	7 7	10,6 "		102	4,76	6
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	75	10,2 1	9	10 J	7108	1
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	20	10,0 1		109	5,72	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	20			105	5,40	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	7.0	11,0 4		100	5,72	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	+)	119,4		107	5,72	9
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	80	AL,C I		700	6,72	6
8       2 $42,6$ 7 $42,6$ 7 $42,6$ 7 $42,6$ $6,684$ $6$ 8       3 $43,0$ $6$ $12$ $6,76$ $4$ 8       4 $42,4$ $6$ $12$ $6,76$ $4$ 8       5 $43,8$ $6$ $12$ $6,76$ $4$ 8       6 $44,2$ $11$ $14$ $7,46$ $7$ 8       6 $13,8$ $6$ $13,72$ $7$ $7$ 8       6 $13,8$ $11$ $14$ $7,46$ $7$ 8       6 $13,72$ $7,16$ $7$ $7,16$ $7$ 8       6 $12,72$ $7$ $7,16$ $7$ $7,16$ $7$ 8       6 $12,72$ $7,16$	81	12,6 1,	<i>b</i>	× 0 )	6,52	
$3 \rightarrow$ $43, 0$ $n \rightarrow 1$ $6, 34$ $a$ $3 + 4$ $n3, 4$ $n \rightarrow 1$ $6, 76$ $a$ $3 + 4$ $n3, 3$ $n \rightarrow 1$ $5, 76$ $a$ $3 + 4$ $n3, 3$ $n \rightarrow 1$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 1$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 1$ $7, 16$ $a$ $4 + 6$ $n \rightarrow 1$ $7, 16$ $a$ $4 + 6$ $n \rightarrow 1$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 1$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $7, 16$ $a$ $3 + 6$ $n \rightarrow 2$ $n \rightarrow 2$ $3, 128$ $a$ $3 + 6$ $n \rightarrow 2$ $n \rightarrow 2$ $3, 128$ $a$ $3 + 6$ $n \rightarrow 2$ $n \rightarrow 2$ <td>8 2</td> <td>12,6 4</td> <td>/</td> <td>110</td> <td>6 (68</td> <td>4</td>	8 2	12,6 4	/	110	6 (68	4
3       4 $A_1 L$ 6,76       4         8       5       13,8       6       6       7,16       7         8       5       13,8       6       6       7,16       7         8       6 $A_{12}$ 7       1       1       1       7       7         8       6 $A_{12}$ 7       10       7       7       7         8       6 $A_{2}$ 7       100       7       7       7         8       6 $A_{2}$ 7       100       7       7       7         8       6       2 $A_{2}$ 7       7       7       7       7         8       6       2 $A_{2}$ 7       7       7       8       7         8       6       2 $A_{2}$ 7       7       1       9       9       20       7         8       6       2 $A_{2}$ 7       1       1       9       9       20       7         8       6       5       2       0       7       1       1       9       1	8 3	13,0 1		111	6,84	
$g, g, g, h$ $A_3, g, h$ $h$	84	A3,4 //		1 1 2	6,76	4
2       6 $Aq_2^2$ 7       7       7 $q_4^2$	85	13,8 /		1 1 2	7,16	4
$3^{2}$ $1$ $1$ $5$ $7,5^{5}$ $7,7^{2}$	8.6	14,2 7	- H	1 1 4	7,48	11
$3'$ $6_{10}$ $AZ$ $Y$ $ZO0 = V$ $A = A_{10}$ $7, 72$ $77$ $3'$ $6_{11}$ $AZ$ $7$ $7$ $7$ $2, 80$ $7$ $3'$ $6_{12}$ $AZ$ $4'$ $n'$ $1 = A_{10}$ $2, 80$ $n'$ $3'$ $6_{12}$ $AZ$ $4'$ $n'$ $1 = A_{10}$ $2, 80$ $n'$ $3'$ $6_{12}$ $AZ$ $4'$ $n'$ $1 = A_{10}$ $2, 80$ $n'$ $3'$ $6_{12}$ $AZ$ $i'$ $i'$ $i'$ $i'$ $i''$ $i''''''''''''''''''''''''''''''''''''$	- aren		24	1 1 5	7,56	4
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	86,0	NZ 9	ZOOmV	116	2,72	
g = 6, 2 $Az = 4$ $n'' = 1, n'' = 1, n''' = 1, n'''$	86,1	12 1	17	117	2,60	11
8       6 $\Lambda$ <	\$62	12 4		1 1 7	7,96	
86.44 $A2$ $7$ $7$ $100$ $8,28$ $*$ $86.5$ $20$ $7$ $7$ $121$ $8,44$ $7$ $86.6$ $20$ $7$ $7$ $121$ $8,44$ $7$ $86.6$ $20$ $7$ $7$ $122$ $8,60$ $7$ $86.6$ $20$ $7$ $7$ $724$ $8,42$ $7$ $86.9$ $20$ $7$ $724$ $8,42$ $7$ $86.9$ $20$ $7$ $728$ $8,74$ $7$ $86.9$ $20$ $7$ $728$ $8,74$ $7$ $877$ $700$ $725$ $9,76$ $7$ $877$ $120$ $728$ $9,40$ $7$ $87,3$ $340$ $728$ $7,56$ $7$ $87,5$ $336$ $732$ $9,64$ $7$ $87,5$ $396$ $7$ $79,94$ $7$ $87,5$ $396$ $7$ $79,94$ $7$ $87,5$ $568$ <td< td=""><td>86,4</td><td>12 1</td><td></td><td>1 1 9</td><td>\$,20</td><td>11</td></td<>	86,4	12 1		1 1 9	\$,20	11
8 6.5 $20$ $7$ $721$ $8,44$ $7$ $8 6.6$ $20$ $7$ $721$ $8,44$ $7$ $8 6.6$ $20$ $7$ $721$ $8,44$ $7$ $8 6.7$ $20$ $7$ $722$ $8,76$ $7$ $8 6.7$ $20$ $7$ $722$ $8,76$ $7$ $8 6.9$ $20$ $7$ $729$ $8,76$ $7$ $8 6.9$ $20$ $7$ $729$ $8,76$ $7$ $8 7.7$ $20$ $7$ $725$ $9,76$ $7$ $877.7$ $120$ $725$ $9,76$ $7$ $740$ $728$ $7,244$ $7$ $877.7$ $120$ $728$ $7,40$ $7$ $8$ $7,40$ $7$ $87.7$ $268$ $728$ $7,40$ $7$ $8$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$ $7$	86.4	N2 1	9	1 20	8,28	"
8       6.6       7.0       7       7       7       7       8       6.60       7         8       6       7       7       7       7       7       8       7         8       6       7       7       7       7       7       7       7         8       6       3       7       7       7       7       7       7         8       6       3       7       7       7       7       7       7       7         8       7       7       7       120       7	86,5	201	4	121	8,44	9
8       6       7       2       5 $g_1 p_1^2$ $f_1^2$ 8       6       7       2       7       7       7       2 $f_1^2$ 8       6       3       2       6       7       7       7       2 $g_1 p_2^2$ $f_1^2$ 8       6       3       2       6       7       7       7       7       7       7       7       7       7       1       20 $f_1^2$ 7 $f_1^2$ <	86,6	20 /	4	122	8,60	F
8 6 . 7       20       7 $20$ 7 $20$ 7         8 6 . 9       20       7 $125$ $9.16$ 7         8 7.0 $140$ 7 $125$ $9.16$ 7         8 7.0 $140$ 7 $125$ $9.16$ 7         8 7.0 $140$ 7 $25$ $9.16$ 7         8 7.0 $140$ 7 $28$ $9.10$ 7         8 7.1 $268$ 7 $28$ $9.40$ 7         8 7.3 $340$ 7 $29$ $9.56$ 7         8 7.4 $396$ $4$ $728$ $9.44$ 7         8 7.5 $436$ $4$ $730$ $9.64$ 7         8 7.5 $436$ $4$ $730$ $9.44$ 6         8 7.5 $436$ $4$ $732$ $10.044$ 7         8 7.6 $568$ $4$ $733$ $10.044$ 7         8 7.6 $566$ $4$ $733$ $10.044$ 7         8 7.6 $566$ $7$	867	20 1		123	8,76	8
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	86,0	20 4		n 2 q	8,92	9
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	06,5	20 4		125	9,16	<i>n</i>
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	870	140 4		126	9,16	<i>Ŷ</i>
$87 \cdot 2$ $268 \cdot 4$ $728$ $940 \cdot 5$ $87,3$ $340 \cdot 6$ $729$ $756 \cdot 5$ $7$ $87,7$ $396 \cdot 4$ $729$ $756 \cdot 5$ $7$ $87,5$ $396 \cdot 4$ $730$ $964$ $6$ $87,5$ $396 \cdot 4$ $730$ $964$ $6$ $87,5$ $396 \cdot 4$ $730$ $964$ $6$ $87,5$ $366 \cdot 4$ $733$ $7960$ $7$ $87,6$ $568 \cdot 4$ $733$ $7960$ $7$ $87,6$ $568 \cdot 4$ $733$ $7004$ $6$ $87,7$ $636 \cdot 4$ $733$ $7004$ $6$ $47,8$ $636 \cdot 4$ $7335$ $70,04$ $6$ $82,9$ $744 \cdot 4$ $735$ $70,28$ $7$ $82,9$ $744 \cdot 4$ $735$ $70,28$ $7$ $82,9$ $800$ $7038$ $70,28$ $7$ $738$ $70,28$ $7$ $738$ $70,28$ $7$	87.1	180 4		127	J, 24	1
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	8712	268 6		.7 2 8	9,40	9
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	87,3	340 6		729	9,56	9
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	87,4	396 4		130	9,64	9
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	87,5	496 4		1 3 1	2,80	4
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	87,6	568 1		132	10,04	9
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	87,7	636 1		Λ 3 3	10,04	G
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	67.8	636 1		134	10,12	4 .
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	82,9	744 4		135	10,20	9
1 3 2 10 28 4 7 3 8 10,44 17	88,0	800 11		136	10,28	9
1 3 8 10,44 1				132	10,28	4
				738	10,44	9
				140	10 400	6



Pictures chr	PD ch2:	Etalon ch:	s: ARBARY Butgut A.	From withy ch (4)	Trigger
s can treguency :	10 Hr			2 	
Br.					
