

Inhaltsverzeichnis

1. Aufgabenstellung	4
2. Theoretische Grundlagen	5
2.1. Die Hyperfeinstruktur	5
2.1.1. Spektrallinien	5
2.1.2. Drehimpuls der Elektronenhülle	5
2.1.3. Drehimpuls des Atomkerns	6
2.1.4. Hyperfeinwechselwirkung	7
2.1.5. Zeeman-Effekt der Hyperfeinstruktur	8
2.2. Optisches Pumpen	10
2.2.1. Optisches Pumpen im vereinfachten Modell	10
2.2.2. Zeeman-Pumpen bei HFS-Termen von ^{87}Rb	12
2.3. Relaxation	13
2.3.1. Relaxationsprozesse	13
2.3.2. Relaxationszeiten	14
2.4. Das im Versuch verwendete Rubidium	15
2.5. Optische Elemente	18
2.5.1. Das $\lambda/4$ -Plättchen	18
2.5.2. Das Etalon	18
2.5.3. Der Laser	19
3. Versuchsaufbau und Durchführung	21
3.1. Aufbau	21
3.2. Laser- und Photodiodenkennlinie	22
3.3. Spektroskopie der Hyperfeinstruktur	22
3.4. Doppelresonanz	22
3.5. Spinpräzession	23
3.6. Messung der Relaxationszeit nach Dehmelt	23
3.7. Messung der Relaxationszeit nach Franzen	24
4. Auswertung	25
4.1. Laser- und Photodiodenkennlinie	25
4.2. Spektroskopie der Hyperfeinstruktur	28
4.3. Doppelresonanz	34
4.4. Spinpräzession	38
4.5. Bestimmung der Relaxationszeit nach Dehmelt	41
4.6. Bestimmung der Relaxationszeit nach Franzen	43
5. Zusammenfassung	46
A. Weitere Grafiken	48
A.1. Spektroskopie der Hyperfeinstruktur	48
A.2. Ausgewählte Spinpräzessions-Aufnahmen	50
A.3. Fits zur Relaxationszeit nach Dehmelt	53

B. Quelltexte	57
B.1. Laserdiodenkennlinie (<code>laser_kennlinie.py</code>)	57
B.2. Photodiodenkennlinie (<code>photo_kennlinie.py</code>)	58
B.3. Neutralfilter (<code>neutral_filter.py</code>)	59
B.4. Frequenzzeichnung der Hyperfeinstruktur (<code>freq_eich.py</code>)	60
B.5. Hyperfeinstrukturspektrum (<code>hfs_spektrum.py</code>)	62
B.6. Berechnung der Intervallkonstanten (<code>intervall_konst.py</code>)	65
B.7. Doppelresonanz (<code>doppelresonanz.py</code>)	67
B.8. Spinpräzession (<code>spin_praez.py</code>)	69
B.9. Fits zur Relaxationszeit nach Dehmelt (<code>dehmelt_fits.py</code>)	71
B.10. Relaxationszeit nach Dehmelt (<code>dehmelt.py</code>)	73
B.11. Relaxation nach Franzen (<code>franzen_mehrfach.py</code>)	74
B.12. Routinen für Oszilloskop-Daten (<code>oszi_messung.py</code>)	75
B.13. Konstanten und angegebene Werte (<code>konst.py</code>)	76
B.14. Hilfsroutinen für Fits (<code>fit_tools.py</code>)	77
B.15. Routinen zur Erzeugung von L ^A T _E X-Tabellen (<code>texgen.py</code>)	79
B.16. Weitere Hilfsroutinen (<code>tools.py</code>)	82

1. Aufgabenstellung

- Aufnahme des Hyperfeinstrukturspektrums der beiden Rubidium-Isotope ⁸⁵Rb und ⁸⁷Rb sowie durchführen einer Frequenzzeichnung mit dem Etalon. Die gemessenen Linien sind zu identifizieren und die Intervallkonstanten zu berechnen.
- Durch Doppelresonanz, d.h. Einstahlung von Radiofrequenz und resonantem Licht, ist der Kernspin beider Rubidium-Isotope sowie die Horizontalkomponente des Erdmagnetfelds zu bestimmen.
- Bestimmung der Vertikalkomponente des Erdmagnetfelds durch beobachten der Spinpräzession.
- Messung der Relaxationszeit nach Dehmelt (Magnetfeldinversion) und nach Franzen (Relaxation im Dunkeln).

2. Theoretische Grundlagen

2.1. Die Hyperfeinstruktur

2.1.1. Spektrallinien

Klassisch gesehen lässt sich die Grundstruktur der Atome durch die Coulombwechselwirkung der Elektronen mit dem Kern erklären: Die Elektronen bewegen sich auf verschiedenen "erlaubten" Bahnen, also mit verschiedenen Coulomb-Potenzialen um den Kern; wenn Übergänge zwischen den Bahnen stattfinden werden die charakteristischen Spektrallinien sichtbar. Mit höherer spektraler Auflösung lässt sich erkennen, dass diese Linien in Wirklichkeit aufgespalten sind, also aus mehreren kleinen Linien (Feinstruktur) bestehen. Diese Aufspaltung wird durch die magnetische Wechselwirkung von Bahnmoment und Spin des Elektrons hervorgerufen. Mit noch höherer Auflösung wird noch eine weitere, sehr kleine Aufspaltung dieser Linien sichtbar: die Hyperfeinstruktur. Ursache dafür ist die Wechselwirkung zwischen magnetischem Moment des Kerns und der Elektronenhülle. Im Folgenden soll dieser Effekt noch genauer beschrieben werden.

2.1.2. Drehimpuls der Elektronenhülle

Der Gesamtdrehimpuls \mathbf{j} eines Elektrons setzt sich aus seinem Eigendrehimpuls (Spin) \mathbf{s} und seinem Bahndrehimpuls \mathbf{l} zusammen:

$$\mathbf{j} = \mathbf{l} + \mathbf{s} . \quad (1)$$

Liegt ein Mehrelektronensystem vor, so koppeln bei vielen Atomen die Elektronenspins zu einem Gesamtspin \mathbf{S} und die Bahndrehimpulse zu einem Gesamtbahndrehimpuls \mathbf{L} . Bei LS -Kopplung (siehe Abbildung 1) präzedieren \mathbf{S} und \mathbf{L} um den raumfesten Vektor \mathbf{J} des Gesamtdrehimpulses. Der Gesamtdrehimpuls

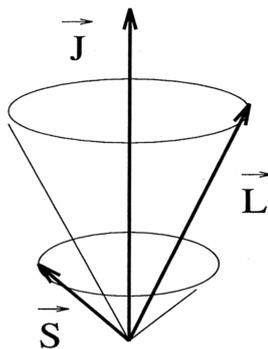


Abbildung 1: Vektordiagramm zur LS -Kopplung [1]

berechnet sich dann also durch

$$\mathbf{J} = \mathbf{L} + \mathbf{S} . \quad (2)$$

Da das Elektron Ladung trägt, ist mit diesem Drehimpuls ein magnetisches Moment verknüpft:

$$\boldsymbol{\mu}_J = -\frac{\mu_B}{\hbar} g_J \mathbf{J} . \quad (3)$$

Dies ist der Gesamtdrehimpuls der Elektronenhülle. Dabei ist $\mu_B = e\hbar/(2m_e)$ das Bohrsche Magneton und g_J der Landésche g-Faktor, der sich aus den Quantenzahlen J , S und L folgendermaßen berechnen lässt:

$$g_J = 1 + \frac{J(J+1) + S(S+1) - L(L+1)}{2J(J+1)} \quad (4)$$

2.1.3. Drehimpuls des Atomkerns

Für den Gesamtdrehimpuls des Atomkernes gilt Ähnliches; die Protonen und Neutronen aus denen der Kern besteht, besitzen jeweils einen Bahndrehimpuls innerhalb des Kerns (Schalenmodell des Atomkerns) und einen Spin 1/2. Die resultierende Größe ist der sogenannte Kernspin \mathbf{I} , für dessen Betrag gilt:

$$|\mathbf{I}| = \sqrt{I(I+1)} \hbar \quad (5)$$

Die so definierte Kernspinquantenzahl I ist ganz- oder halbzahlig und nimmt Werte zwischen 0 und $\frac{15}{2}$ an. Nach den Regeln der Quantenmechanik ist der Kernspin richtungsquantisiert, das heißt, er kann bezüglich einer fest gewählten Achse nur diskrete Werte annehmen. Diese sind festgelegt durch:

$$I_z = \hbar m_I \quad (6)$$

Hierbei ist m_I die magnetische Quantenzahl; auch sie kann ganz- oder halbzahlig sein und in ganzzahligen Schritten die Werte von $-I$ bis I annehmen. Damit ergeben sich insgesamt $2I + 1$ Einstellmöglichkeiten für I_z .

Auch mit dem Kernspin ist, aufgrund der Ladung des Kerns, ein magnetisches Moment verknüpft:

$$\boldsymbol{\mu}_I = \gamma \mathbf{I} = -\frac{\mu_K}{\hbar} g_I \mathbf{I} \quad (7)$$

Die Proportionalitätskonstante $\gamma = g_I \mu_K / \hbar$ bezeichnet man als das gyromagnetische Verhältnis. g_I ist dabei der Kern-g-Faktor, er ist eine dimensionslose Zahl, die typisch für den jeweiligen Kern ist und bisher noch nicht aus anderen Quantenzahlen berechnet werden kann. μ_K ist das sogenannte Kernmagneton; es berechnet sich analog zum Bohrschen Magneton durch $\mu_K = e\hbar/(2m_p)$ jedoch mit der Masse des Protons statt der Elektronenmasse. Für die z -Komponente des magnetischen Kernmoments gilt, ausgehend von Gleichung (7) und mit (6):

$$(\boldsymbol{\mu}_I)_z = \gamma I_z = \gamma \hbar m_I = g_I \mu_K m_I \quad (8)$$

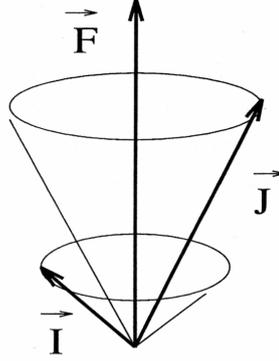


Abbildung 2: Vektordiagramm zur IJ -Kopplung [1]