File nound	scan offset	scan amplitule	set current	feed forward	notes
OP1 NF1	27,0	8,2	164.0	-0,81	300l
Q82 NFL	27,3	10,3	166,6	-1,11	ff tos low
ars NFS	27,3	10,3	166,6	-0,11	ft the high
OPH NF4	273	10,3	166,6	+1,810	Af too high, more hops
QPS NF5	21,n	10,3	172,2	-0,81	adjust seen officet
QP6 NF6	22,2	10,3	171,3	-0,81	- FP peaks world -
2P7 NF2	21,7	4,00	171,7	-0,81	adjust scan applitude
QPS NF8	21,7	25,9	171,7	-0,82	- Different and of FP pontes
QP9 NF9	79,83	35,59	1578	-0,47	RB parts scour range -
QPTO NE 10	78 47	40,5	156,72	-0,678	T=2020 T=23
QPIN NEIN	18,91	40,5	155,57	-0,649	T=2022 Fine-tui
apaz NFAZ	13,68	40,5	153,97	-0,796	T=24,22 with Rb
QPA3 PF13	R	le .	le	9	in reference withert Rb
QPA4 NFA4	- U	1	· ·	Le n	" with Rb
GPAS NEAS	11,38	23,8	155,58	-1.122	T=20,22 LithRL
GP16 NFA6	er	<i>a</i> .	te -	4	reference without Rh
OPAZ NEAZ	12,88	13/3 13,8	156,10	-1002	ref
QP18 NF18	1	li li	G	4	with RS
SPAY NEAS	17,88	13,8	156 10	- 1 002	WE W/0 Rb T=2072
0870 NEZA	6	1	1		111 Rb
GPZA NEZA	4	11	11	-4	The second
BP22 NED	Ir .	11	11	11	RS W/O NUT
NET NET		6	~	5	Re with NO.1
OPTH NETH	F F		1		
OPIE NETE	- 4	6	4		· · · · · · · · · · · · · · · · · · ·
OP26 NEX	01	a	· Ir		
MODIT NOT			U.		Pet. Hr 0.P25 WIO ND
ADDA WOOD	q b	U	C		batman W/ONPALA
OCCO NECO	1			ę	with VDA
UPCO NEZO		<u>(</u>		· · ·	with ND 1
UTSU NF3)	7	1.	1		Net Sur QP23 WIO NON
2P31 NF31	11,48	15,8	156,50	-1,002	with K-L
QUSZ NF32	11,18	20,10	156,38	- 01846	
QP35 NF35	8	q	9 .	4	
CUISY NESY	F	Å	·	<i>(</i> /	
6135 NP35		C'	01	C1	
OP36 NF36	61	<i>a</i>	6	a	
0137 NF37	And 11,18	20,070	156,38	-0,902	ver after w/o NDA loss
2P38 NF38	t	4	6	I.c.	
6P39 NF39	(r	4	7		
QPUO NF40	9	2	· · ·	î.	
QPYN NF41	ie .	4	60	(-	with ND1
dezz NF42	k	ę	ę	G.	ref. for 41 w/D N
-					
0P.47 NF47	10,28	20,10	157.70	-0,902	with NP1
OP 48 NEWS	ne .	le.	C/	CT	4
QP49 NF49	C/	4	G .	(	1
0.950 N=50	4	8	a	· · ·	WEF. FUT US W/O NOT
ORSA NESA	4	8		ę .	NF-5-42 a a
		10		1	······································
OPSZ NESI	4	9	2	2	n n us p-
QPSZ NESI		7	2	2	u a as p-

FI. waren			PPD			
· cyorescer	DD spectrosco	Ry. all s				
MONEY	P W WEST TO	o Ro cont	L KS			
P.S.		e., (/ )		A	1.01.0.1	
Gere n	ane .	An 28	in any tode	Setensent	teed forward	note
aras	NEAN	20100	(0,10 r	157,70	-0,30	
aper	Altre	4		7.74		
0045	NF43	10,68	20,10	ASTITS	-0,30	no light
51-16	N1 46					
						-
_						
pino to di	14					
Filo no	me	scan offset sc	in applitude	set current	feed former	wike
0.753	NF53	10,28	20,10	157,70	-0,902	Supermith Reb
Q.P54	MP54	Y	cr	e -	4	) with sof Rb cell- m
			1			
	2					
				Wenters	chen.	
						1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
				OR.	324	
					.	
		· ·				
						- ****b_2
1.2						
	-					

# B. Python Code

#### **Transition Energies**

```
1 # -*- coding: utf-8 -*-
2
3 import numpy as np
4 import mymodules.usefultools as mmu
5
6
7 # %%
8
_9 # Transitions and MHz distances to 87 Rb 1 -> 2
11 transitions_85 = np.array( # MHz
12
      Г
          dline_85 + 1.7708439228e3 + 150.659, # 85 Rb P 1/2, F=3 <--> S
13
      1/2, F=2 # 4915 MHz
          dline_85 + 1.7708439228e3 - 210.923, # 85 Rb P 1/2, F=2 <--> S
14
     1/2, F=2 # 4554 MHz
          dline_85 - 1.2648885163e3 + 150.659, # 85 Rb P 1/2, F=3 <--> S
15
      1/2, F=3 # 1880 MHz
          dline_85 - 1.2648885163e3 - 210.923, # 85 Rb P 1/2, F=2 <--> S
16
      1/2, F=3 # 1518 MHz
17
      ٦
18)
19 dline_87 = 377.107463308e6 # MHz
20 transitions_87 = np.array( # MHz
21
      Г
          dline_87 + 4.271676631815181e3 + 305.44, # 87 Rb P 1/2, F=2 <-->
22
      S 1/2, F=1 # 7649 MHz
          dline_87 + 4.271676631815181e3 - 509.06, # 87 Rb P 1/2, F=1 <-->
23
       S 1/2, F=1 # 6834 MHz
          dline_87 - 2.563005979089109e3 + 305.44, # 87 Rb P 1/2, F=2 <-->
24
      S 1/2, F=2 # 814 MHz
          dline_87 - 2.563005979089109e3 - 509.06, # 87 Rb P 1/2, F=1 <-->
25
      S 1/2, F=2 # 0 MHz
      ]
26
27 )
28
29
30 # distances_85 = transitions_85[0] - transitions_85
31 # distances_87 = transitions_87[0] - transitions_87
32
33 # dist_etalon_85 = distances_85 / 700
_{34} \# dist_etalon_87 = distances_87 / 700
35
36 # print("distances in etalon 85:", dist_etalon_85)
37 # print("distances in etalon 87:", dist_etalon_87)
38 # print("distances 85:", distances_85)
39 # print("distances 87:", distances_87)
40
_{41} # these are the distances of the transition frequencies to the Rb87
_{42} # S_1/2 F=2 to P_1/2 F=1 transition
```

FP-II

```
43 distances_all_to_lowest = np.array([transitions_85 - transitions_87[3],
     transitions_87 - transitions_87[3]])
44 # S_1/2 F=1 to P_1/2 F=2 transition
45 distances_all_to_highest = np.array([transitions_85 - transitions_87[0],
      transitions_87 - transitions_87[0]])
46
47 # print(transitions_85)
48 # print(transitions_87[2])
49 # print("min:", np.min(transitions_87))
50 # print("min:", np.min(transitions_85))
51
52 # print(distances_all)
53 dist_etalon_all_to_lowest = distances_all_to_lowest / 700
54 dist_etalon_all_to_highest = distances_all_to_highest / 700
55 # print(transitions_85)
56 # print(transitions_87)
57 # print(abs(transitions_85 - dline_85))
58 # print(abs(transitions_87 - dline_87))
59
60 print("distances to the lowest-frequency peak in etalon:\n",
      dist_etalon_all_to_lowest)
61 print(list(np.sort(dist_etalon_all_to_lowest.flatten() * 700)))
62 print ("distances to the highest-frequency peak in etalon:\n",
      dist_etalon_all_to_highest)
63 print(list(np.sort(abs(dist_etalon_all_to_highest.flatten() * 700))))
64
65
66 # %%
67
68 print({"85": distances_all_to_lowest[0], "87": distances_all_to_lowest
      [1]})
69 mmu.save_json({"85": distances_all_to_lowest[0], "87":
      distances_all_to_lowest[1]}, "rubidium_peaks")
```

#### Photodiode Calibration

```
1 # -*- coding: utf-8 -*-
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 # import scipy
7 # import scipy.odr as s_odr
8 # import mymodules.usefultools as mmu
9 # import mymodules.calculate as mmc
10 # import mymodules.measure as mmm
11 # import mymodules.optimize as mmo
12 # import mymodules.functions as mmf
13 import mymodules.plot as mmp
14
15 # verbose = True
16 # si_format = False
17 # plot = True
```

```
18 # draft = False
19 save_images = False
20 # write_data = False
21
22
23 # %%
24
25 data = pd.read_csv("../data/laser_photodiod_calibration.csv", header=0,
      names=["set_current", "intensity"])
26
27
28 # %%
29
30 # fig, ax = mmp.make_fig(grid=True, inset=[0.07, 0.55, 0.4, 0.4])
31 fig, ax = mmp.make_fig(grid=True, inset=[0.6, 0.08, 0.37, 0.37],
      inset_origin=[86, -0.05, 88, 0.8])
32
33 x = data["set_current"]
34 y = data["intensity"]
35
36 mmp.plot(ax[0], x, y, config="scatter", label="Data points")
37
38 x = data["set_current"][(data["set_current"] > 86) & (data["set_current"]
       < 88)]
39 y = data["intensity"][(data["set_current"] > 86) & (data["set_current"] <</pre>
       88)]
40 mmp.plot(ax[1], x, y, config="scatter")
41
42 mmp.plot(ax[0], 87, None, x_err=0.1, config="vspan", color="darkgreen",
      label="Estimated Lasing Threshold")
43 # ax[0].axvline(87, color="darkgreen", label="Estimated Laser Threshold")
44 # ax[1].axvline(87, color="darkgreen")
45 mmp.plot(ax[1], 87, None, x_err=0.1, config="vspan", color="darkgreen")
46
47 ax[0].set_xlim(60, 140)
48 ax[0].set_ylim(-0.4, 11)
49 ax[0].set_title("Lasing Threshold Determination")
50 ax[0].set_xlabel("Set Current [mA]")
51 ax[0].set_ylabel("Photo Diode Intensity [V]")
52
53 # ax[1].set_xlim(86, 88)
54 # ax[1].set_ylim(-0.05, 0.8)
55 # ax[1].set_xlabel("Set Current [mA]")
56 # ax[1].set_ylabel("Photo Diode Intenisty [V]")
57
58 mmp.legend(ax[0], loc=2)
59
60 if save_images:
      mmp.save_fig(fig, path="../report/figures", name="Laser Threshold
61
      Determination", extension="pdf")
```

```
Spectroscopy
```

1 # -\*- coding: utf-8 -\*-

```
2
3 import numpy as np
4 import pandas as pd
5 # import matplotlib.pyplot as plt
6 import matplotlib as mpl
7 # import scipy
8 # import scipy.odr as s_odr
9 import mymodules.usefultools as mmu
10 import mymodules.calculate as mmc
11 # import mymodules.measure as mmm
12 # import mymodules.optimize as mmo
13 import mymodules.functions as mmf
14 import mymodules.plot as mmp
15
16 # verbose = True
17 # si_format = False
18 # plot = True
19 # draft = False
20 save_images = False
21 # write_data = False
22
23
24 # %%
25
26 # saturation gaussian with constant term
27 def saturation_gauss(x,
                        A1=1, mu1=0, sigma1=1,
28
29
                        A2=1, mu2=0, sigma2=1):
      x = np.array(x)
30
      return A1 * np.exp(-(x - mu1)**2 / (2 * sigma1**2)) +\
31
          A2 * np.exp(-(x - mu2)**2 / (2 * sigma2**2))
32
33
34
35 saturation_gauss = mmf.fit_function(
36
      f=saturation_gauss,
      bounds=([-np.inf, -np.inf, 0, 0, -np.inf, 0], [0, np.inf, np.inf, np.
37
      inf, np.inf, np.inf]),
      params=["A1", "mu1", "sigma1", "A2", "mu2", "sigma2"],
38
39)
40
41
42 # %%
43
44 spectrum = {46: "Fluorescence", 47: "Saturation", 48: "Saturation", 51: "
      Absorption"}
45
46 F = 700 # MHz
47 F_err = F / 300 \# MHz
48
49
50 def ch_to_etalon(channel, channel_err, etalon_ch):
    if channel < etalon_ch[0]:</pre>
51
          etalon_ch -= etalon_ch[0]
52
```

```
assert isinstance(channel, float), "No float given"
       x = np.interp(channel, etalon_ch, [i * F for i in range(0, len(
54
      etalon_ch))])
       x_err = np.interp(channel + channel_err, etalon_ch, [i * F for i in
55
      range(0, len(etalon_ch))]) - x
56
       return x, x_err
57
58
59 # %%
60
61 file = 47 # saturation
62
63 data = pd.read_csv(f"../data/oszi/NewFile{file}.csv", header=1, names=["x
      ", "CH1", "CH2", "CH3", "CH4", ""])
64
65 \text{ fp}_x = \text{data}["x"]
66 \text{ fp}_y = \text{data}["CH2"] * 1e3
67
68 \text{ fp}_x_max_num = []
69 \text{ fp}_x_max_used = \{33: (7, 15),
                     45: (11, 22),
70
71
                     46: (1, 12),
                     47: (1, 8),
72
                     48: (2, 14),
73
74
                     50: (8, 16),
75
                     51: (0, 6),
                     52: (1, 12)}
76
77 fp_x_max = list(fp_x[fp_y \ge 2])
78 fp_y_max = list(fp_y[fp_y \ge 2])
79
80 # Find peaks n calibration etalon spectrum
81 appended = False
82 for i in range(len(fp_x_max)):
       if appended is False:
83
           fp_x_max_num.append(1)
84
85
           appended = True
86
       if i < len(fp_x_max) - 1:
87
           if fp_x_max[i] - 5 <= fp_x_max[i + 1] <= fp_x_max[i] + 5:</pre>
88
                fp_x_max_num[-1] += 1
89
90
           else:
91
                appended = False
92
93
94 assert np.sum(fp_x_max_num) == len(fp_x_max), f"{file}: wrong length"
95
96 fp_x_max_f = []
97 for i in range(len(fp_x_max_num)):
       if file in fp_x_max_used.keys():
98
           if fp_x_max_used[file][0] <= i <= fp_x_max_used[file][1]:</pre>
99
                fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i
100
      ])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
       else:
```

```
fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i])):
      int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])))
104 # calibration etalon uncertainty
105 fp_x_max_f_err = [5] * len(fp_x_max_f)
106
107 fig, ax = mmp.make_fig(grid=True, figsize=(9, 5))
108
109 # Plot lines in calibration spectrum
110 mmp.plot(ax, fp_x, fp_y, config="plot", label="Oscilloscope Spectrum of
      the FPI")
111 for i in range(len(fp_x_max_f)):
112
       # ax.text(fp_x_max_f[i] + 2, 3.3, i)
       mmp.plot(ax, fp_x_max_f[i], None, x_err=fp_x_max_f_err[i], config="
113
      vspan", color="tab:red")
114
      if i < len(fp_x_max_f) - 1:</pre>
115
           ax.text(np.mean([fp_x_max_f[i], fp_x_max_f[i + 1]]), 2.01, fr"$\
      Delta\,$ch = {round(fp_x_max_f[i + 1] - fp_x_max_f[i])}",
      horizontalalignment="center")
116
117 mmp.add_to_legend(ax, "Estimated Peak Positions with Uncertainties",
      linestyle="-", color="tab:red")
118
119 ax.set_title("Fabry-Perot Interferometer (FPI) Calibration")
120 ax.set_ylabel(r"Amplitude $A$ [mV]")
121 ax.set_xlabel("Channel ch [a.u.]")
123 ax.set_ylim(-0.9, 3.8)
124 ax.set_xlim(0, 1200)
125 mmp.legend(ax, loc=9, ncol=2, bbox_to_anchor=(0.5, -0.12))
126
127 if save_images:
      mmp.save_fig(fig, path="../report/figures", name="FPI Calibration",
128
      extension="pdf")
129
130
131 # %%
132
133 fig, ax = mmp.make_fig(grid=True, figsize=(9, 5))
134
135 # Plot lines in calibration spectrum
136 mmp.plot(ax, fp_x, fp_y, config="plot", label="Oscilloscope Spectrum of
      the FPI")
137 for i in range(len(fp_x_max_f)):
       # ax.text(fp_x_max_f[i] + 2, 3.3, i)
138
       mmp.plot(ax, fp_x_max_f[i], None, x_err=fp_x_max_f_err[i], config="
139
      vspan", color="tab:red")
140
141 mmp.add_to_legend(ax, "Estimated Peak Positions with Uncertainties",
      linestyle="-", color="tab:red")
142
143 ax.set_title("Fabry-Perot Interferometer (FPI) Calibration on Frequency
      Axis")
144 ax.set_ylabel(r"Amplitude $A$ [mV]")
```

```
145 ax.set_xlabel(r"Frequency-Differences $\delta\nu$ to Lowest FPI Peak [MHz
      1")
146
147 ax.set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f))])
148 minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0, len(
       fp_x_max_f), len(fp_x_max_f) * 4 + 1), np.linspace(0, len(fp_x_max_f)
       - 1, len(fp_x_max_f)), fp_x_max_f))
149 ax.xaxis.set_minor_locator(minor_locator)
150
151 ax.set_ylim(-0.9, 3.8)
152 ax.set_xlim(0, 1200)
153 mmp.legend(ax, loc=9, ncol=2, bbox_to_anchor=(0.5, -0.12))
154
155 if save_images:
156
      mmp.save_fig(fig, path="../report/figures", name="FPI Calibration
       Frequency", extension="pdf")
157
158
159 # %%
160
161 file = 47 # saturation
162
163 data = pd.read_csv(f"../data/oszi/NewFile{file}.csv", header=1, names=["x
       ", "CH1", "CH2", "CH3", "CH4", ""])
164
165 # Apply running average to correct for double line
166 ch1, x = mmc.running_average(data["CH1"], n=11, x=data["x"])
167
168 fig, ax = mmp.make_fig(ncols=1, nrows=2, grid=True, figsize=(9, 3),
       sharex=True, gridspec_kw={'height_ratios': [0.5, 1], 'hspace': 0})
169
170 xx = np.linspace(0, 1200, 400)
171
172 fp_x = data["x"]
173 fp_y = data["CH2"] * 1e3
174 \text{ pd}_x = x
175 \text{ pd}_y = \text{ch1} * 1e3
176
177 fp_x_max_num = []
178 \text{ fp}_x_max_used = \{33: (7, 15),
                     45: (11, 22),
179
                     46: (1, 12),
180
                     47: (1, 8),
181
                     48: (2, 14),
182
                     50: (8, 16),
183
                     51: (0, 6),
184
                     52: (1, 12)
185
186 fp_x_max = list(fp_x[fp_y \ge 2])
187 fp_y_max = list(fp_y[fp_y \ge 2])
188
189 # Find peaks n calibration etalon spectrum
190 appended = False
191 for i in range(len(fp_x_max)):
      if appended is False:
192
```

```
FP-II
```

fp\_x\_max\_num.append(1)

```
appended = True
194
195
       if i < len(fp_x_max) - 1:
196
           if fp_x_max[i] - 5 <= fp_x_max[i + 1] <= fp_x_max[i] + 5:</pre>
197
                fp_x_max_num[-1] += 1
198
199
200
           else:
201
                appended = False
202
203 assert np.sum(fp_x_max_num) == len(fp_x_max), f"{file}: wrong length"
204
205 \, fp_x_max_f = []
206 for i in range(len(fp_x_max_num)):
207
       if file in fp_x_max_used.keys():
208
           if fp_x_max_used[file][0] <= i <= fp_x_max_used[file][1]:</pre>
209
                fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i
      ])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
       else:
210
211
           fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i])):
      int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
212
213 # calibration etalon uncertainty
214 fp_x_max_f_err = [5] * len(fp_x_max_f)
215
216 # Plot lines in calibration spectrum
217 mmp.plot(ax[0], fp_x, fp_y, config="plot", label="Oscilloscope Spectrum
      of the FPI")
218 for i in range(len(fp_x_max_f)):
219
       # ax[0].text(fp_x_max_f[i] + 2, 3.3, i)
       mmp.plot(ax[0], fp_x_max_f[i], None, x_err=fp_x_max_f_err[i], config=
220
       "vspan", color="tab:red")
221
222 mmp.add_to_legend(ax[0], "Estimated Peak Positions with Uncertainties",
      linestyle="-", color="tab:red")
223
224 mmp.plot(ax[1], data["x"], data["CH1"] * 1e3, config="scatter", s=0.5,
      color="tab:orange")
225 mmp.add_to_legend(ax[1], "Oscilloscope Spectrum of the Photo Diode",
      linestyle="dotted", color="tab:orange", lw=3.5)
226 mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5, color="tab:green")
227 mmp.add_to_legend(ax[1], "Data after Running Average", linestyle="dotted"
    , color="tab:green", lw=3.5)
228
229 ax[0].set_title(f"{spectrum[file]} Spectrum - Running Average")
230 ax[0].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f))])
231 ax[1].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f))])
232 minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0, len(
      fp_x_max_f), len(fp_x_max_f) * 4 + 1), np.linspace(0, len(fp_x_max_f)
       - 1, len(fp_x_max_f)), fp_x_max_f))
233 ax[0].xaxis.set_minor_locator(minor_locator)
234 ax[1].xaxis.set_minor_locator(minor_locator)
235 ax[1].set_xlabel(r"Frequency-Differences $\delta\nu$ to Lowest FPI Peak [
      MHz]")
```

```
236 # ax[1].set_xlabel("Channel ch [a.u.]")