2.1.4. Hyperfeinwechselwirkung

Wie anfangs schon erwähnt, entsteht die Hyperfeinstruktur durch die Wechselwirkung des magnetischen Moments des Kerns mit dem Feld der Elektronenhülle. Das magnetische Moment eines Kernes ist aber etwa um das 1000-fache kleiner als das Moment des Elektrons und die Wechselwirkungsenergie ist somit auch entsprechend gering. Daher wird die Kopplung der Elektronen innerhalb der Hülle nicht aufgehoben, sondern der Gesamtdrehimpuls \mathbf{J} der Elektronen und der Kernspin \mathbf{I} koppeln zu einem neuen Gesamtdrehimpuls \mathbf{F} des Atoms (siehe Abbildung 2):

$$\mathbf{F} = \mathbf{I} + \mathbf{J}. \quad (9)$$

Für die zugehörige Quantenzahl F gilt die Beziehung:

$$|I - J| \leq F \leq I + J \quad (10)$$

Somit kann F insgesamt $2I + 1$ oder $2J + 1$ Werte annehmen, je nachdem ob $I < J$ oder $J < I$. Die potentielle Energie des kernmagnetischen Moments $\boldsymbol{\mu}_I$ im Feld \mathbf{B}_J der Elektronenhülle ist nun gegeben durch:

$$E_{\text{HFS}} = -\boldsymbol{\mu}_I \cdot \mathbf{B}_J \quad (11)$$

damit ergibt sich für jede mögliche Einstellung des Spins ein potentielles Energieniveau, die Hyperfeinstruktur (siehe Abbildung 3). Der Betrag dieser magnetischen Zusatzenergien berechnet sich mit Gleichungen (5) und (7) durch:

$$\begin{aligned} E_{\text{HFS}} &= -|\boldsymbol{\mu}_I| |\mathbf{B}_J| \cos(\boldsymbol{\mu}_I, \mathbf{B}_J) \\ &= \mu_K g_I \sqrt{I(I+1)} B_J \cos(\mathbf{I}, \mathbf{J}) . \end{aligned} \quad (12)$$

Quadriert man nun Gleichung (9) und berücksichtigt, dass analog zu Gleichung (5) gilt: $|\mathbf{F}| = \sqrt{F(F+1)} \hbar$, so erhält man durch Auflösen nach $\cos(\mathbf{I}, \mathbf{J})$:

$$\cos(\mathbf{I}, \mathbf{J}) = \frac{I(I+1) + J(J+1) - F(F+1)}{2\sqrt{I(I+1) + J(J+1)}} . \quad (13)$$

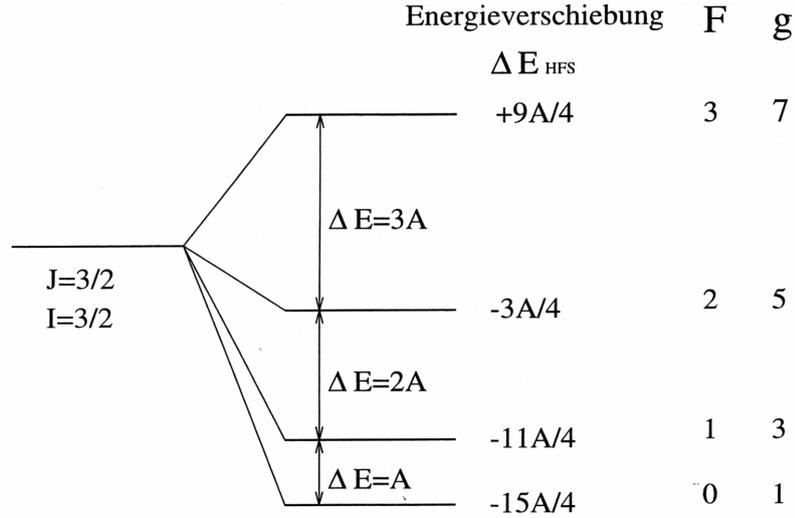


Abbildung 3: Hyperfeinstruktur und Intervallkonstante [1]

Durch Einsetzen von Gleichung (13) in Gleichung (12) ergibt sich:

$$E_{\text{HFS}} = \frac{A}{2} [F(F+1) - I(I+1) - J(J+1)] \quad (14)$$

mit der sogenannten *Intervallkonstante*

$$A = \frac{\mu_K g_I B_J}{\sqrt{J(J+1)}}. \quad (15)$$

Dadurch lässt sich mit Hilfe von A die Energiedifferenz zwischen zwei benachbarten Hyperfeinstruktur-Termen ($\Delta F = 1$) ausdrücken als:

$$\Delta E_{F \rightarrow (F+1)} = E_{\text{HFS}}(F+1) - E_{\text{HFS}}(F) = A(F+1). \quad (16)$$

Dieser Zusammenhang ist Abbildung 3 grafisch dargestellt.

2.1.5. Zeeman-Effekt der Hyperfeinstruktur

Befindet sich das Atom in einem äußeren Magnetfeld \mathbf{B} , so wird die Entartung der HFS-Terme aufgehoben und die einzelnen Niveaus spalten in $2F+1$ Unterniveaus mit $m_F = -F, -F-1, \dots, F-1, F$ auf, was als Zeeman-Effekt bekannt ist. Der Zeeman-Effekt tritt jedoch nur auf, wenn es sich bei dem äußeren Feld um ein relativ schwaches Feld handelt, das bedeutet es muss die Bedingung

$$g_J \mu_B B \ll A \quad (17)$$

erfüllt sein. A ist dabei die im obigen Abschnitt eingeführte Intervallkonstante.

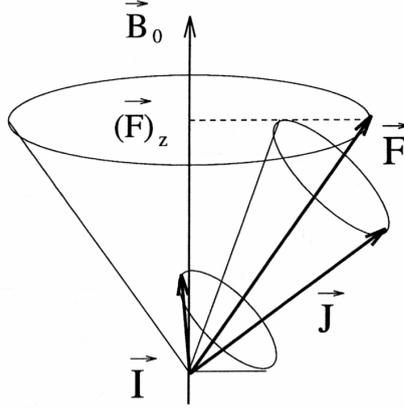


Abbildung 4: Vektordiagramm zum Zeeman-Effekt der Hyperfeinstruktur [1]

Liegt nun solch ein schwaches Magnetfeld vor, so wird die Kopplung von \mathbf{I} und \mathbf{J} innerhalb des Atoms nicht aufgehoben und der Kernspin \mathbf{I} und der Drehimpuls der Elektronenhülle \mathbf{J} präzedieren viel schneller um \mathbf{F} als der Gesamtdrehimpuls \mathbf{F} selbst um die Achse des äußeren Magnetfeldes präzediert (siehe Abbildung 4). \mathbf{F} ist auch wieder richtungsquantisiert, das heißt es muss gelten:

$$|\mathbf{F}_z| = m_F \hbar, \quad (18)$$

da m_F wie oben beschrieben die Werte von $-F$ bis F annimmt, ergeben sich für \mathbf{F} insgesamt $2F + 1$ Einstellmöglichkeiten relativ zum äußeren Feld. Durch das resultierende magnetische Moment $\boldsymbol{\mu}_F = \boldsymbol{\mu}_I + \boldsymbol{\mu}_J$ entsteht eine zusätzliche Wechselwirkungsenergie mit dem äußeren Magnetfeld:

$$E_B = -\boldsymbol{\mu}_F \cdot \mathbf{B}. \quad (19)$$

Da \mathbf{I} und \mathbf{J} schnell präzedieren, präzediert auch das magnetische Moment $\boldsymbol{\mu}_F$ sehr schnell und es kann deswegen zeitlich gemittelt werden. Damit wird Gleichung (19) zu:

$$E_B = g_F \mu_B m_F B \quad (20)$$

mit dem Landéschen g-Faktor des Gesamtatoms:

$$g_F = g_J \frac{F(F+1) + J(J+1) - I(I+1)}{2F(F+1)} - g_I \frac{\mu_k}{\mu_B} \frac{F(F+1) + I(I+1) - J(J+1)}{2F(F+1)} \quad (21)$$

Der zweite Term des g-Faktors kann jedoch gegenüber dem ersten vernachlässigt werden weil das Verhältnis $\frac{\mu_k}{\mu_B} \approx \frac{1}{2000}$ ist.

Für Zustände mit der Quantenzahl $J = 1/2$ kann, unter Vernachlässigung des zweiten Terms aus Gleichung (21), der g-Faktor des Gesamtatoms zu

$$g_F = \pm \frac{g_J}{2(I + 1/2)} \quad (22)$$

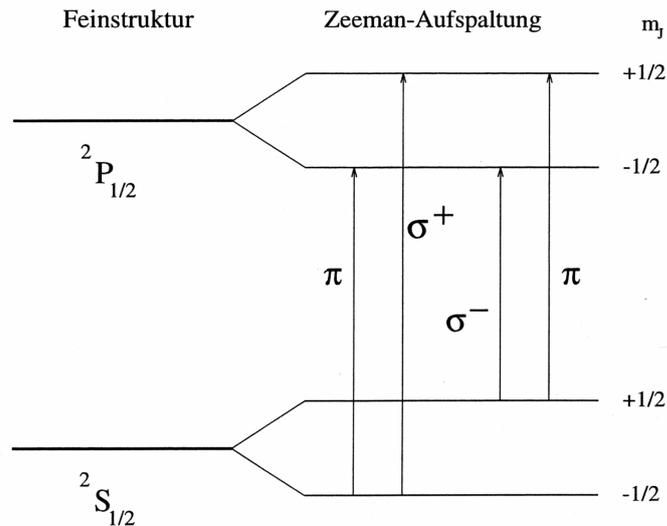


Abbildung 5: Vereinfachtes Termschema von Rubidium ohne HFS [1]

berechnet werden. Damit kann nun auch der Abstand zweier benachbarter Zeeman-Niveaus, also zwei Niveaus für die $\Delta m_F = 1$ gilt, berechnet werden. Setzt man Beziehung (22) in Gleichung (20) ein, so ergibt sich für den energetischen Abstand:

$$\Delta E_B = \Delta \nu h = \frac{gJ}{2(I + 1/2)} \mu_B B. \quad (23)$$

2.2. Optisches Pumpen

Optisches Pumpen bezeichnet eine Methode mit der Besetzungszahlen atomarer Zustände verändert werden können. Durch Einstrahlen von resonantem Licht (daher 'optisches' Pumpen) lässt sich eine Abweichung der Besetzungszahlen vom statistischen Gleichgewicht erreichen. Das Pumpen bedeutet in diesem Fall, dass Atome sich in bestimmten Energieniveaus ansammeln, auf Kosten derer andere Niveaus dagegen entleert werden. Das bedeutet letztendlich, dass die Spins der Atome in einem Ensemble in eine bestimmte Vorzugsrichtung ausgelenkt werden.