237 ax[0].set_ylabel(r"Amplitude $A$ [mV]")
238 ax[1].set_ylabel(r"Amplitude $A$ [mV]")
239
240 mmp.legend(ax[0], loc=9, ncol=2, bbox_to_anchor=(0.5, -2.3))
241 mmp.legend(ax[1], loc=9, ncol=2, bbox_to_anchor=(0.5, -0.28))
242
243 ax[0].set_ylim(-0.9, 3.8)
244 ax[1].set_xlim(0, 1200)
245
246 if save_images:
       mmp.save_fig(fig, path="../report/figures", name="Saturation Spectrum")
247
       Running Average", extension="pdf")
248
249
250 # %%
251
252 file = 47 # saturation
253
254 data = pd.read_csv(f"../data/oszi/NewFile{file}.csv", header=1, names=["x
       ", "CH1", "CH2", "CH3", "CH4", ""])
255
256 # Apply running average to correct for double line
257 ch1, x = mmc.running_average(data["CH1"], n=11, x=data["x"])
258
259 fig, ax = mmp.make_fig(ncols=1, nrows=2, grid=True, figsize=(9, 3),
       sharex=True, gridspec_kw={'height_ratios': [0.5, 1], 'hspace': 0})
260
261 \text{ xx} = \text{np.linspace}(0, 1200, 400)
262
263 \text{ fp}_x = \text{data}["x"]
264 fp_y = data["CH2"] * 1e3
265 \text{ pd}_x = x
266 pd_y = ch1 * 1e3
267
268 \text{ fp}_x_max_num = []
269 \text{ fp}_x_max_used = \{33: (7, 15),
270
                      45: (11, 22),
                      46: (1, 12),
271
                      47: (1, 8),
272
                      48: (2, 14),
273
                      50: (8, 16),
274
                      51: (0, 6),
275
                      52: (1, 12)
276
277 fp_x_max = list(fp_x[fp_y \ge 2])
278 fp_y_max = list(fp_y[fp_y \ge 2])
279
280 # Find peaks n calibration etalon spectrum
281 appended = False
282 for i in range(len(fp_x_max)):
       if appended is False:
283
           fp_x_max_num.append(1)
284
            appended = True
285
286
```

```
if i < len(fp_x_max) - 1:
287
           if fp_x_max[i] - 5 <= fp_x_max[i + 1] <= fp_x_max[i] + 5:</pre>
288
289
                fp_x_max_num[-1] += 1
290
           else:
291
292
               appended = False
293
294 assert np.sum(fp_x_max_num) == len(fp_x_max), f"{file}: wrong length"
295
296 \, fp_x_max_f = []
297 for i in range(len(fp_x_max_num)):
       if file in fp_x_max_used.keys():
298
299
           if fp_x_max_used[file][0] <= i <= fp_x_max_used[file][1]:</pre>
                fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i
300
       ])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
301
       else:
302
            fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i])):
       int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])))
303
304 # calibration etalon uncertainty
305 \text{ fp}_x_max_f_err = [5] * len(fp_x_max_f)
306
307 # Plot lines in calibration spectrum
308 mmp.plot(ax[0], fp_x, fp_y, config="plot", label="Oscilloscope Spectrum
      of the FPI")
309 for i in range(len(fp_x_max_f)):
       # ax[0].text(fp_x_max_f[i] + 2, 3.3, i)
310
       mmp.plot(ax[0], fp_x_max_f[i], None, x_err=fp_x_max_f_err[i], config=
311
       "vspan", color="tab:red")
312
313 mmp.add_to_legend(ax[0], "Estimated Peak Positions with Uncertainties",
       linestyle="-", color="tab:red")
314
315 # mmp.plot(ax[1], data["x"], data["CH1"] * 1e3, config="scatter", s=0.5,
      color="tab:orange")
316 # mmp.add_to_legend(ax[1], "Oscilloscope Spectrum of the Photo Diode",
      linestyle="dotted", color="tab:orange", lw=3.5)
317 mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5, color="tab:green")
318 mmp.add_to_legend(ax[1], "Data after Running Average", linestyle="dotted"
       , color="tab:green", lw=3.5)
319
320 # finding peak positions for absorption spectroscopy
321 # Correct for linear underground
322 slope_1_x = pd_x[:130]
323 slope_l_y = pd_y[:130]
324 \text{ slope_r_x = pd_x[840:]}
325 \text{ slope}_r_y = pd_y[840:]
326 mmp.plot(ax[1], slope_l_x, slope_l_y, color="tab:red", s=0.5, config="
       scatter")
327 mmp.plot(ax[1], slope_r_x, slope_r_y, color="tab:red", s=0.5, config="
      scatter")
328 mmp.add_to_legend(ax[1], "Data Used for the Fit", linestyle="dotted",
       color="tab:red", lw=3.5)
```

```
329
```

```
330 out_linear_underground = mmp.fit(mmf.poly_1,
                                    np.concatenate([slope_l_x, slope_r_x]),
331
332
                                    np.concatenate([slope_l_y, slope_r_y]),
333
                                    ax=ax[1],
                                    color="tab:red",
334
                                    label="Linear Underground Fit")
335
336
337 pd_y = pd_y - mmf.poly_1(pd_x, *out_linear_underground[0])
338 mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5, color="tab:blue")
339 mmp.add_to_legend(ax[1], "Data after Underground Subtraction", linestyle=
      "dotted", color="tab:blue", lw=3.5)
340
341 ax[0].set_title(f"{spectrum[file]} Spectrum - Underground Subtraction")
342 ax[0].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f))])
343 ax[1].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f))])
344 minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0, len(
      fp_x_max_f), len(fp_x_max_f) * 4 + 1), np.linspace(0, len(fp_x_max_f)
      - 1, len(fp_x_max_f)), fp_x_max_f))
345 ax[0].xaxis.set_minor_locator(minor_locator)
346 ax[1].xaxis.set_minor_locator(minor_locator)
347 ax[1].set_xlabel(r"Frequency-Differences $\delta\nu$ to Lowest FPI Peak [
      MHz]")
348 # ax[1].set_xlabel("Channel ch [a.u.]")
349 ax[0].set_ylabel(r"Amplitude $A$ [mV]")
350 ax[1].set_ylabel(r"Amplitude $A$ [mV]")
351
352 ax[0].set_ylim(-0.9, 3.8)
353 ax[1].set_xlim(0, 1200)
354
355 mmp.legend(ax[0], loc=9, ncol=2, bbox_to_anchor=(0.5, -2.3))
356 mmp.legend(ax[1], loc=9, ncol=2, bbox_to_anchor=(0.5, -0.28))
357
358 if save_images:
      mmp.save_fig(fig, path="../report/figures", name="Saturation Spectrum")
359
       Underground Subtraction", extension="pdf")
360
361
362 # %%
363
364 show_all = False
365
366 fit_results = {}
367
_{368} # flourescence: 45, 46
369 # absorption: 50, 51, 52
370 # saturation: 35, 47, 48
371
33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44]
373
374 # files = [47] # saturation
375 # files = [51] # absorption
376 # files = [46] # fluorescence
```

```
378 for file in files:
       data = pd.read_csv(f"../data/oszi/NewFile{file}.csv", header=1, names
379
       =["x", "CH1", "CH2", "CH3", "CH4", ""])
380
381
       # Apply running average to correct for double line
       ch1, x = mmc.running_average(data["CH1"], n=11, x=data["x"])
382
383
       fig, ax = mmp.make_fig(ncols=1, nrows=2, grid=True, figsize=(9, 3),
384
       sharex=True, gridspec_kw={'height_ratios': [0.5, 1], 'hspace': 0})
385
       xx = np.linspace(-50, 1200, 800)
386
387
388
       fp_x = data["x"]
389
       fp_y = data["CH2"] * 1e3
390
       pd_x = x
391
       pd_y = ch1 * 1e3
392
393
       fp_x_max_num = []
394
       fp_x_max_used = \{33: (7, 15),
                          45: (11, 22),
395
                          46: (1, 12),
396
                          47: (1, 8),
397
                          48: (2, 14),
398
399
                          50: (8, 16),
400
                          51: (0, 6),
                          52: (1, 12)
401
       fp_x_max = list(fp_x[fp_y \ge 2])
402
       fp_y_max = list(fp_y[fp_y \ge 2])
403
404
       # Find peaks n calibration etalon spectrum
405
       appended = False
406
       for i in range(len(fp_x_max)):
407
           if appended is False:
408
                fp_x_max_num.append(1)
409
410
                appended = True
411
           if i < len(fp_x_max) - 1:
412
                if fp_x_max[i] - 5 \le fp_x_max[i + 1] \le fp_x_max[i] + 5:
413
                    fp_x_max_num[-1] += 1
414
415
                else:
416
                    appended = False
417
418
       assert np.sum(fp_x_max_num) == len(fp_x_max), f"{file}: wrong length"
419
420
       fp_x_max_f = []
421
       for i in range(len(fp_x_max_num)):
422
           if file in fp_x_max_used.keys():
423
                if fp_x_max_used[file][0] <= i <= fp_x_max_used[file][1]:</pre>
424
                    fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(
425
       fp_x_max_num[:i])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
           else:
426
```

```
fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i
427
      ])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
428
       # calibration etalon uncertainty
429
       fp_x_max_f_err = [5] * len(fp_x_max_f)
430
431
432
       # Calculate peak distances and ch to frequency by averaging
433
       # peak_dist = np.mean([fp_x_max_f[i + 1] - fp_x_max_f[i] for i in
      range(len(fp_x_max_f) - 1)])
       # peak_dist_err = np.std([fp_x_max_f[i + 1] - fp_x_max_f[i] for i in
434
      range(len(fp_x_max_f) - 1)], ddof=1)
       # print(peak_dist, peak_dist_err)
435
436
       # ch_to_frequency = F / peak_dist
437
       # ch_to_frequency_err = np.sqrt((F_err / peak_dist)**2 + (F /
438
      peak_dist**2 * peak_dist_err)**2)
       # print("Channel to frequency [MHz]:", ch_to_frequency,
439
      ch_to_frequency_err)
440
441
       # Plot lines in calibration spectrum
       mmp.plot(ax[0], fp_x, fp_y, config="plot", label="Oscilloscope
442
      Spectrum of the FPI")
       for i in range(len(fp_x_max_f)):
443
           # ax[0].text(fp_x_max_f[i] + 2, 3.3, i)
444
           mmp.plot(ax[0], fp_x_max_f[i], None, x_err=fp_x_max_f_err[i],
445
      config="vspan", color="tab:red")
446
       mmp.add_to_legend(ax[0], "Estimated Peak Positions with Uncertainties
447
      ", linestyle="-", color="tab:red")
448
       fit_results[file] = {}
449
       fit_results[file]["fp"] = fp_x_max_f
450
451
       fit_results[file]["fp_err"] = fp_x_max_f_err
452
       if show_all:
453
454
           mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5, color="tab:
      green")
           mmp.plot(ax[1], data["x"], data["CH1"] * 1e3, config="scatter", s
455
      =0.5, color="tab:orange")
           mmp.add_to_legend(ax[1], "Oscilloscope Spectrum of the Photo
456
      Diode", linestyle="dotted", color="tab:orange", lw=3.5)
           mmp.add_to_legend(ax[1], "Data after Running Average", linestyle=
457
      "dotted", color="tab:green", lw=3.5)
458
       # finding peak positions for saturation spectroscopy
459
       if file == 47:
460
           # Correct for linear underground
461
           slope_1_x = pd_x[:130]
462
           slope_l_y = pd_y[:130]
463
           slope_r_x = pd_x[840:]
464
           slope_r_y = pd_y[840:]
465
466
           if show_all:
               mmp.plot(ax[1], slope_l_x, slope_l_y, color="tab:red", s=0.5,
467
       config="scatter")
```

```
mmp.plot(ax[1], slope_r_x, slope_r_y, color="tab:red", s=0.5,
468
        config="scatter")
                mmp.add_to_legend(ax[1], "Data Used for the Fit", linestyle="
469
       dotted", color="tab:red", lw=3.5)
470
471
           if show_all:
472
                out_linear_underground = mmp.fit(mmf.poly_1,
473
                                                   np.concatenate([slope_1_x,
       slope_r_x]),
                                                   np.concatenate([slope_l_y,
474
       slope_r_y]),
                                                   ax=ax[1],
475
476
                                                   x_range=xx,
                                                   color="tab:red",
477
478
                                                   label="Linear Underground")
479
            else:
480
                out_linear_underground = mmp.fit(mmf.poly_1,