2.2.1. Optisches Pumpen im vereinfachten Modell

An einem einfachen Beispiel soll nun das Prinzip des optischen Pumpens erläutert werden. Dazu betrachten wir ein vereinfachtes Termschema von Rubidium, bei dem die Annahme gemacht wird, dass der Kern des Rubidiumatoms kein magnetisches Moment besitzt, also keine Hyperfeinstruktur auftritt. Liegt ein externes Magnetfeld an, spalten Grund- und angeregter Zustand wie in Abbildung 5 zu sehen durch den Zeeman-Effekt in jeweils zwei Unterniveaus auf. Zwischen

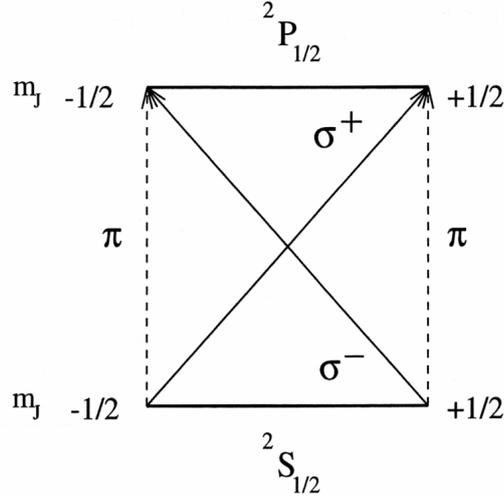


Abbildung 6: Mögliche Übergänge im vereinfachten Termschema von Rb [1]

diesen Niveaus sind insgesamt 4 optische Übergänge möglich. Bei den beiden Übergängen mit $\Delta m_J = 0$ wird linear (π) polarisiertes Licht emittiert bzw. absorbiert und für die beiden anderen Übergänge ergibt sich mit $\Delta m_J = +1$ rechtszirkular (σ^+) polarisiertes und mit $\Delta m_J = -1$ linkszirkular (σ^-) polarisiertes Licht. In Abbildung 6 sind die möglichen Übergänge nochmals in einem anderen Schema aufgetragen. Hier kann man gut erkennen was passiert wenn resonantes Licht eingestrahlt wird. Bei π -Licht wird sich aus Symmetriegründen keine Änderung der Besetzungszahlen ergeben; strahlt man jedoch σ^+ -Licht ein, kann nur einer der Übergänge, nämlich der Übergang von ${}^2S_{1/2}$, $m_J = -1/2$ nach ${}^2P_{1/2}$, $m_J = 1/2$ angeregt werden. Der angeregte Zustand kann nun entweder wieder zurück in den ${}^2S_{1/2}$, $m_J = -1/2$ Zustand oder in den anderen Grundzustand mit ${}^2S_{1/2}$, $m_J = +1/2$ zerfallen. Die Wahrscheinlichkeit für diese beiden Möglichkeiten ist 2:1, das heißt, zwei Drittel der Atome zerfallen zurück in ihren Ausgangszustand, von dem sie auch wieder erneut angeregt werden können, ein Drittel jedoch zerfällt in den anderen Grundzustand ${}^2S_{1/2}$, $m_J = +1/2$. Von dort aus kann durch σ^+ -Licht kein weiterer Übergang induziert werden. Strahlt man also kontinuierlich Licht ein, werden sich die Atome in diesem Zustand ansammeln, während der ${}^2S_{1/2}$, $m_J = -1/2$ Zustand entleert wird. Theoretisch könnte auf diese Weise der ${}^2S_{1/2}$, $m_J = -1/2$ Zustand vollkommen leer gepumpt werden. Es treten jedoch Prozesse auf, die dem Pumpvorgang entgegenwirken, also tendenziell auf die statistische Verteilung der Atome auf die verfügbaren Energiezustände hinwirken. Diese Prozesse sind als Relaxation bekannt und werden im folgenden Abschnitt 2.3 beschrieben. Durch das optische Pumpen erreicht man aber trotz dieser Prozesse eine signifikante Ausrichtung der Spins und spricht deshalb von einem optisch ausgerichteten Gas.

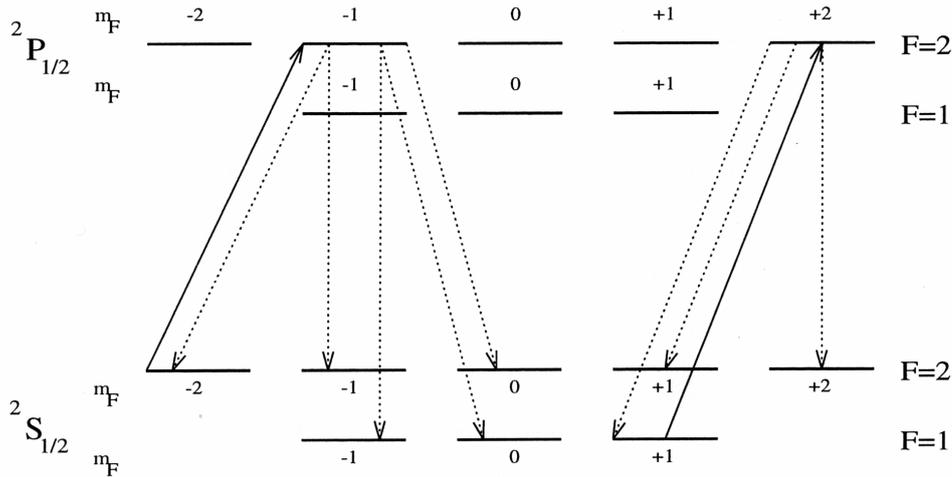


Abbildung 7: Optisches Pumpen bei Zeeman-Niveaus der HFS [1]

2.2.2. Zeeman-Pumpen bei HFS-Termen von ^{87}Rb

Nun soll noch das optische Pumpen bei vorhandener Hyperfeinstruktur beschrieben werden. Dazu betrachten wir das Rubidium-Isotop ^{87}Rb . In einem äußeren Magnetfeld spalten die Hyperfeinstrukturterme in $2F + 1$ Zeeman-Niveaus auf, das bedeutet, es müssen jetzt gegenüber dem obigen vereinfachten Beispiel viel mehr Übergänge berücksichtigt werden. In Abbildung 7 ist das HFS-Termschema von ^{87}Rb inklusive Zeeman-Aufspaltung zu sehen. Strahlt man σ^+ polarisiertes Licht der D_1 -Linie in ein solches Ensemble ein, so werden Atome aus den Zeeman-Niveaus des Grundzustands durch Absorption von Licht in Zeeman-Niveaus des ersten angeregten Zustands gehoben. Da es sich um σ^+ -Licht handelt muss sich bei diesen Übergängen die magnetische Quantenzahl m_F um eins erhöhen; es können also nur Atome angeregt werden, die sich nicht im Zustand $^2S_{1/2}$, $F = 2$, $m_F = +2$ befinden. Zerfällt ein angeregter Zustand, so gilt nach den Auswahlregeln für elektrische Dipolstrahlung für die magnetische Quantenzahl $\Delta m_F = 0, \pm 1$. Somit hat das angeregte Atom eine Wahrscheinlichkeit von $2/3$, in einen Zustand mit einer höheren magnetischen Quantenzahl als $m_F = -1$ zurück zu zerfallen, was dazu führt, dass die Atome tendenziell in den Zustand $^2S_{1/2}$, $F = 2$, $m_F = +2$ gepumpt werden, und sich dort mangels weiterer Anregung anreichern. Entsprechendes gilt auch für das Rubidium-Isotop ^{85}Rb , dort wird der Zustand $^2S_{1/2}$, $F = 3$, $m_F = +3$ angereichert, während sich die anderen Niveaus des Grundzustands zunehmend leeren.

2.3. Relaxation

2.3.1. Relaxationsprozesse

Relaxationsprozesse wirken dem optischen Pumpen entgegen, das bedeutet in einem Ensemble das durch optisches Pumpen ausgerichtet wurde, wo sich also Atome in einem bestimmten Zustand angesammelt haben, bewirken die Relaxationsprozesse, dass sich die Atome wieder statistisch auf alle verfügbaren Niveaus verteilen. Die folgenden Relaxationsprozesse sind in diesem Versuch relevant.

Stöße mit der Wand: Befindet sich in der Zelle mit den gepumpten Atomen (in unserem Fall Rubidium) ein Puffergas, welches keine Auswirkung auf die Ausrichtung der Rubidium Atome hat (z.B. Edelgase), so kommt als Relaxationsprozess nur noch der Stoß mit der Zellenwand in Frage. Man nimmt an, dass bei einem solchen Stoß die vorhandene Orientierung vollständig verloren geht. Ist der Druck des Puffergases genügend hoch, kann die Bewegung der Rubidiumatome als Diffusion beschrieben werden und der Relaxationsprozess der Wandstöße durch die Differentialgleichung

$$\frac{\partial n}{\partial t} = D \nabla^2 n \quad (24)$$

ausgedrückt werden. Dabei ist $D = D(p)$ der vom Druck des Puffergases abhängige Diffusionskoeffizient und n der Besetzungszahlenunterschied. Da die longitudinale Relaxationszeit mit der Zerfallsrate $1/T_D$ der Grunddiffusionsmode gleichgesetzt werden kann ergibt sich für eine kugelförmige Zelle mit Durchmesser d :

$$\frac{1}{T_D} = \frac{D_0 p_0}{p} \left(\frac{2\pi}{d} \right)^2 \quad (25)$$

Puffergasstöße: Selbst vollständig nichtmagnetische Stoffe wie Edelgase sind keine perfekten Puffer, sondern das Rubidiumatom kann auch bei einem Stoß mit einem solchen Puffergasatom seine Ausrichtung verlieren. Das passiert jedoch nur relativ selten da dabei die Relaxation, also die Neuorientierung des Spins, nur über schwache magnetische Dipol-Dipol-Wechselwirkung oder Spin-Bahn-Wechselwirkung zweiter Ordnung stattfinden kann. Berücksichtigt man nun, dass die Relaxation sowohl durch Diffusion der Atome zur Zellwand als auch durch Stöße mit dem Puffergas stattfindet, so kann die Relaxationszeit T_R für eine kugelförmige Resonanzzelle berechnet werden durch:

$$\frac{1}{T_R} = \frac{D_0 p_0}{p} \left(\frac{2\pi}{d} \right)^2 + N_0 \sigma(\langle v \rangle) \langle v_{rel} \rangle \left(\frac{p}{p_0} \right). \quad (26)$$

Dabei ist D_0 der Diffusionskoeffizient für Rubidium in Puffergas bei Normaldruck p_0 , p und d sind, wie gehabt, Druck des Puffergases und Zellendurchmesser, N_0 ist die Dichte der Puffergasatome bei Normaldruck, $\sigma(\langle v \rangle)$ ist der über die Geschwindigkeit gemittelte Depolarisationsquerschnitt für Puffergas und Rubidium,

und $\langle v_{rel} \rangle$ schließlich ist die mittlere relative Geschwindigkeit zwischen Gas- und Rb-Atomen.

Spinaustausch: Als drittes kommen für die Relaxation auch noch Stöße von Rubidiumatomen mit anderen Rubidiumatomen in Betracht. Stoßen zwei Rubidiumatome zusammen, so können die Valenzelektronen ihren Spin austauschen. Obwohl dabei für das gesamte Ensemble die Polarisation der Elektronenspins gleich bleibt, führt der Prozess zu einer Entkopplung von Kern- und Elektronenspin, was wiederum einen schnellen Besetzungsausgleich zwischen den beiden Hyperfeinniveaus $F = I \pm 1/2$ zur Folge hat. Dieser Vorgang wird daher auch ΔF -Relaxation genannt und führt zu einem Ausgleich der Besetzungszahlen beim Hyperfeinpumpen.