                                                   np.concatenate([slope_l_x,
481
       slope_r_x]),
482
                                                   np.concatenate([slope_l_y,
       slope_r_y]),
                                                   # ax=ax[1],
483
                                                   color="tab:red")
484
485
486
           pd_y = pd_y - mmf.poly_1(pd_x, *out_linear_underground[0])
487
           mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5, color="tab:
       blue")
           mmp.add_to_legend(ax[1], "Data after Underground Subtraction",
488
       linestyle="dotted", color="tab:blue", lw=3.5)
489
           if show_all is False:
490
                out_s_1 = mmp.fit(saturation_gauss,
491
492
                                   pd_x[210:330], pd_y[210:330],
                                   ax=ax[1],
493
                                   p0=[-3, 270, 40, 0.4, 260, 10],
494
495
                                   color="tab:red",
                                   bounds=True,
496
                                   verbose=False,
497
                                   label="")
498
                mmp.plot(ax[1], xx, saturation_gauss(xx, *out_s_1[0][0:3], 0,
499
        0, 1), color="tab:blue")
                mmp.plot(ax[1], xx, saturation_gauss(xx, 0, 0, 1, *out_s_1
500
       [0][3:6]), color="tab:orange")
501
                out_s_2 = mmp.fit(saturation_gauss,
502
                                   pd_x[370:535], pd_y[370:535],
503
                                   ax=ax[1],
504
                                   p0=[-3, 550, 10, 0.4, 450, 10],
505
                                   bounds=True,
506
                                   # show_values=True,
507
                                   color="tab:red",
508
                                   label="")
509
                mmp.plot(ax[1], xx, saturation_gauss(xx, *out_s_2[0][0:3], 0,
510
        0, 1), color="tab:blue")
```

```
mmp.plot(ax[1], xx, saturation_gauss(xx, 0, 0, 1, *out_s_2
511
       [0][3:6]), color="tab:orange")
512
               out_s_3 = mmp.fit(saturation_gauss,
513
                                  pd_x[610:665], pd_y[610:665],
514
515
                                  ax=ax[1],
                                  p0=[-5, 635, 40, 0.5, 650, 10],
516
517
                                  bounds=True,
                                  color="tab:red",
518
                                  label="")
519
               mmp.plot(ax[1], xx, saturation_gauss(xx, *out_s_3[0][0:3], 0,
520
       0, 1), color="tab:blue")
               mmp.plot(ax[1], xx, saturation_gauss(xx, 0, 0, 1, *out_s_3
521
       [0][3:6]), color="tab:orange")
522
523
               out_s_4 = mmp.fit(saturation_gauss,
524
                                  pd_x[690:790], pd_y[690:790],
                                  ax=ax[1],
525
                                  p0=[-3, 740, 20, 0.8, 740, 5],
526
527
                                  bounds=True,
                                  color="tab:red",
528
                                  label="Saturation Gauss Fits")
529
               mmp.plot(ax[1], xx, saturation_gauss(xx, *out_s_4[0][0:3], 0,
530
       0, 1), color="tab:blue")
               mmp.plot(ax[1], xx, saturation_gauss(xx, 0, 0, 1, *out_s_4
531
       [0][3:6]), color="tab:orange")
               # to_frequency = [ch_to_etalon(out_s_1[0][i], out_s_1[1][i],
533
       fp_x_max_f)[0] for i in range(len(out_s_1[0]))]
               # to_frequency_err = [ch_to_etalon(out_s_1[0][i], out_s_1[1][
534
      i], fp_x_max_f)[1] for i in range(len(out_s_1[0]))]
               # mmu.print_to_table(saturation_gauss.params_tex, "merge[]",
535
       ["[mV]", "[MHz]", "[MHz]", "[mV]", "[MHz]", "[MHz]"],
               #
                                     mmu.sc_round(out_s_1[0], out_s_1[1], SI=
536
      True),
               #
                                     mmu.sc_round(to_frequency,
537
       to_frequency_err, SI=True)
               #
                                      )
538
               # to_frequency = [ch_to_etalon(out_s_2[0][i], out_s_2[1][i],
539
       fp_x_max_f)[0] for i in range(len(out_s_2[0]))]
               # to_frequency_err = [ch_to_etalon(out_s_2[0][i], out_s_2[1][
540
      i], fp_x_max_f)[1] for i in range(len(out_s_2[0]))]
               # mmu.print_to_table(saturation_gauss.params_tex, "merge[]",
541
       ["[mV]", "[MHz]", "[MHz]", "[mV]", "[MHz]", "[MHz]"],
               #
                                     mmu.sc_round(out_s_2[0], out_s_2[1], SI=
542
      True),
                                     mmu.sc_round(to_frequency,
543
               #
       to_frequency_err, SI=True)
                                     )
544
               #
               # to_frequency = [ch_to_etalon(out_s_3[0][i], out_s_3[1][i],
545
      fp_x_max_f)[0] for i in range(len(out_s_3[0]))]
               # to_frequency_err = [ch_to_etalon(out_s_3[0][i], out_s_3[1][
546
      i], fp_x_max_f)[1] for i in range(len(out_s_3[0]))]
```

```
# mmu.print_to_table(saturation_gauss.params_tex, "merge[]",
547
       ["[mV]", "[MHz]", "[MHz]", "[mV]", "[MHz]", "[MHz]"],
                                      mmu.sc_round(out_s_3[0], out_s_3[1], SI=
548
      True),
                                      mmu.sc_round(to_frequency,
549
               #
      to_frequency_err, SI=True)
550
                                      )
               # to_frequency = [ch_to_etalon(out_s_4[0][i], out_s_4[1][i],
       fp_x_max_f)[0] for i in range(len(out_s_4[0]))]
                # to_frequency_err = [ch_to_etalon(out_s_4[0][i], out_s_4[1][
552
      i], fp_x_max_f)[1] for i in range(len(out_s_4[0]))]
               # mmu.print_to_table(saturation_gauss.params_tex, "merge[]",
553
       ["[mV]", "[MHz]", "[MHz]", "[mV]", "[MHz]", "[MHz]"],
                                      mmu.sc_round(out_s_4[0], out_s_4[1], SI=
554
      True),
555
               #
                                      mmu.sc_round(to_frequency,
      to_frequency_err, SI=True)
               #
                                      )
556
557
               mmp.add_to_legend(ax[1], "Absorption Peaks", linestyle="-",
558
      color="tab:blue")
               mmp.add_to_legend(ax[1], "Saturation Peaks", linestyle="-",
559
      color="tab:orange")
560
               fit_results[file]["fits"] = [out_s_1, out_s_2, out_s_3,
561
      out_s_4]
562
       # finding peak positions for flourescence spectroscopy
563
       if file == 46:
564
           slope_1_x = pd_x[5:35]
565
           slope_1_y = pd_y[5:35]
566
           slope_2_x = pd_x[390:500]
567
           slope_2_y = pd_y[390:500]
568
           slope_3_x = pd_x [675:730]
569
           slope_3_y = pd_y[675:730]
570
571
           if show_all:
572
               mmp.plot(ax[1], slope_1_x, slope_1_y, color="tab:red", s=0.5,
573
        config="scatter")
               mmp.plot(ax[1], slope_2_x, slope_2_y, color="tab:red", s=0.5,
574
        config="scatter")
               mmp.plot(ax[1], slope_3_x, slope_3_y, color="tab:red", s=0.5,
575
        config="scatter")
               mmp.add_to_legend(ax[1], "Data Used for the Fit", linestyle="
576
      dotted", color="tab:red", lw=3.5)
577
           if show_all:
578
               out_linear_underground = mmp.fit(mmf.poly_1,
579
                                                  np.concatenate([slope_1_x,
580
      slope_2_x, slope_3_x]),
                                                  np.concatenate([slope_1_y,
581
      slope_2_y, slope_3_y]),
                                                  ax=ax[1],
582
                                                  color="tab:red",
583
```

x\_range=xx, 584 label="Linear Underground") 585586else: out\_linear\_underground = mmp.fit(mmf.poly\_1, 587 np.concatenate([slope\_1\_x, 588slope\_2\_x, slope\_3\_x]), 589 np.concatenate([slope\_1\_y, slope\_2\_y, slope\_3\_y]), # ax=ax[1], 590 color="tab:red") 591 592 pd\_y = pd\_y - mmf.poly\_1(pd\_x, \*out\_linear\_underground[0]) 593 mmp.plot(ax[1], pd\_x, pd\_y, config="scatter", s=0.5) 594mmp.add\_to\_legend(ax[1], "Data after Underground Subtraction", 595linestyle="dotted", color="tab:blue", lw=3.5) 596 597 if show\_all is False: out\_f = mmp.fit(mmf.sum\_of\_n(mmf.gauss, n=4), 598 pd\_x[50:680], pd\_y[50:680], 599 600 ax=ax[1], p0=[1, 100, 50, 601 1, 180, 20, 602 1, 300, 50, 603 1, 600, 50], 604 605 bounds=([ 606 0, -np.inf, 0, 0, -np.inf, 0, 607 0, -np.inf, 0, 608 0, -np.inf, 0 609 ],[ 610 np.inf, np.inf, np.inf, 611 np.inf, np.inf, np.inf, 612 613 np.inf, np.inf, np.inf, np.inf, np.inf, np.inf 614 ]), 615 616 color="tab:red", label="") 617 mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_f[0][:3]), color="tab: 618 orange") mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_f[0][3:6]), color="tab 619 :orange") mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_f[0][6:9]), color="tab 620 :orange") mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_f[0][9:]), color="tab: 621 orange") 622 # to\_frequency = [ch\_to\_etalon(out\_f[0][:3][i], out\_f[1][:3][ 623 i], fp\_x\_max\_f)[0] for i in range(len(out\_f[0][:3]))] # to\_frequency\_err = [ch\_to\_etalon(out\_f[0][:3][i], out\_f 624 [1][:3][i], fp\_x\_max\_f)[1] for i in range(len(out\_f[0][:3]))] # mmu.print\_to\_table(mmf.gauss.params\_tex, "merge[]", ["[mV 625 ]", "[MHz]", "[MHz]"], mmu.sc\_round(out\_f[0][:3], out\_f[1][:3], 626 SI=True),

627	# mm	<pre>1.sc_round(to_frequency,</pre>
	<pre>to_frequency_err , SI=True)</pre>	
628	# )	
629	# to frequency = [ch to	etalon(out f[0][3:6][i], out f
	[1][3:6][i], fp x max f) $[0]$ for	i in range (len (out $f[0][3:6])$ )]
620	# to frequency err = [c]	$f = \frac{1}{2} $
030	$\frac{1}{1} \begin{bmatrix} 2 \cdot 6 \end{bmatrix} \begin{bmatrix} i \end{bmatrix}  \text{fn } x \text{ max } f \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 2 \cdot 6 \end{bmatrix} \begin{bmatrix} i \end{bmatrix}$	$\frac{1}{100} = \frac{1000}{100} \left( \frac{1000}{100} + \frac{1000}{100} \right) \left[ \frac{1000}{100} + \frac{1000}{100} \right]$
001	[1][3.0][1], 1P_x_max_1/[1] 101	f manage (ren(out_r[o][3.0]))]
631	# mmu.print_to_table(mm.	gauss.params_tex, merge[], [ [mv
	]", "[MHZ]", "[MHZ]"], "	
632		1.sc_round(out_f[0][3:6], out_f
	[1][3:6], SI=True),	
633	#	1.sc_round(to_frequency,
	<pre>to_frequency_err , SI=True)</pre>	
634	# )	
635	<pre># to_frequency = [ch_to]</pre>	_etalon(out_f[0][6:9][i], out_f
	<pre>[1][6:9][i], fp_x_max_f)[0] for</pre>	<pre>i in range(len(out_f[0][6:9]))]</pre>
636	<pre># to_frequency_err = [c]</pre>	h_to_etalon(out_f[0][6:9][i], out_f
	<pre>[1][6:9][i], fp_x_max_f)[1] for</pre>	<pre>i in range(len(out_f[0][6:9]))]</pre>
637	<pre># mmu.print_to_table(mm:</pre>	f.gauss.params_tex, "merge[]", ["[mV
	]", "[MHz]", "[MHz]"],	
638	#	1.sc round(out f[0][6:9], out f
	[1][6:9], SI=True),	
639	#	1.sc round(to frequency.
	to frequency err. SI=True)	
640	#	
641	# to frequency = $\begin{bmatrix} ch & to \end{bmatrix}$	etalon(out f[0][9:][i]. out f[1][9:][
	i] fp x max f) [0] for i in range	e(len(out f[0][9·]))]
642		h to etalon(out $f[0][9:][i]$ , out f
012	$\begin{bmatrix} 1 \\ 9 \end{bmatrix} \begin{bmatrix} 9 \end{bmatrix} \begin{bmatrix} 1 \\ 1 $	in range (len (out $f[0][9:1])$ ]
643	# mmu print to table(mm	f gauss params tex "merge[]" ["[mV
010	]" "[MHz]" "[MHz]"]	
644	± mmi	$s_{1}$ sc round(out f[0][9.] out f[1][9.]