2.3.2. Relaxationszeiten

Geht man beim optischen Pumpen von einer Differenz $n = N_+ - N_-$ der Besetzungszahlen der pumpbaren Niveaus N_- und der Besetzungszahl N_+ des Pumpniveaus aus, so ergibt sich für den Pumpprozess die Differentialgleichung:

$$\left(\frac{dn}{dt} \right)_{\text{pump}} = \frac{N - n}{T_P}, \quad (27)$$

wobei N die Gesamtzahl der Atome, also $N = N_+ + N_-$, und T_P die *Pumpzeit* ist, welche über eine Proportionalitätskonstante α von der Intensität I des Pumplichts abhängt:

$$T_P = \frac{1}{\alpha I}. \quad (28)$$

Der durch das optische Pumpen erreichten Differenz der Besetzungszahlen n wirken Relaxationsprozesse (siehe Abschnitt 2.3.1) entgegen. Setzt man voraus, dass nicht weiter gepumpt wird, dann wird die Differenz im Verlauf der Zeit also immer kleiner und n strebt gegen Null. Vereinfachend lässt sich annehmen, dass für den gesamten Relaxationsvorgang folgende Differentialgleichung gilt:

$$\left(\frac{dn}{dt} \right)_{\text{relax}} = -\frac{n}{T_R}, \quad (29)$$

dabei ist T_R die den Relaxationsprozess charakterisierende *Relaxationszeit*. Der zeitliche Verlauf der reinen Relaxation, also die Lösung der Differentialgleichung (29) ist für $n(t=0) = n_0$:

$$n(t) = n_0 \exp\left(-\frac{t}{T_R}\right). \quad (30)$$

Da die Relaxationsprozesse auch während des Pumpprozesses ablaufen, muss man zur Beschreibung des Orientierungsprozesses, der während der Einstrah-

lung des Pumplichts stattfindet von einer Überlagerung der beiden Differentialgleichungen (27) und (29) ausgehen:

$$\left(\frac{dn}{dt}\right)_{\text{orient}} = \frac{N-n}{T_P} - \frac{n}{T_R}. \quad (31)$$

Bei diesem Vorgang stellt sich im Laufe der Zeit ein dynamisches Gleichgewicht zwischen Pump- und Relaxationsprozessen ein, bei dem die Differenz n der Besetzungszahlen maximal wird. Es gilt dabei:

$$\frac{dn}{dt} = 0 \quad \Rightarrow \quad n_{\text{max}} = \frac{N}{1 + \frac{T_P}{T_R}}. \quad (32)$$

Durch Einführung der *Orientierungszeit* τ , mit

$$\frac{1}{\tau} = \frac{1}{T_P} + \frac{1}{T_R} \quad (33)$$

lässt sich Gleichung (31) folgendermaßen vereinfachen:

$$\left(\frac{dn}{dt}\right)_{\text{orient}} = \frac{N}{T_P} - n \left(\frac{1}{T_P} + \frac{1}{T_R}\right) = \frac{N}{T_P} - \frac{n}{\tau}. \quad (34)$$

Mit $n(t=0) = n_0$ und Gleichung (32) lässt sich diese Differentialgleichung lösen durch:

$$n(t) = (n_0 - n_{\text{max}}) \exp\left(-\frac{t}{\tau}\right) + n_{\text{max}}. \quad (35)$$

Startet man mit dem optischen Pumpen aus dem statistischen Gleichgewicht, also mit $n_0 = 0$, dann erhält man für den zeitlichen Verlauf des Besetzungunterschieds die vereinfachte Lösung:

$$n(t) = n_{\text{max}} \left(1 - \exp\left(-\frac{t}{\tau}\right)\right). \quad (36)$$

2.4. Das im Versuch verwendete Rubidium

Die bei diesem Versuch verwendete Resonanzzelle enthält Rubidium im natürlichen Isotopengemisch von 72,8% ^{85}Rb und 27,2% ^{87}Rb . Die Quantenzahlen und g -Faktoren der beiden Isotope für den Grundzustand und den ersten angeregten Zustand sind in Tabelle 1 angegeben. Im Termschema in Abbildung 8 ist die Hyperfeinstruktur der beiden Rb-Isotope dargestellt. Die Zeeman-Aufspaltung der Hyperfeinstruktur zeigen Abbildungen 9 und 10. Weiter sind in Abbildung 11 für beide Isotope die Frequenzen der erlaubten optischen Übergänge zwischen den Hyperfeinstrukturen des ersten Anregungs- und des Grundzustands relativ zur D_1 -Linie dargestellt.

Isotop	I	Term	S	L	J	g_J	F	g_F
^{85}Rb	$\frac{5}{2}$	$^2S_{1/2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	2	2	$-\frac{1}{2}$
							3	$+\frac{3}{2}$
		$^2P_{1/2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{2}{3}$	2	$-\frac{1}{9}$
							3	$+\frac{1}{9}$
^{87}Rb	$\frac{3}{2}$	$^2S_{1/2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	2	1	$-\frac{1}{2}$
							2	$+\frac{1}{2}$
		$^2P_{1/2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{2}{3}$	1	$-\frac{1}{6}$
							2	$+\frac{1}{6}$

Tabelle 1: Quantenzahlen und g -Faktoren von Rubidium [1]

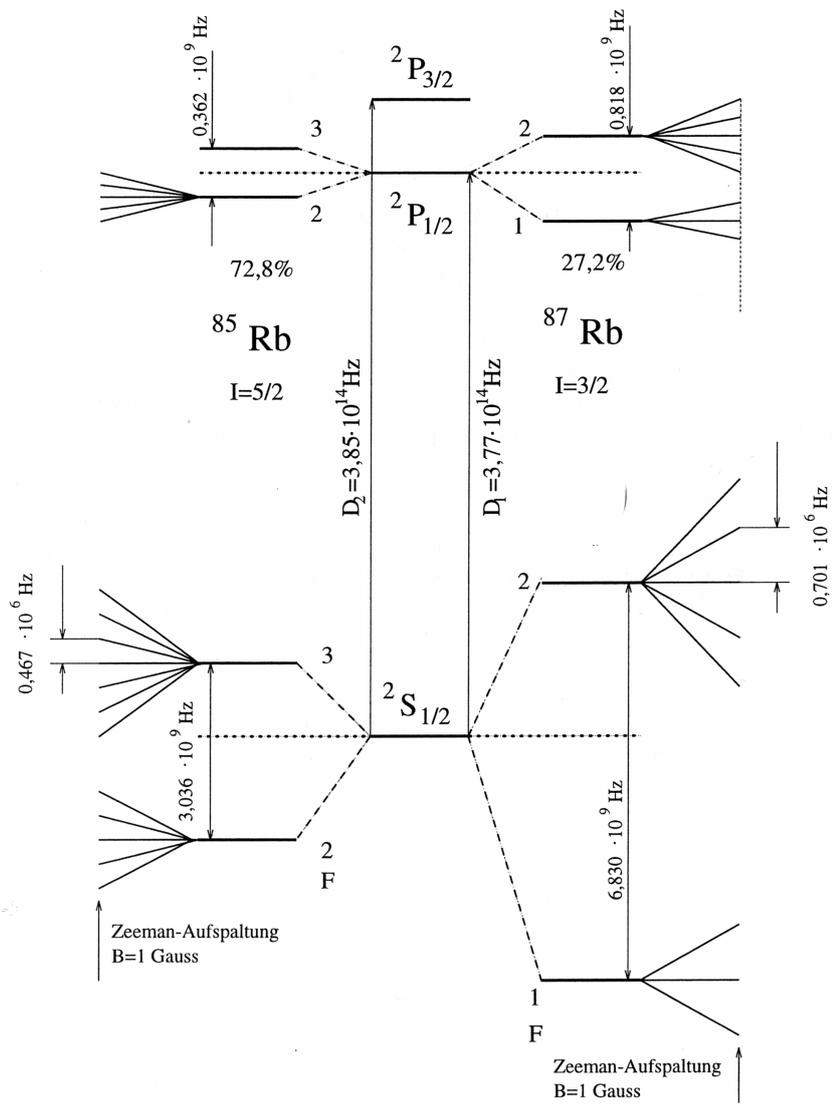


Abbildung 8: Termschema von ^{85}Rb und ^{87}Rb [1]

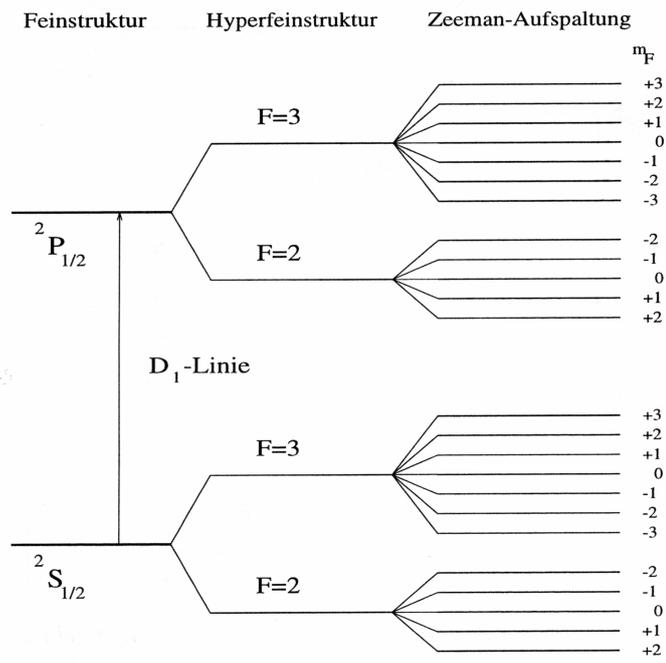


Abbildung 9: Zeeman-Aufspaltung der Hyperfeinstruktur bei ^{85}Rb [1]

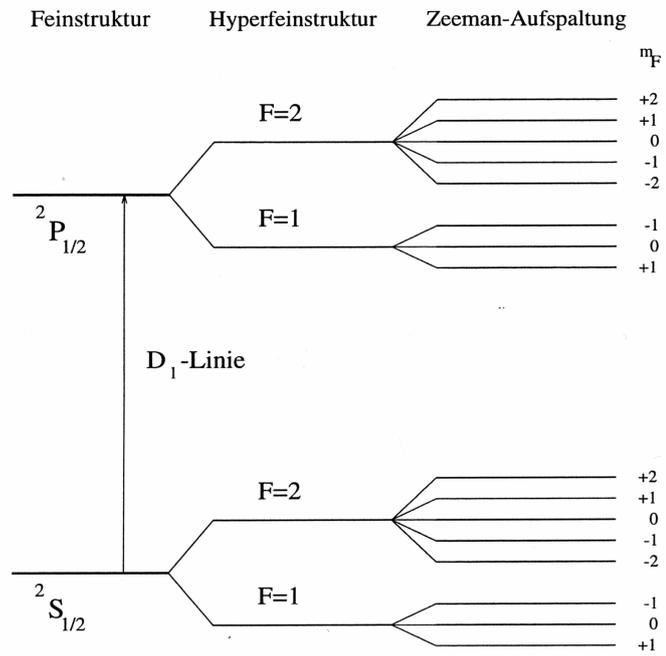


Abbildung 10: Zeeman-Aufspaltung der Hyperfeinstruktur bei ^{87}Rb [1]

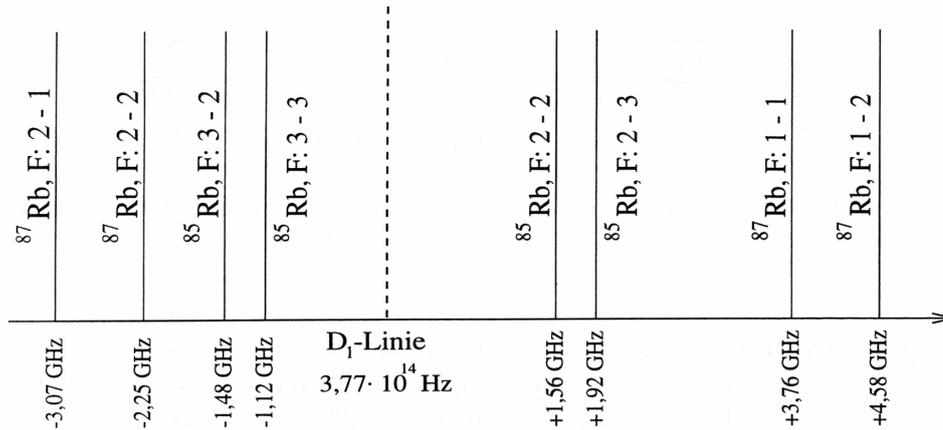


Abbildung 11: Übergangsfrequenzen zwischen der Hyperfeinstruktur des Übergangs $^2S_{1/2} \rightarrow ^2P_{1/2}$ beider Rb-Isotope relativ zur D_1 -Linie [1]

2.5. Optische Elemente

2.5.1. Das $\lambda/4$ -Plättchen

Mit einem $\lambda/4$ -Plättchen lässt sich aus linear polarisiertem Licht zirkular polarisiertes Licht erzeugen. Es besteht aus einem Material welches für eine eintreffende Lichtwelle in x- und y-Richtung zwei unterschiedliche Brechungsindizes besitzt. Der eine Anteil der Welle, z.B. der Anteil in x-Richtung, durchläuft das Material also mit einer geringeren Geschwindigkeit als der senkrecht dazu stehende andere Anteil. Durch diesen Laufzeitunterschied kommt es zwischen den beiden Komponenten des Lichts zu einer Phasenverschiebung. Für eine bestimmte Dicke des Plättchens (abhängig von der Wellenlänge) erreicht man eine Phasenverschiebung der beiden Komponenten von genau $\lambda/4$, wodurch aus linear polarisiertem Licht, dessen Polarisationsachse genau in einem Winkel von 45° zu den Achsen des Plättchens steht, zirkular polarisiertes Licht entsteht. Ist das Plättchen nicht genau einjustiert erhält man elliptisch polarisiertes Licht.

2.5.2. Das Etalon

Das Etalon besteht aus einer planparallelen Glasplatte welche an Vorder- und Rückseite teilverspiegelt ist. Ein Etalon arbeitet nach dem Prinzip der Vielstrahlinterferenz (siehe Abbildung 12); der Lichtstrahl tritt durch die Rückseite der ersten verspiegelten Fläche in die Platte ein und wird dann an der zweiten verspiegelten Fläche zum Teil hindurch gelassen und zum Teil wieder in die Platte zurückreflektiert. Dadurch kommt es innerhalb des Etalons zu einer Interferenz eines Strahlenbündels mit sich selbst. Bei einer festen Dicke d der Platte findet

dann, abhängig von der Wellenlänge und dem Einfallswinkel, konstruktive oder destruktive Interferenz statt.

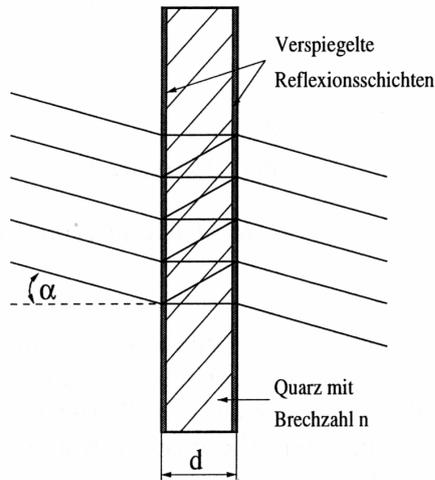


Abbildung 12: Schematische Darstellung des Etalons [1]

2.5.3. Der Laser

Das Pumplicht wird in diesem Versuch von einer Laserdiode geliefert. Es handelt sich dabei um eine Doppelhetero-Streifenlaserdiode aus GaAlAs die fast ihre gesamte Laserleistung in einem schmalen Frequenzband abstrahlt. Das austretende Licht ist linear polarisiert, jedoch stark divergent ($14^\circ - 30^\circ$), da an der abstrahlenden Fläche Beugung auftritt.

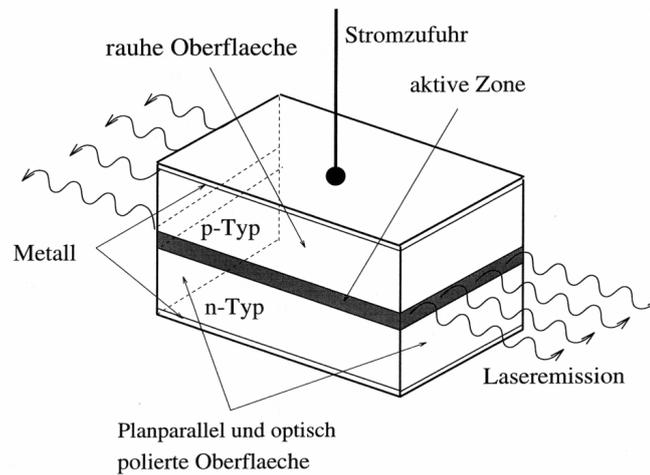


Abbildung 13: Schematischer Aufbau einer pn-Laserdiode [1]

Laserdioden bestehen aus einer oder meist mehreren Halbleiterschichten mit p-n-Übergängen (siehe Abbildung 13). Ein p-n-Übergang besteht aus einem Halbleiter des p-Typs (Löcherüberschuss, Akzeptor) und einem des n-Typs (Elektronenüberschuss, Donator) die miteinander in Kontakt gebracht werden. Das führt zu einer Diffusion von Elektronen in die p-Zone und von Löchern in die n-Zone. Dadurch ergibt sich aber wiederum im p-Gebiet eine negative und im n-Gebiet eine positive Raumladungsdichte, sodass ein elektrisches Feld entsteht welches der Diffusion entgegenwirkt. Dieses Feld bewirkt einen Potentialsprung, also eine Spannung in der Übergangszone, die zu der in Abbildung 14 dargestellten Verbiegung der Bandkanten von Valenz- und Leitungsband führt. Wird eine Spannung in Durchlassrichtung angelegt, so rekombinieren immer neue Elektronen und Löcher innerhalb der p-n-Grenzschicht unter Emission ihrer Rekombinationsenergie in Form von Licht. Die Fermienergie F_n der n-Region ist gegenüber der Fermienergie F_p der p-Region verschoben, in einer schmalen Zone existieren sowohl Löcher als auch Elektronen, damit ist ein Bereich der Besetzungsinversion realisiert und der Laserbetrieb wird möglich. Der zur Erzeugung von Laserlicht notwendige optische Resonator sind die Endflächen des Halbleiterkristalls selbst. Sind die Flächen planparallel und poliert, so tritt aufgrund des hohen Brechungsindex von $n = 3,6$ bei GaAs, am Halbleiter-Luft-Übergang eine Reflexion von ca. 32% auf.

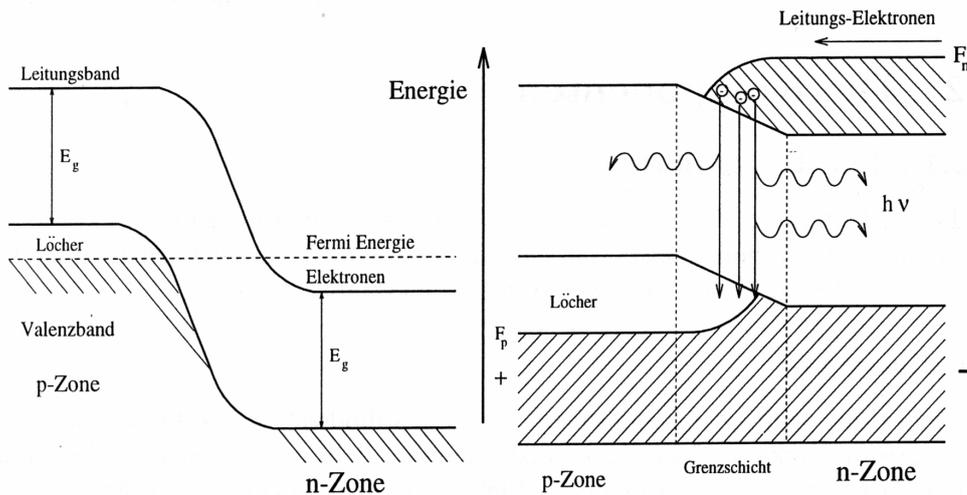


Abbildung 14: Links: pn-Übergang ohne äußere Spannung, Rechts: pn-Übergang mit in Durchlassrichtung angelegter Spannung [1]

3. Versuchsaufbau und Durchführung

3.1. Aufbau

Der Gesamtaufbau besteht hauptsächlich aus einer mit Rubidium gefüllten Glaszelle, um die herum verschiedene Helmholtzspulenpaare (siehe Abbildung 15) angebracht sind, sowie einer Peltier-gekühlten Laserdiode die als optische Pumpe für die in der Zelle befindlichen Rb-Atome dient. Das Licht der Laserdiode wird über eine Linse parallel in die Rb-Zelle eingestrahlt und danach mit einer weiteren Linse auf eine Photodiode fokussiert, welche zur Messung des transmittierten Lichtes dient. Das Signal der Photodiode wird an ein Oszilloskop weitergegeben, das wiederum mit einer auf dem Computer befindlichen Software ausgelesen werden kann. Außerdem stehen noch ein Linearpolarisator, ein $\lambda/4$ -Plättchen, ein Etalon und verschiedene Neutralfilter zur Abschwächung des Laserlichts zur Verfügung welche in den Strahlengang eingebracht werden können. Desweiteren kann mit einem in der Nähe der Zelle angebrachten RF-Sender Radiofrequenzstrahlung in die Zelle eingestrahlt werden. Die genaue Frequenz wird mit einem Frequenzmesser gemessen. Dieser Grundaufbau wurde während der Durchführung für die einzelnen Versuchsteile entsprechend modifiziert.