044	SI=True)	1.50_10una(0u0_1[0][0.], 0u0_1[1][0.],
645	# mm	sc round (to frequency
040	$\pi$ mutual to frequency orr $SI=True$ )	1. Sc_round(to_rrequency;
C A C	#	
647	π )	
047	mmn add to logond (av [1]	"Fluerescence Peaks" linestyle="-"
048	<pre>mmp.add_t0_tegend(ax[1] color="tob.orange")</pre>	, Findlescence reaks , finestyle ,
0.40	color - tab: orange )	
649		r = r = 1400 = 920
650	mmp.plot(ax[0], 1400, N)	She, $x_{err} = 1400 - 830$ , $config = "vspan"$ ,
	color="tab:orange")	
651	mmp.plot(ax[1], 1400, N	one, x_err=1400 - 830, config="vspan",
	color="tab:orange")	
652	<pre>ax[1].fill_between([0,</pre>	1], [0, 0], [0, 0], color="tab:orange"
	, label="Channels out of Mode Ho	<pre>p Free Range", alpha=0.2)</pre>
653		_
654	fit_results[file]["fits	"] = out_f
655		
656	<pre># finding peak positions for ab</pre>	sorption spectroscopy
657	if file == 51:	
658	$slope_1_x = pd_x[40:140]$	
659	$slope_1_y = pd_y[40:140]$	
660	slope_2_x = pd_x[380:440]	

```
slope_2_y = pd_y[380:440]
661
           slope_3_x = pd_x[800:1200]
662
663
           slope_3_y = pd_y[800:1200]
           if show_all:
664
                mmp.plot(ax[1], slope_1_x, slope_1_y, color="tab:red", s=0.5,
665
        config="scatter")
666
                mmp.plot(ax[1], slope_2_x, slope_2_y, color="tab:red", s=0.5,
        config="scatter")
                mmp.plot(ax[1], slope_3_x, slope_3_y, color="tab:red", s=0.5,
667
        config="scatter")
                mmp.add_to_legend(ax[1], "Data Used for the Fit", linestyle="
668
       dotted", color="tab:red", lw=3.5)
669
670
           if show_all:
671
                out_linear_underground = mmp.fit(mmf.poly_1,
672
                                                   np.concatenate([slope_1_x,
       slope_2_x, slope_3_x]),
                                                   np.concatenate([slope_1_y,
673
       slope_2_y, slope_3_y]),
674
                                                   ax=ax[1],
675
                                                   x_range=xx,
                                                   color="tab:red",
676
                                                   label="Linear Underground")
677
678
            else:
679
                out_linear_underground = mmp.fit(mmf.poly_1,
680
                                                   np.concatenate([slope_1_x,
       slope_2_x, slope_3_x]),
                                                   np.concatenate([slope_1_y,
681
       slope_2_y, slope_3_y]),
                                                   # ax=ax[1], x_range=xx,
682
                                                   color="tab:red")
683
684
685
           pd_y = pd_y - mmf.poly_1(pd_x, *out_linear_underground[0])
           mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5)
686
           mmp.add_to_legend(ax[1], "Data after Underground Subtraction",
687
       linestyle="dotted", color="tab:blue", lw=3.5)
688
           if show_all is False:
689
                out_a_1 = mmp.fit(mmf.gauss,
690
                                   pd_x[250:295], pd_y[250:295],
691
                                   ax=ax[1],
692
                                   color="tab:red",
693
                                   # show_values=True,
694
                                   bounds=True,
695
                                   verbose=False,
696
                                   label="")
697
                mmp.plot(ax[1], xx, mmf.gauss(xx, *out_a_1[0]), color="tab:
698
       orange")
699
                out_a_2 = mmp.fit(mmf.gauss,
700
                                   pd_x[520:610], pd_y[520:610],
701
                                   ax=ax[1],
702
                                   bounds=True.
703
                                   # show_values=True,
704
```

color="tab:red", 705 label="") 706 707 mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_a\_2[0]), color="tab: orange") 708 709 out\_a\_3 = mmp.fit(mmf.gauss, pd\_x[635:660], pd\_y[635:660], 710 711 ax=ax[1], 712 bounds=True, # show\_values=True, 713 color="tab:red", 714 label="") 715 716 mmp.plot(ax[1], xx, mmf.gauss(xx, \*out\_a\_3[0]), color="tab: orange") 717 718 # to\_frequency = [ch\_to\_etalon(out\_a\_1[0][i], out\_a\_1[1][i], fp\_x\_max\_f)[0] for i in range(len(out\_a\_1[0]))] # to\_frequency\_err = [ch\_to\_etalon(out\_a\_1[0][i], out\_a\_1[1][ 719 i], fp\_x\_max\_f)[1] for i in range(len(out\_a\_1[0]))] 720 # mmu.print\_to\_table(mmf.gauss.params\_tex, "merge[]", ["[mV ]", "[MHz]", "[MHz]"], # mmu.sc\_round(out\_a\_1[0], out\_a\_1[1], SI= 721 True), mmu.sc\_round(to\_frequency, to\_frequency\_err, SI=True) 723 # ) # to\_frequency = [ch\_to\_etalon(out\_a\_2[0][i], out\_a\_2[1][i], 724 fp\_x\_max\_f)[0] for i in range(len(out\_a\_2[0]))] # to\_frequency\_err = [ch\_to\_etalon(out\_a\_2[0][i], out\_a\_2[1][ 725i], fp\_x\_max\_f)[1] for i in range(len(out\_a\_2[0]))] # mmu.print\_to\_table(mmf.gauss.params\_tex, "merge[]", ["[mV 726 ]", "[MHz]", "[MHz]"], # mmu.sc\_round(out\_a\_2[0], out\_a\_2[1], SI= 727 True), # mmu.sc\_round(to\_frequency, 728 to\_frequency\_err, SI=True) # ) 729 # to\_frequency = [ch\_to\_etalon(out\_a\_3[0][i], out\_a\_3[1][i], 730 fp\_x\_max\_f)[0] for i in range(len(out\_a\_3[0]))] # to\_frequency\_err = [ch\_to\_etalon(out\_a\_3[0][i], out\_a\_3[1][ 731 i], fp\_x\_max\_f)[1] for i in range(len(out\_a\_3[0]))] # mmu.print\_to\_table(mmf.gauss.params\_tex, "merge[]", ["[mV 732 ]", "[MHz]", "[MHz]"], # mmu.sc\_round(out\_a\_3[0], out\_a\_3[1], SI= 733 True), mmu.sc\_round(to\_frequency, 734 # to\_frequency\_err, SI=True) 735 # 736 mmp.add\_to\_legend(ax[1], "Absorption Peaks", linestyle="-", 737 color="tab:orange") 738 fit\_results[file]["fits"] = [out\_a\_1, out\_a\_2, out\_a\_3] 739 740

```
# for ab in [0.0, 814.499999999999, 1518.6364627480507,
741
       1880.2184627056122\,,\ 4554.368901848793\,,\ 4915.9509018063545\,,
       6834.682610869409, 7649.182610869408]:
             # for ab in [0.0, 814.499999999999, 2733.231709063053,
742
      3094.8137090206146, 5768.9641481637955, 6130.546148121358,
       6834.682610869409, 7649.182610869408]:
             x_off = {33: -2,}
743
       #
                       45: 2,
744
       #
                       46: 2.
745
       #
                       47: 20.
       #
746
                       48: 6,
       #
747
       #
                       50: 10,
748
749
       #
                       51: 25,
750
       #
                       52: 12}
       #
             x_ch = np.interp(ab, [i * F for i in range(-1, len(fp_x_max_f)
751
       - 1)], np.array(fp_x_max_f)) + x_off[file]
752
       #
             ax[1].axvline(x_ch, color="tab:green")
753
754
755
       if show_all is False:
           ax[0].set_title(f"{spectrum[file]} Spectrum - Peak Positions") #
756
        " - {file}")
       else:
757
           ax[0].set_title(f"{spectrum[file]} Spectrum - Underground
758
       Considerations") # " - {file}")
759
       if file == 46:
760
           ax[0].set_xticks(fp_x_max_f[::2], [F * i for i in range(len(
761
       fp_x_max_f))][::2])
           ax[1].set_xticks(fp_x_max_f[::2], [F * i for i in range(len(
762
      fp_x_max_f))][::2])
           minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0,
763
      len(fp_x_max_f[::2]), len(fp_x_max_f[::2]) * 4 + 1), np.linspace(0,
      len(fp_x_max_f[::2]) - 1, len(fp_x_max_f[::2])), fp_x_max_f[::2]))
           ax[0].xaxis.set_minor_locator(minor_locator)
764
765
           ax[1].xaxis.set_minor_locator(minor_locator)
       else:
766
           ax[0].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f
767
      ))])
           ax[1].set_xticks(fp_x_max_f, [F * i for i in range(len(fp_x_max_f
768
      ))])
           minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0,
769
      len(fp_x_max_f), len(fp_x_max_f) * 4 + 1), np.linspace(0, len(
      fp_x_max_f) - 1, len(fp_x_max_f)), fp_x_max_f))
770
           ax[0].xaxis.set_minor_locator(minor_locator)
771
           ax[1].xaxis.set_minor_locator(minor_locator)
       ax[1].set_xlabel(r"Frequency-Differences $\delta\nu$ to Lowest FPI
772
      Peak [MHz]")