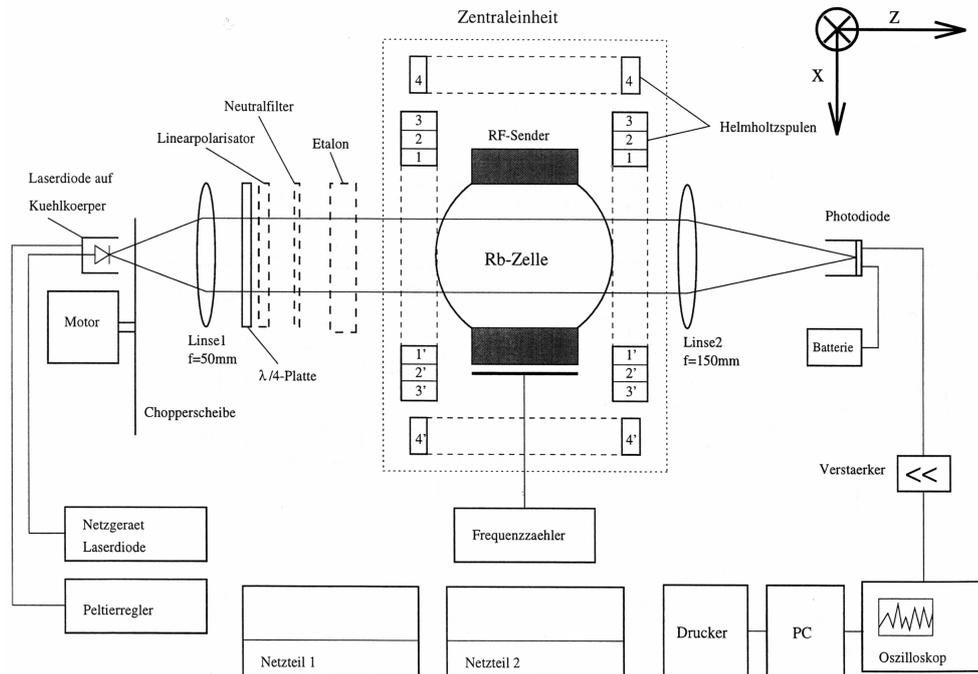


Abbildung 15: Schema des Versuchsaufbaus [1]

3.2. Laser- und Photodiodenkennlinie

Bevor wir mit den eigentlichen Messungen begannen, nahmen wir eine Laserkennlinie und eine Photodiodenkennlinie auf. Dazu waren die Rb-Zelle und alle anderen optischen Elemente, außer den Linsen, aus dem Strahlengang herausgenommen. Für die Laserkennlinie stellten wir eine feste Temperatur des Lasers von $34,6^\circ$ ein. Dann variierten wir den Laserdiodenstrom in kleinen Schritten von 0 mA bis etwa 77 mA und nahmen in Abhängigkeit dazu die Photodiodenspannung auf. Für die Photodiodenkennlinie stellten wir sowohl Temperatur als auch Laserdiodenstrom auf einen festen Wert ein. Dann brachten wir verschiedene Neutralfilter in den Strahlengang ein und notierten die Intensität des transmittierten Lichts anhand der Spannung der Photodiode. Diese Messung wiederholten wir am letzten Versuchstag noch einmal mit eingesetztem $\lambda/4$ -Plättchen und den Neutralfiltern die wir bei der Relaxationszeitmessung nach Dehmelt verwendet hatten.

3.3. Spektroskopie der Hyperfeinstruktur

Wir setzten die Rb-Zelle in den Strahlengang ein und stellten eine feste Temperatur von $34,0^\circ$ ein. Der Laserdiodenstrom wurde mit einer dreieckförmigen Spannung moduliert; dabei veränderten wir den Konstantanteil so, dass das Absorptionsspektrum der HFS auf dem Oszilloskopschirm sichtbar wurde. Als wir geeignete Einstellungen gefunden hatten lasen wir das Bild mit dem Computer aus. Danach nahmen wir die Rb-Zelle aus dem Strahlengang heraus und setzten bei unveränderten Einstellungen das Etalon ein. Hiermit nahmen wir dann ein Bild auf, das der Frequenzzeichnung diente. Auch diese Messung wiederholten wir am letzten Tag noch einmal, da wir, nachdem wir die Messung beendet hatten, bemerkten, dass eine Linse nicht richtig im Strahlengang befestigt war und wir deswegen schlechte Aufnahmen erhalten hatten.

3.4. Doppelresonanz

Der Laser wurde wie gehabt bei fester Temperatur von $34,0^\circ$ und festem Strom betrieben, die Werte für den Laserdiodenstrom mussten jedoch erst noch aufgesucht werden. Die Rb-Zelle befindet sich im Strahlengang. Zunächst brachten wir das $\lambda/4$ -Plättchen in den Strahlengang ein, um aus dem linear polarisierten Licht der Laserdiode zirkular polarisiertes Licht zu erzeugen. Damit auch wirklich rein zirkular polarisiertes Licht entstand, was das optische (Zeeman-)Pumpen ermöglichte, überprüften wir die Einstellung des Plättchens mit dem Linearpolarisator. Nun mussten die richtigen Einstellungen des Laserdiodenstroms gefunden werden. Dazu modulierten wir den Spulenstrom einer Spule (Spule 2) sägezahnförmig mit großer Amplitude und strahlten gleichzeitig eine feste Radiofrequenz in die Rb-Zelle ein. Durch Variation des Laserstroms konnten jetzt zwei feste Werte gefunden werden, bei denen Absorptionen auf dem Oszilloskop

zu erkennen waren, also optisches Zeeman-Pumpen bei einem der beiden Rb-Isotope auftrat. Bei einem dieser Werte wurde der Laser nachfolgend betrieben.

Nun modulierten wir den Strom von Spule 2 sinusförmig mit kleinerer Amplitude und ließen zusätzlich noch einen konstanten Strom durch Spule 1 fließen. Gleichzeitig wurde immernoch eine feste Radiofrequenz von etwa 500 kHz in die Zelle eingestrahlt. Den Strom von Spule 1 stellten wir so ein, dass die Absorptionssignale, die wir auf dem Oszilloskop betrachteten, äquidistant waren. Das bedeutete, dass die Resonanz zur Radiofrequenz und damit die Absorptionen genau beim Nulldurchgang des Feldes von Spule 2 auftraten, also beim alleinigen Magnetfeld von Spule 1. Dann sollte mit Hilfe einer weiteren Spule das vertikale Erdmagnetfeld kompensiert werden. Den richtigen Wert für den Spulen-Kompensationsstrom konnte man daran erkennen, dass die Absorptionssignale maximal wurden. Dann polten wir den Strom durch Spule 1 um und stellten ihn wieder auf einen Wert ein, bei dem die Absorptionen äquidistant waren. Danach wiederholten wir die gesamte Messung mit der zweiten Einstellung des Laserdiodenstroms, bei der das Licht die Resonanz des anderen Isotops traf. Mit den Werten des Spulenstromes konnten dann Kernspin und Horizontalkomponente des Erdmagnetfelds berechnet werden (siehe Auswertung, Abschnitt 4.3).

3.5. Spinpräzession

Bei dieser Messung kompensierten wir zunächst das horizontale Erdmagnetfeld durch Spule 1. Dann wurde eine kleine Spule von geringer Intensität schnell (mit einer Frequenz von etwa 100 Hz) ein- und ausgeschaltet, das vertikale Erdfeld wurde nicht kompensiert. Wird das Spulenfeld der kleinen Spule, in einer Zeit die kürzer als die Präzessionszeit der Rb-Atome im Erdmagnetfeld ist, ausgeschaltet, so kann die Polarisation des Ensembles nicht dem schnell wechselnden Magnetfeld folgen und bleibt stationär. In Abwesenheit des Spulenfeldes erfährt das Ensemble aufgrund des vertikalen Erdfeldes ein Drehmoment und es beginnt eine Präzessionsbewegung um die Richtung des Vertikalfeldes. Diese Spinpräzession kann mit dem Oszilloskop beobachtet werden, denn der Polarisationsvektor folgt dieser Präzession und der Rb-Dampf wechselt bei jedem Umlauf von einem Zustand maximaler Transparenz in einen Zustand maximaler Absorption.

Wir machten für beide Rb-Isotope jeweils Aufnahmen für unterschiedliche Abschwächungen des vertikalen Erdfeldes was durch einen Strom durch Spule 4 bewerkstelligt wurde.

3.6. Messung der Relaxationszeit nach Dehmelt

Bei diesem Versuchsteil führten wir die Messung der Relaxationszeit nach Dehmelts Methode der Magnetfeldinversion durch. Dazu kompensierten wir mit Spule 4 das vertikale Erdfeld durch einen Spulestrom von $I_{S4} = 70 \text{ mA}$ und erzeugten

durch ein Rechtecksignal (mit einer Frequenz von 55 Hz) mit einer kleinen Spule geringer Induktivität (Spule 5 mit 5 Wicklungen) ein magnetisches Wechselfeld.

Mit Hilfe der Laserdiode und des $\lambda/4$ -Plättchens strahlten wir resonantes, zirkular polarisiertes Licht in die Rb-Zelle ein, so dass es zu Zeeman-Pumpen des Ensembles kommt. Wir wählten dazu einen Laserstrom von $I_L = 66,0 \text{ mA}$, wodurch Resonanz für das Isotop ^{85}Rb erreicht wurde. Nach ausreichend langer Zeit - die Zeit zwischen dem Umpolen des Magnetfelds von Spule 5 reicht aus - stellt sich dabei ein Gleichgewichtszustand zwischen Pump- und Relaxationsprozessen ein, in dem die Rb-Atome weitgehend im vorherrschenden Magnetfeld ausgerichtet sind. Wird nun das Magnetfeld aufgrund der geringen Induktivität von Spule 5 schnell umgekehrt, so ist der Großteil der Atome nicht mehr im Magnetfeld ausgerichtet und es kann wieder verstärkt gepumpt werden. Dies äußert sich in einem sprunghaften Abfall der gemessenen Intensität. Durch das Pumpen kommt es wieder zu einer zunehmenden Polarisierung der Atome, wodurch die pumpbaren Zeeman-Niveaus des Grundzustand immer weiter entvölkert werden und die nach der Zelle gemessene Intensität wieder zunimmt, bis sich erneut ein Gleichgewichtszustand einstellt. Diesen Verlauf der gemessenen Intensität nahmen wir für verschiedene eingestrahlte Laserintensitäten auf, welche wir durch in den Strahlengang eingebrachte Neutralfilter erreichten. Um bessere Aufnahmen zu erhalten, erhitzen wir die Zelle vor den Messungen jeweils kurz mit einem Föhn, wobei wir versuchten die Temperatur bei der Aufnahme der verschiedenen Messungen konstant zu halten.

3.7. Messung der Relaxationszeit nach Franzen

Bei der Methode von Franzen zur Bestimmung der Relaxationszeit wird die Relaxation im Dunkeln beobachtet. Dazu wird das eingestrahlte Licht für eine kurze Zeit unterbrochen, wodurch das gepumpte Ensemble zum Teil relaxieren kann. Wird das Licht wieder eingeschaltet, so setzt der Pumpvorgang erneut ein, wobei der Verlauf der Intensität des transmittierten Lichts von der Länge der Dunkelzeit abhängt. Durch Analyse der Signalform für verschiedene Dunkelzeiten kann die Relaxationszeit bestimmt werden (siehe Abschnitt 4.6).

Zum periodischen Ein- und Ausschalten des Lichts stand uns eine Drehscheibe mit gleich weit entfernten, gleich großen Aussparungen, ein sogenannter *Chopper*, zur Verfügung, der in den Strahlengang eingebracht werden konnte. Das vertikale Erdfeld wurde auch in diesem Versuchsteil mit einem Spulenstrom von $I_{S4} = 70 \text{ mA}$ durch Spule 4 kompensiert und das optische Pumpen durch die Einstrahlung von σ^+ -Licht mit dem $\lambda/4$ -Plättchen ermöglicht. Um überhaupt einen brauchbaren Anstieg nach der Dunkelzeit erkennen zu können, war es nötig die Zelle durchgehend mit dem Föhn zu heizen. Dabei die Temperatur konstant zu halten gestaltete sich äußerst schwierig. Wir versuchten dies jedoch dadurch zu erreichen, dass wir die gemessene Amplitude des Signals bei allen Messungen etwa auf gleicher Höhe hielten.