       # ax[1].set_xlabel("Channel ch [a.u.]")
773
       ax[0].set_ylabel(r"Amplitude $A$ [mV]")
774
       ax[1].set_ylabel(r"Amplitude $A$ [mV]")
775
776
       ax[0].set_ylim(-0.9, 3.8)
777
       ax[1].set_xlim(0, 1200)
778
```

```
mmp.legend(ax[0], loc=9, ncol=2, bbox_to_anchor=(0.5, -2.3))
780
       mmp.legend(ax[1], loc=9, ncol=2, bbox_to_anchor=(0.5, -0.28))
781
782
       if show_all is False:
783
784
           if save_images:
                mmp.save_fig(fig, path="../report/figures", name=f"{spectrum[
785
       file]} Spectrum Peak Positions", extension="pdf")
786
       else:
            if save_images:
787
                mmp.save_fig(fig, path="../report/figures", name=f"{spectrum[
788
       file]} Spectrum Underground Considerations", extension="pdf")
789
790
791 # %%
792
793 mmu.save_json(fit_results, "spectroscopy_results.json")
794
795
796 # %%
797
798 file = 48 # saturation
799
800 data = pd.read_csv(f"../data/oszi/NewFile{file}.csv", header=1, names=["x
       ", "CH1", "CH2", "CH3", "CH4", ""])
801
802 # Apply running average to correct for double line
803 ch1, x = mmc.running_average(data["CH1"], n=11, x=data["x"])
804
805 fig, ax = mmp.make_fig(ncols=1, nrows=2, grid=True, figsize=(9, 3),
       sharex=True, gridspec_kw={'height_ratios': [0.5, 1], 'hspace': 0})
806
807 xx = np.linspace(0, 1200, 400)
808
809 \text{ fp}_x = \text{data}["x"]
810 fp_y = data["CH2"] * 1e3
811 \text{ pd}_x = x
812 \text{ pd}_y = \text{ch1} * 1e3
813
814 \text{ fp}_x_max_num = []
815 \text{ fp}_x_max_used = \{33: (7, 15),
                      45: (11, 22),
816
                      46: (1, 12),
817
                      47: (1, 8),
818
                      48: (2, 14),
819
                      50: (8, 16),
820
                      51: (0, 6),
821
                      52: (1, 12)
822
823 fp_x_max = list(fp_x[fp_y \ge 2])
824 fp_y_max = list(fp_y[fp_y \ge 2])
825
826 # Find peaks n calibration etalon spectrum
827 appended = False
828 for i in range(len(fp_x_max)):
```

```
if appended is False:
829
           fp_x_max_num.append(1)
830
831
           appended = True
832
       if i < len(fp_x_max) - 1:
833
834
           if fp_x_max[i] - 5 <= fp_x_max[i + 1] <= fp_x_max[i] + 5:</pre>
835
                fp_x_max_num[-1] += 1
836
837
           else:
                appended = False
838
839
840 assert np.sum(fp_x_max_num) == len(fp_x_max), f"{file}: wrong length"
841
842 fp_x_max_f = []
843 for i in range(len(fp_x_max_num)):
844
       if file in fp_x_max_used.keys():
845
           if fp_x_max_used[file][0] <= i <= fp_x_max_used[file][1]:</pre>
                fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i
846
       ])):int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
847
       else:
848
            fp_x_max_f.append(np.mean(fp_x_max[int(np.sum(fp_x_max_num[:i])):
       int(np.sum(fp_x_max_num[:i]) + fp_x_max_num[i])]))
849
850 # calibration etalon uncertainty
851 fp_x_max_f_err = [5] * len(fp_x_max_f)
852
853 # Plot lines in calibration spectrum
854 mmp.plot(ax[0], fp_x, fp_y, config="plot", label="Oscilloscope Spectrum")
      of the FPI")
855 for i in range(len(fp_x_max_f)):
       # ax[0].text(fp_x_max_f[i] + 2, 3.3, i)
856
       mmp.plot(ax[0], fp_x_max_f[i], None, x_err=fp_x_max_f_err[i], config=
857
       "vspan", color="tab:red")
858
859 mmp.add_to_legend(ax[0], "Estimated Peak Positions with Uncertainties",
       linestyle="-", color="tab:red")
860
861 # mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5)
862 # mmp.plot(ax[1], data["x"], data["CH1"] * 1e3, config="scatter", s=0.5)
863
864 # finding peak positions for absorption spectroscopy
865 # Correct for linear underground
866 \text{ slope}_1_x = pd_x[150:210]
867 slope_1_y = pd_y[150:210]
868 slope_2_x = pd_x[500:680]
869 slope_2_y = pd_y[500:680]
870 \text{ slope}_3_x = \text{pd}_x[800:900]
871 slope_3_y = pd_y[800:900]
872 # mmp.plot(ax[1], slope_1_x, slope_1_y, color="tab:red", s=0.5, config="
       scatter")
873 # mmp.plot(ax[1], slope_2_x, slope_2_y, color="tab:red", s=0.5, config="
      scatter")
874 # mmp.plot(ax[1], slope_3_x, slope_3_y, color="tab:red", s=0.5, config="
       scatter")
```

```
876 out_linear_underground = mmp.fit(mmf.poly_1,
877
                                     np.concatenate([slope_1_x, slope_2_x,
      slope_3_x]),
                                     np.concatenate([slope_1_y, slope_2_y,
878
      slope_3_y]),
879
                                     # ax=ax[1],
                                     color="tab:red")
880
881
882 pd_y = pd_y - mmf.poly_1(pd_x, *out_linear_underground[0])
883 mmp.plot(ax[1], pd_x, pd_y, config="scatter", s=0.5)
884
885 mmp.add_to_legend(ax[1], "Data after Underground Subtraction", linestyle=
      "dotted", color="tab:blue", lw=3.5)
886
887 mmp.plot(ax[0], 1400, None, x_err=1400 - 990, config="vspan", color="tab:
      orange")
888 mmp.plot(ax[1], 1400, None, x_err=1400 - 990, config="vspan", color="tab:
      orange")
889 ax[1].fill_between([0, 1], [0, 0], [0, 0], color="tab:orange", label="
      Channels out of Mode Hop Free Range", alpha=0.2)
890
891
892 ax[0].set_title(f"{spectrum[file]} Spectrum - Large Range")
893 ax[0].set_xticks(fp_x_max_f[::2], [F * i for i in range(len(fp_x_max_f))
      ][::2])
894 ax[1].set_xticks(fp_x_max_f[::2], [F * i for i in range(len(fp_x_max_f))
      ][::2])
895 minor_locator = mpl.ticker.FixedLocator(np.interp(np.linspace(0, len(
      fp_x_max_f[::2]), len(fp_x_max_f[::2]) * 4 + 1), np.linspace(0, len(
      fp_x_max_f[::2]) - 1, len(fp_x_max_f[::2])), fp_x_max_f[::2]))
896 ax[0].xaxis.set_minor_locator(minor_locator)
897 ax[1].xaxis.set_minor_locator(minor_locator)
898 ax[1].set_xlabel(r"Frequency-Differences $\delta\nu$ to Lowest FPI Peak [
      MHz]")
899 # ax[1].set_xlabel("Channel ch [a.u.]")
900 ax[0].set_ylabel(r"Amplitude $A$ [mV]")
901 ax[1].set_ylabel(r"Amplitude $A$ [mV]")
902
903 ax[0].set_ylim(-0.9, 3.8)
904 ax[1].set_xlim(0, 1200)
905
906 mmp.legend(ax[0], loc=9, ncol=2, bbox_to_anchor=(0.5, -2.3))
907 mmp.legend(ax[1], loc=9, ncol=2, bbox_to_anchor=(0.5, -0.28))
908
909 if save_images:
       mmp.save_fig(fig, path="../report/figures", name=f"{spectrum[file]}
910
      Spectrum Large Range", extension="pdf")
```

#### Analysis of the Spectroscopy Results

1 # -\*- coding: utf-8 -\*-

```
3 import numpy as np
4 # import pandas as pd
5 # import matplotlib.pyplot as plt
6 # import scipy
7 # import scipy.odr as s_odr
8 import mymodules.usefultools as mmu
9 # import mymodules.calculate as mmc
10 # import mymodules.measure as mmm
11 # import mymodules.optimize as mmo
12 # import mymodules.functions as mmf
13 import mymodules.plot as mmp
14
15 # verbose = True
16 # si_format = False
17 # plot = True
18 # draft = False
19 save_images = False
20 # write_data = False
21
22
23 # %%
24
25 spectrum = {"46": "Fluorescence", "47": "Saturation", "51": "Absorption"}
26
27 fit_results = mmu.read_json("spectroscopy_results.json")
28 data_peaks = mmu.read_json("rubidium_peaks.json") # MHz
29
30
31 def ch_to_etalon(channel, file):
      \# F = \{ "46" : 760 ,
32
      #
              "51": 950,
33
              "47": 760}
      #
34
      F = \{ "46" : 700, 
35
            "51": 700,
36
            "47": 700}
37
38
      etalon_ch = fit_results[file]["fp"]
      return np.interp(channel, etalon_ch, [i * F[file] for i in range(0,
39
      len(etalon_ch))])
40
41
42 # %%
43
44 # file = "47"
45
46 # x_S = [ch_to_etalon(fit_results[file]["fits"][0][0], file) for i in
      range(len(fit_results[file]["fits"]))]
47 # x_S_err = [ch_to_etalon(fit_results[file]["fits"][0][0] + fit_results[
      file]["fits"][0][1], file) - ch_to_etalon(fit_results[file]["fits
      "][0][0], file) for i in range(len(fit_results[file]["fits"]))]
48
49
50 # mmu.print_to_table(["A1", "mu1", "sigma1", "A2", "mu2", "sigma2"],
                        mmu.sc_round(fit_results[file]["fits"][0][0],
51 #
      fit_results[file]["fits"][0][1], SI=True),
```

```
mmu.sc_round(x_S, x_S_err, SI=True),
52 #
53 #
                        )
54
55
56 # %%
57
58 # Peak-Positions and identify transitions
59
60 fig, ax = mmp.make_fig(grid=True, figsize=(9, 5))
61
62 file = "46"
63 x_F = [ch_to_etalon(fit_results[file]["fits"][0][3 * i + 1], file) for i
     in range(4)]
64 y_F = [2 for i in range(len(fit_results[file]["fits"]))]
65 x_F_err = [ch_to_etalon(fit_results[file]["fits"][0][3 * i + 1] +
      fit_results[file]["fits"][0][3 * i + 2], file) - ch_to_etalon(
      fit_results[file]["fits"][0][3 * i + 1], file) for i in range(4)]
66 x_F, x_F_err, y_F = mmu.convert_to_array(x_F, x_F_err, y_F)
67 x_F = x_F - x_F[0]
68 x_F_err = np.sqrt(x_F_err**2 + x_F_err[0]**2)
69
70 mmp.plot(ax,
71
           x_F, y_F,
72
           x_err=x_F_err,
73
           config="scatter",
74
           label="Fluorescence Peaks",
            color="tab:blue")
75
76
77
78 file = "51"
79 x_A = [ch_to_etalon(fit_results[file]["fits"][i][0][1], file) for i in
      range(len(fit_results[file]["fits"]))]
80 y_A = [1 for i in range(len(fit_results[file]["fits"]))]
81 x_A_err = [ch_to_etalon(fit_results[file]["fits"][i][0][1] + fit_results[
      file]["fits"][i][0][2], file) - ch_to_etalon(fit_results[file]["fits"
      [i][0][1], file) for i in range(len(fit_results[file]["fits"]))]
82 x_A, x_A_err, y_A = mmu.convert_to_array(x_A, x_A_err, y_A)
83 x_A = x_A - x_A[0]