4. Auswertung

4.1. Laser- und Photodiodenkennlinie

Die aufgenommene Laserkennlinie ist in Abbildung 16 dargestellt. Dazu trugen wir die gemessene Photodiodenspannung U gegen den eingestellten Laserdiodenstrom I_L auf. Als Fehler für den Laserdiodenstrom nahmen wir $s_{I_L} = 0,1 \text{ mA}$ an. Die Spannung U maßen wir mit Hilfe des Oszilloskops, wobei wir den Laserdiodenstrom so einstellten, dass die Photodiodenspannung genau auf einer Skaleneinteilung des Oszilloskopschirms lag. Dadurch ließ sich die Spannung sehr genau ablesen und wir konnten einen Fehler von $0,02 \text{ V}$ annehmen. Man kann in Abbildung 16 deutlich erkennen, dass ab einem Laserdiodenstrom von $I_L \approx 52 \text{ mA}$ der Laserbetrieb der Diode einsetzt, und dass ab dieser Stelle wie erwartet die Intensität (Photodiodenspannung) linear mit dem Laserdiodenstrom ansteigt.

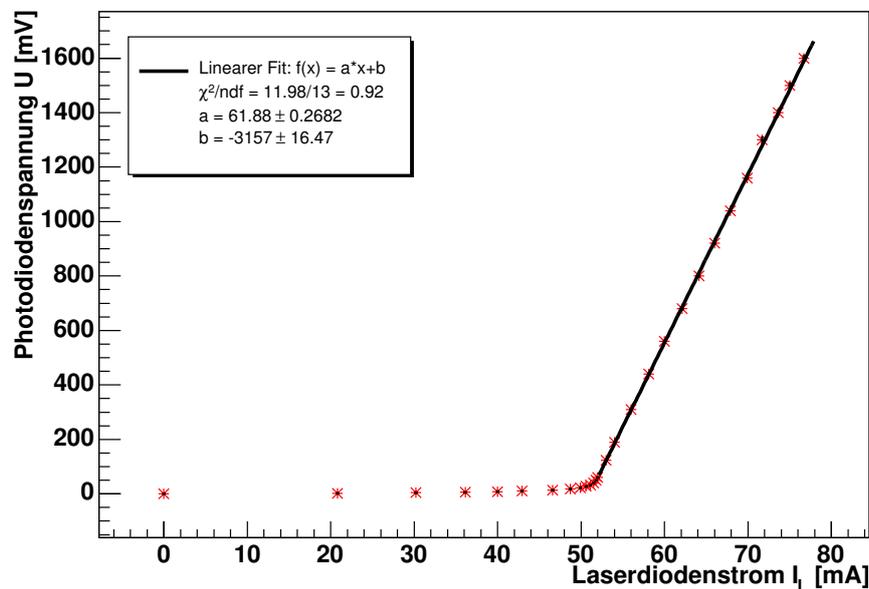


Abbildung 16: Die aufgenommene Laserkennlinie

Bei der eingestellten Temperatur von $T = 34,6^\circ\text{C}$ kam es im Bereich der aufgenommenen Werte zu keinem Modensprung des Lasers und es wurde auch der lineare Bereich nicht verlassen. Um die Laserdiode zu schonen vermieden wir es den Laserstrom weiter zu erhöhen und blieben daher unter einem Wert von etwa $I_L \approx 77 \text{ mA}$. Im weiteren Verlauf der Messung wurde von uns die Temperatur der Laserdiode auf $T = 34,0^\circ\text{C}$ gesenkt, wodurch man bei höheren Laserdiodenströmen einen Modensprung erkennen konnte. Wir führten bei dieser Temperatur keine weitere Aufnahme der Laserkennlinie durch, achteten aber bei der Wahl der

Bezeichnung	Filter x_{alt}	Filter x_{neu}	Photodiodenspannung [V]
leer	0	0	$1,384 \pm 0,008$
(-)	-	-	$1,144 \pm 0,008$
(1)	-	-	$0,696 \pm 0,008$
-0,37	-0,37	-0,378	$0,552 \pm 0,008$
-0,48	-0,48	-0,514	$0,512 \pm 0,008$
-1,03	-1,03	-1,170	$0,120 \pm 0,008$
1,6	-1,6	-1,142	$0,136 \pm 0,008$
(1+48)	-	-	$0,256 \pm 0,008$

Tabelle 2: Angegebene Abschwächungswerte und gemessene Photodiodenspannungen für die verwendeten Neutralfilter

Temperatur darauf, dass die Modensprünge ausreichend weit über dem verwendeten Arbeitsstrom von etwa $I_L \approx 67 \text{ mA}$ auftraten.

Obwohl die Linearität der Photodiode schon aus Abbildung 16 ersichtlich ist, sollte sie trotzdem noch durch die Aufnahme einer Photodiodenkennlinie verifiziert werden, bei der die Laserintensität durch unterschiedliche in den Strahlengang eingebrachte Neutralfilter abgeschwächt wurde. Für einen Teil der Neutralfilter waren die Abschwächungswerte x angegeben, wobei folgender Zusammenhang zwischen x und der relativen Intensität $I_{\text{rel}} = I/I_0$ besteht:

$$x = \log \left(\frac{I}{I_0} \right). \quad (37)$$

Das Problem, das wir dabei hatten, war aber, dass nur eine sehr begrenzte Zahl brauchbarer Neutralfilter vorhanden war und dass nicht zu allen Filtern die Abschwächungswerte angegeben waren. Weiter gab es zu jedem der beschrifteten Filtern insgesamt drei verschiedene Werte von x (zwei im Praktikumsordner und einen weiteren vor Ort), die erheblich voneinander abwichen und für die auch keine Unsicherheiten angegeben waren. Wir versuchten trotzdem einen linearen Zusammenhang zwischen den aus den angegebenen x berechneten relativen Intensitäten und der jeweils gemessenen Photodiodenspannung U herzustellen. Die dazu verwendeten angegebenen und die gemessenen Werte sind in Tabelle 2 aufgeführt. Wie man aber den Abbildungen 17 und 18 sehen kann, ist uns dies aber leider nicht gelungen. Weder die im Versuchsordner angegebenen Werte (als x_{alt} bezeichnet), noch die am Versuchsaufbau angebrachten Werte x_{neu} stimmen mit unseren Messungen überein. Ein Defekt der Photodiode kann als Grund dafür ausgeschlossen werden, da wir sonst auch bei der Laserkennlinie keinen linearen Zusammenhang hätten erreichen können.

Da wir in Abschnitt 4.5 bei der Bestimmung der Relaxationszeit nach Dehmelt verlässliche Abschwächungswerte der Neutralfilter brauchen und wir bei dieser Messung auch Filter ohne Wertangaben verwendet haben, führten wir eine eigene Bestimmung der Abschwächungswerte durch. Die Ergebnisse sind in Tabelle 3 angegeben. Dabei berechnet sich der Fehler der relativen Intensität $I_{\text{rel}} = I/I_0$

Bezeichnung	Intensität $I[V]$	relative Intensität I_{rel}	Filter x
leer	$1,384 \pm 0,008$	$1,0000 \pm 0,0000$	$0,000 \pm 0,000$
(-)	$1,144 \pm 0,008$	$0,8266 \pm 0,0075$	$-0,190 \pm 0,009$
(1)	$0,696 \pm 0,008$	$0,5029 \pm 0,0065$	$-0,687 \pm 0,013$
-0,37	$0,552 \pm 0,008$	$0,3988 \pm 0,0062$	$-0,919 \pm 0,016$
-0,48	$0,512 \pm 0,008$	$0,3699 \pm 0,0062$	$-0,994 \pm 0,017$
-1,03	$0,120 \pm 0,008$	$0,0867 \pm 0,0058$	$-2,445 \pm 0,067$
1,6	$0,136 \pm 0,008$	$0,0983 \pm 0,0058$	$-2,320 \pm 0,059$
(1+48)	$0,256 \pm 0,008$	$0,1850 \pm 0,0059$	$-1,688 \pm 0,032$

Tabelle 3: Gemessene Intensitäten (Photodiode), relative Intensitäten und bestimmte Abschwächungswerte der verwendeten Neutralfilter

aus dem Fehler der unabgeschwächten Intensität I_0 des Lasers und den abgeschwächten Intensitäten I (bzw. den jeweiligen Photodiodenspannungen):

$$s_{I_{\text{rel}}} = \sqrt{\left(\frac{\partial I_{\text{rel}}}{\partial I}\right)^2 s_I^2 + \left(\frac{\partial I_{\text{rel}}}{\partial I_0}\right)^2 s_{I_0}^2} = |I_{\text{rel}}| \sqrt{\left(\frac{s_I}{I}\right)^2 + \left(\frac{s_{I_0}}{I_0}\right)^2},$$

und pflanzt sich für die Abschwächung x wie folgt fort:

$$s_x = \sqrt{\left(\frac{\partial x}{\partial I_{\text{rel}}}\right)^2 s_{I_{\text{rel}}}^2} = \left|\frac{s_{I_{\text{rel}}}}{I_{\text{rel}}}\right|.$$

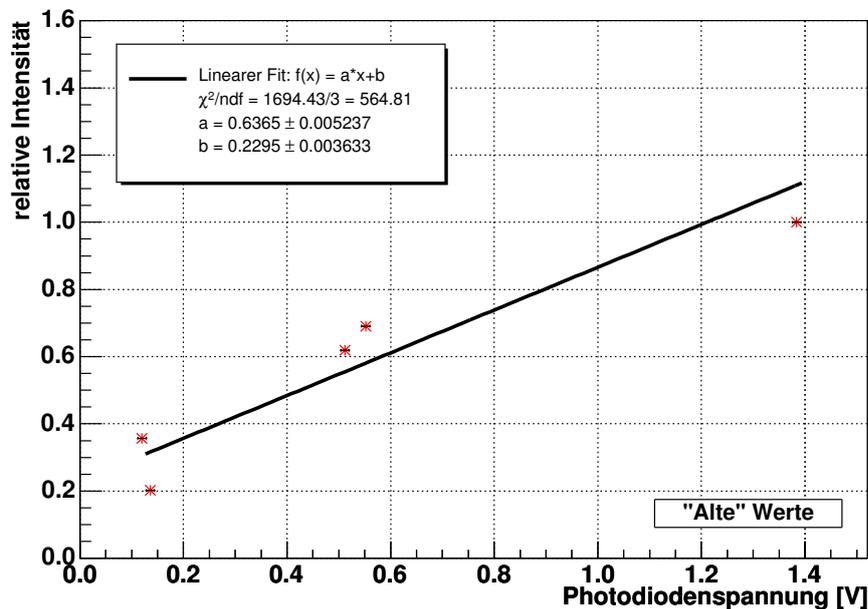


Abbildung 17: Die aus den im Praktikumsordner angegebenen Abschwächungswerten berechneten relativen Intensitäten gegen die Photodiodenspannung aufgetragen ergaben keinen linearen Zusammenhang