84 x_A_err = np.sqrt(x_A_err**2 + x_A_err[0]**2)
85
86 mmp.plot(ax,
87
           x_A, y_A,
           x_err=x_A_err,
88
           config="scatter",
89
           label="Absorption Peaks",
90
           color="tab:orange")
91
92 ax.scatter(ch_to_etalon(400, file) - [ch_to_etalon(fit_results[file]["
      fits"][i][0][1], file) for i in range(len(fit_results[file]["fits"]))
      ][0], 1,
              color="tab:grey",
93
94
               alpha=0.8,
              marker = "x")
95
96 ax.errorbar(ch_to_etalon(400, file) - [ch_to_etalon(fit_results[file]["
      fits"][i][0][1], file) for i in range(len(fit_results[file]["fits"]))
```

```
FP-II
```

```
][0], 1,
               xerr=np.sqrt((ch_to_etalon(400 + 60, file) - ch_to_etalon
97
       (400, file))**2 + [ch_to_etalon(fit_results[file]["fits"][i][0][1] +
      fit_results[file]["fits"][i][0][2], file) - ch_to_etalon(fit_results[
      file]["fits"][i][0][1], file) for i in range(len(fit_results[file]["
      fits"]))][0]**2),
98
               color="tab:grey",
99
               alpha=0.8,
               fmt="--", capsize=5, label="")[-1][0].set_linestyle("--")
100
103 file = "47"
104 x_S = [ch_to_etalon(fit_results[file]["fits"][i][0][4], file) for i in
      range(len(fit_results[file]["fits"]))]
105 y_S = [0 for i in range(len(fit_results[file]["fits"]))]
106 x_S_err = [ch_to_etalon(fit_results[file]["fits"][i][0][4] + fit_results[
      file]["fits"][i][0][5], file) - ch_to_etalon(fit_results[file]["fits"
      [i][0][4], file) for i in range(len(fit_results[file]["fits"]))]
107 x_S, x_S_err, y_S = mmu.convert_to_array(x_S, x_S_err, y_S)
108 x_S = x_S - x_S[0]
109 x_S_err = np.sqrt(x_S_err**2 + x_S_err[0]**2)
110
111 mmp.plot(ax,
112
            x_S, y_S,
113
            x_err=x_S_err,
114
            config="scatter",
            label="Saturation Peaks",
            color="tab:green")
116
117
118 for ab in data_peaks["87"]:
       ax.axvline(ab, linestyle="--", color="tab:orange")
119
120 mmp.add_to_legend(ax, "Expected 87-Rb Peak Positions", linestyle="--",
      color="tab:orange")
121 for ab in data_peaks["85"]:
       ax.axvline(ab, linestyle="--", color="tab:red")
122
123 mmp.add_to_legend(ax, "Expected 85-Rb Peak Positions", linestyle="--",
      color="tab:red")
124
125 ax.axvline(3070, color="darkgreen")
126 # mmp.plot(ax, 3070, 2, x_err=x_F_err[0], color="tab:blue", config="
      scatter")
127 # mmp.plot(ax, 3070, 1, x_err=x_A_err[0], color="tab:orange", config="
      scatter")
128 # mmp.plot(ax, 3070, 0, x_err=x_S_err[0], color="tab:green", config="
      scatter")
129 ax.text(3070, 2.16, r"D$_1$", horizontalalignment="center")
130 mmp.add_to_legend(ax, r"Expected D$_1$-Line", color="darkgreen",
      linestyle="-")
131
133 # ax.text(50, 1.7, r"$^{87}$Rb P$_{1/2}$, F=1 <--> S$_{1/2}$, F=2")
134 # ax.text(50, 1.7, r"^{87}Rb ^{1/2}, F=1}_{S_{1/2}}, F=2}")
135 ax.text(0, 2.16, r"$^{87}$Rb $^{F=1}_{F=2}$", horizontalalignment="center
      ")
```

```
136 ax.text(0 + 50, 1.7, r"$T_1$")
137 ax.text(814, 2.16, r"$^{87}$Rb $^{F=2}_{F=2}$", horizontalalignment="
      center")
138 ax.text(814 + 50, 1.7, r"$T_2$")
139 ax.text(1518, 2.3, r"$^{85}$Rb $^{F=2}_{F=3}$", horizontalalignment="
      center")
140 ax.text(1518 + 50, 1.7, r"$T_3$")
141 ax.text(1880, 2.16, r"$^{85}$Rb $^{F=3}_{F=3}$", horizontalalignment="
      center")
142 ax.text(1880 + 50, 1.7, r"T_4")
143 ax.text(4554, 2.3, r"$^{85}$Rb $^{F=2}_{F=2}$", horizontalalignment="
      center")
144 ax.text(4554 + 50, 1.7, r"T_5")
145 ax.text(4915, 2.16, r"$^{85}$Rb $^{F=3}_{F=2}$", horizontalalignment="
      center")
146 ax.text(4915 + 50, 1.7, r"$T_6$")
147
148 ax.set_title("Spectroscopy Peaks on a Frequency Axis", y=1.16)
149 ax.set_xlabel(r"Frequency-Differences \delta\nu to ^{7}Bb ^{F=1}_{F=1}_{F}
      =2}$ [MHz]")
150
151 ax.set_yticks([0, 1, 2], [])
152 ax.set_xlim(-600, 5800)
154 mmp.legend(ax, loc=9, ncol=2, bbox_to_anchor=(0.5, -0.14))
156 if save_images:
       mmp.save_fig(fig, path="../report/figures", name="Spectroscopy
157
      Results", extension="pdf")
158
159
160 # %%
161
162 mmu.print_to_table(np.round(np.sort(np.concatenate((data_peaks["87"],
      data_peaks["85"]))), 1),
                      mmu.sc_round(x_S, x_S_err, SI=True),
163
                      mmu.sc_round(x_A, x_A_err, SI=True),
164
                      mmu.sc_round(x_F, x_F_err, SI=True),
165
                       environment=True, header=True, SI=True, copy=True)
166
167
168
169 # %%
170
171
172 sigma_sat = np.array([fit_results[file]["fits"][i][0][5] for i in range(
      len(fit_results[file]["fits"]))])
173 sigma_sat_err = np.array([fit_results[file]["fits"][i][1][5] for i in
      range(len(fit_results[file]["fits"]))])
174 sigma_abs = np.array([fit_results[file]["fits"][i][0][2] for i in range(
      len(fit_results[file]["fits"]))])
175 sigma_abs_err = np.array([fit_results[file]["fits"][i][1][2] for i in
      range(len(fit_results[file]["fits"]))])
176 print(1 - sigma_sat / sigma_abs)
```
```
177 print(np.sqrt((sigma_sat_err / sigma_abs)**2 + (sigma_sat / sigma_abs**2
      * sigma_abs_err)**2))
178
179
180 # %%
181
182 # Determine Rb composition of the cell
183
184 print("Saturation")
185 file = "47"
186 A_S = [fit_results[file]["fits"][i][0][0] for i in range(len(fit_results[
      file]["fits"]))]
187 A_S_err = [fit_results[file]["fits"][i][1][0] for i in range(len(
      fit_results[file]["fits"]))]
188
189 # ratio: % of 87 in 87 + 85
190 ratio_1 = A_S[0] / (A_S[2] + A_S[0])
191 ratio_1_err = np.sqrt((A_S_err[0] / (A_S[2] + A_S[0]) + A_S[0] / (A_S[2]
      + A_S[0])**2 * A_S_err[0])**2
192
                          + (A_S[0] / (A_S[2] + A_S[0])**2 * A_S_err[2])**2)
193 print(fr"87 Rb F=1 -> F=2 to 87 Rb F=1 -> F=2 + 85 Rb F=2 -> F=3 ratio: {
      ratio_1} +- {ratio_1_err}")
194
195 \text{ ratio}_2 = A_S[1] / (A_S[1] + A_S[3])
196 \text{ ratio}_2 \text{err} = \text{np.sqrt}((A_S \text{err}[1] / (A_S[3] + A_S[1]) + A_S[1] / (A_S[3]))
      + A_S[1])**2 * A_S_err[1])**2
                          + (A_S[1] / (A_S[3] + A_S[1])**2 * A_S_err[3])**2)
197
198 print(fr"87 Rb F=2 -> F=2 to 87 Rb F=2 -> F=2 + 85 Rb F=3 -> F=3 ratio: {
      ratio_2} +- {ratio_2_err}")
199
200
201 print("Absorption")
202 file = "51"
203 A_A = [fit_results[file]["fits"][i][0][0] for i in range(len(fit_results[
      file]["fits"]))]
204 A_A_err = [fit_results[file]["fits"][i][1][0] for i in range(len(
      fit_results[file]["fits"]))]
205
206 # ratio: % of 87 in 87 + 85
207 ratio_3 = A_A[0] / (A_A[0] + A_A[1] + A_A[2])
208 ratio_3_err = np.sqrt((A_A_err[0] / (A_A[0] + A_A[1] + A_A[2]) + A_A[0] /
        (A_A[0] + A_A[1] + A_A[2]) **2 * A_A_err[0]) **2
                          + (A_A[0] / (A_A[0] + A_A[1] + A_A[2])**2 * A_A_err
209
       [1])**2
                          + (A_A[0] / (A_A[0] + A_A[1] + A_A[2])**2 * A_A_err
210
       [2]) * * 2)
211 print(fr"87 Rb F=1 -> F=2 + F=2 -> F=2 to 85 Rb F=2 -> F=3 + F=3 -> F=3
      ratio: {ratio_3} +- {ratio_3_err}")
212
213
214 print("Fluorescence")
215 file = "46"
216 A_F = [fit_results[file]["fits"][0][3 * i] for i in range(4)]
217 A_F_err = [fit_results[file]["fits"][1][3 * i] for i in range(4)]
```

218

```
219 # ratio: % of 87 in 87 + 85
220 ratio_3 = (A_F[0] + A_F[1]) / (A_F[0] + A_F[1] + A_F[2])
221 ratio_3_err = np.sqrt(((A_F_err[0]) / (A_F[0] + A_F[1] + A_F[2]) + (A_F
       [0] + A_F[1]) / (A_F[0] + A_F[1] + A_F[2])**2 * A_F_err[0])**2
222
                          + ((A_F_err[1]) / (A_F[0] + A_F[1] + A_F[2])**2 *
       A_F_err[1])**2
                          + ((A_F[0] + A_F[1]) / (A_F[0] + A_F[1] + A_F[2])
223
       **2 * A_F_err[2])**2)
224 print (fr"87 Rb F=1 -> F=2 + F=2 -> F=2 to 87 Rb F=1 -> F=2 + F=2 -> F=2 +
       85 Rb F=2 -> F=3 + F=3 -> F=3 ratio: {ratio_3} +- {ratio_3_err}")
225
226
227 # %%
228
229 # Calculate hyperfie structure constant for saturation
230 # frequency-differences of two adjacent states
231
232 # All in MHz
233
234 file = "47"
235 x_S = [ch_to_etalon(fit_results[file]["fits"][i][0][4], file) for i in
      range(len(fit_results[file]["fits"]))]
236 x_S_err = [ch_to_etalon(fit_results[file]["fits"][i][0][4] + fit_results[
       file]["fits"][i][0][5], file) - ch_to_etalon(fit_results[file]["fits"
       [i][0][4], file) for i in range(len(fit_results[file]["fits"]))]
237 x_S, x_S_err = mmu.convert_to_array(x_S, x_S_err)
238
239 Rb87_A = (x_S[1] - x_S[0]) / 2
240 Rb87_A_err = 1 / 2 * np.sqrt(x_S_err[1]**2 + x_S_err[0]**2)
241 print("Rb87_A:", Rb87_A, "+-", Rb87_A_err)
242 \text{ Rb87}_A_\text{lit} = 407.25
243 Rb87_A_lit_err = 0.63
244 print("Sigma deviation:", abs(Rb87_A_lit - Rb87_A) / np.sqrt(
       Rb87_A_lit_err**2 + Rb87_A_err**2))
245 print("Percentile deviation:", abs(Rb87_A_lit - Rb87_A) / Rb87_A_lit)
246 # Magnetic Dipole Constant, 87Rb 52 P1/2A52 P1/2h 407.25(63) MHz [10, 11,
        27]
247
248 \text{ Rb85} = (x_S[3] - x_S[2]) / 3
249 Rb85_A_err = 1 / 3 * np.sqrt(x_S_err[3]**2 + x_S_err[2]**2)
250 print("Rb85_A:", Rb85_A, "+-", Rb85_A_err)
251 \text{ Rb85} \text{A} \text{lit} = 120.527
252 \text{ Rb85}_A_\text{lit}_\text{err} = 0.56
253 print("Sigma deviation:", abs(Rb85_A_lit - Rb85_A) / np.sqrt(
       Rb85_A_lit_err**2 + Rb85_A_err**2))
254 print("Percentile deviation:", abs(Rb85_A_lit - Rb85_A) / Rb85_A_lit)
255 # Magnetic Dipole Constant, 52 P1/2A52 P1/2h 120.527(56) MHz
```