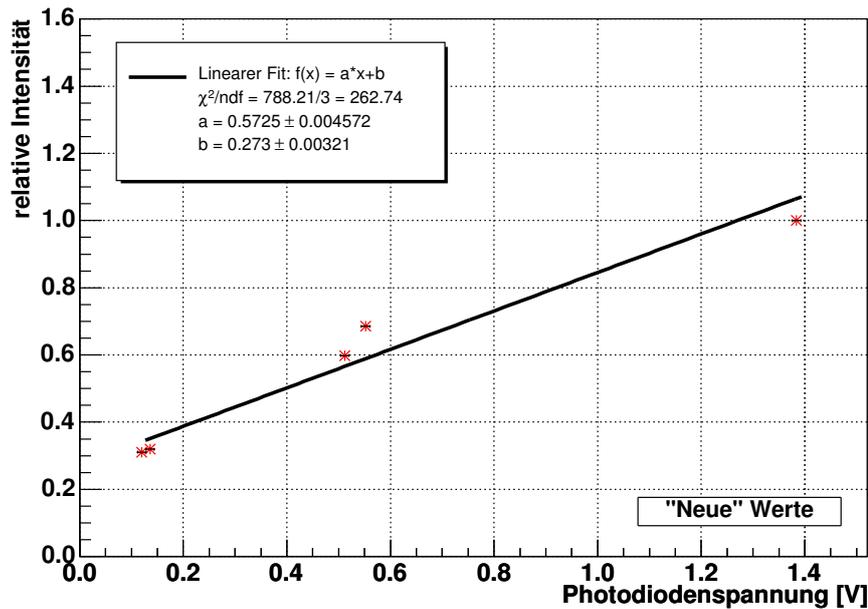


Abbildung 18: Die aus den vor Ort angegebenen Abschwächungswerten berechneten relativen Intensitäten gegen die Photodiode Spannung aufgetragen ergaben ebenfalls keinen linearen Zusammenhang

4.2. Spektroskopie der Hyperfeinstruktur

Frequenzzeichnung

Nach der Aufnahme des Hyperfeinspektrums führten wir eine Eichung mit Hilfe eines Etalons mit einem freien Spektralbereich von $\text{FSR} = 9924 \pm 30 \text{ MHz}$ durch. Da wir zwischen der Aufnahme der Hyperfeinstruktur und der Eichung keine Änderung an den Einstellungen vornahmen, konnten wir über die Zeit die Frequenzdifferenzen der Absorptionspeaks im Hyperfeinspektrum bestimmen. Um die Abstände zwischen den Maxima der Eichung besser bestimmen zu können, subtrahierten wir am Oszilloskop die Dreiecksspannung vom Signal der Photodiode (siehe dazu Abbildungen 29 und 30 in Anhang A.1). Die Abstände zwischen den verschiedenen Maxima sind in Abbildung 19 dargestellt. Wie man sieht unterscheiden sich die zeitlichen Abstände zwischen den Peaks bei steigender Dreiecksspannung (linke Seite) von denen bei fallender Dreiecksspannung (rechte Seite). Außerdem nehmen die Abstände zwischen den Peaks auf der linken Seite nach außen hin zu und auf der rechten Seite nach außen hin ab. Trotz längeren Überlegens fanden wir keine Erklärung für dieses Phänomen, insbesondere da der Anstieg des Dreiecksignals recht linear verläuft und sowohl der Spannungsanstieg, als auch die Anstiegszeit auf beiden Seiten ziemlich genau übereinstimmen.

Da der zur Aufnahme der HFS verwendete Abschnitt im Bereich der inneren

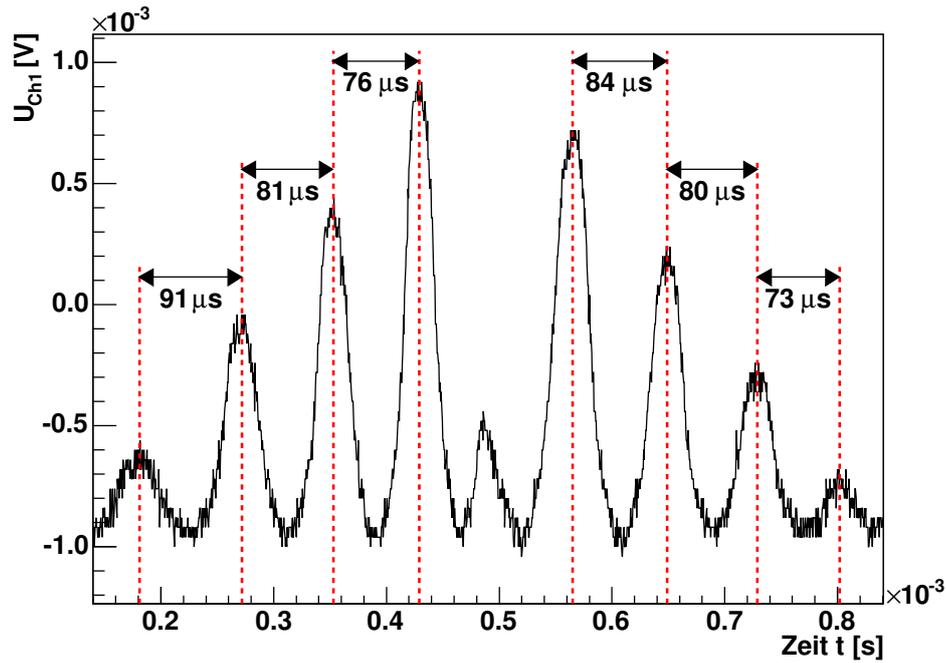


Abbildung 19: Frequenzzeichnung mit Hilfe des FSR des Etalons

beiden Interferenzmaxima der Etalonaufnahme liegt (vgl. Abbildungen 29 und 31 in Anhang A.1), sahen wir davon ab alle Maxima zur Frequenzzeichnung heranzuziehen, sondern verwendeten für beide Seiten jeweils nur die beiden inneren Peaks. Als Ablesefehler für die beiden inneren Peaks nahmen wir $s_t = 1 \mu\text{s}$ an. Der Fehler für die Differenz $\Delta t = |t_1 - t_2|$ berechnet sich nach gaußscher Fehlerfortpflanzung durch

$$s_{\Delta t} = \sqrt{s_{t_1}^2 + s_{t_2}^2} = \sqrt{2} \mu\text{s} .$$

Der Eichfaktor f berechnet sich aus dem Quotienten von FSR und Zeitdifferenz, also durch $f = \text{FSR}/\Delta t$. Der Fehler lässt sich damit fortpflanzen durch:

$$s_f = |f| \sqrt{\left(\frac{s_{\text{FSR}}}{\text{FSR}}\right)^2 + \left(\frac{s_{\Delta t}}{\Delta t}\right)^2}$$

Für die linke Seite erhielten wir als Zeitdifferenz und als Eichfaktor:

$$\Delta t_l = (76,0 \pm 1,4) \mu\text{s} \quad \text{und} \quad f_l = (130,58 \pm 2,46) \text{ MHz}/\mu\text{s} ,$$

und für die rechte Seite:

$$\Delta t_r = (84,0 \pm 1,4) \mu\text{s} \quad \text{und} \quad f_r = (118,14 \pm 2,02) \text{ MHz}/\mu\text{s} .$$

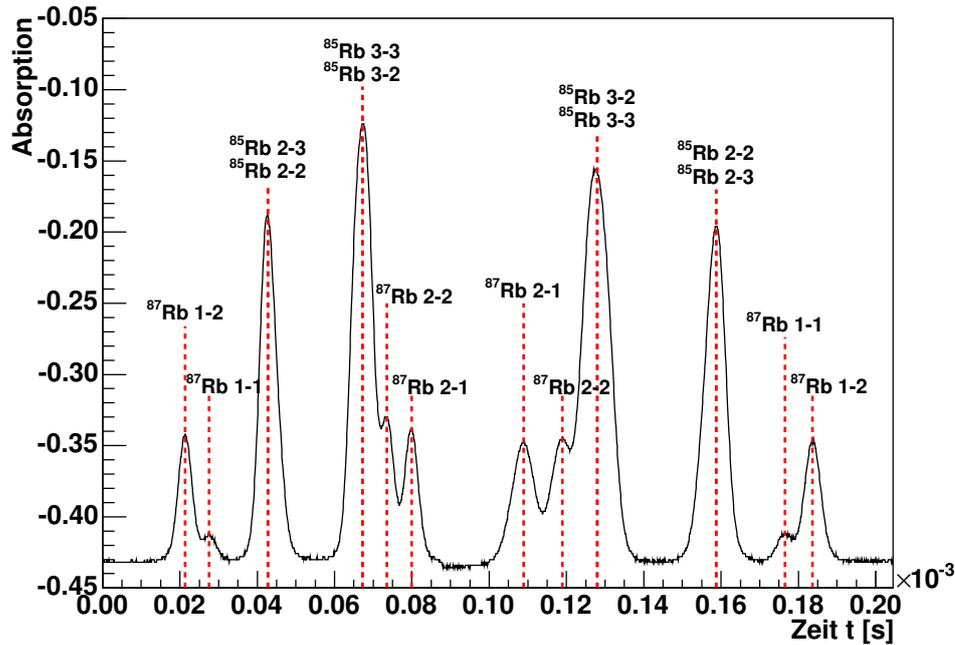


Abbildung 20: Spektrum der Hyperfeinstruktur

Das Hyperfeinstrukturspektrum

Mit Hilfe der Frequenzzeichnung ließen sich die einzelnen Absorptionspeaks im aufgenommenen Hyperfeinstrukturspektrum identifizieren (siehe Abbildung 20). Dazu ermittelten wir die zeitliche Position der Peaks und berechneten daraus die relative Zeit zum ersten Peak, jeweils getrennt für die linke und die rechte Seite. Dabei handelt es sich auch hier bei der linken Seite um das mit steigender Dreiecksspannung (bzw. steigender Frequenz des Lasers) aufgenommene Spektrum und bei der rechten Seite um das mit fallender Dreiecksspannung (bzw. sinkender Laserfrequenz) aufgenommene Spektrum. Zur Identifikation der Peaks rechneten wir mit Hilfe der Eichung die relativen Zeiten in Frequenzen um, und setzten den ersten Peak auf die im Praktikumsordner [1] relativ zur D_1 -Linie angegebene theoretische Frequenz. Wie man in Abbildung 20 sieht, ließen sich 6 der theoretisch 8 Linien auflösen. Während beim ^{87}Rb -Isotop alle Linien sichtbar sind, erscheinen bei ^{85}Rb jeweils die Übergänge $F : 2 \rightarrow 3$ und $F : 2 \rightarrow 2$, sowie die Übergänge $F : 3 \rightarrow 3$ und $F : 3 \rightarrow 2$ als eine Linie. Während wir für die abgelesene Zeit der Peaks von ^{87}Rb einen Fehler von 0,1 bis 0,2 μs abschätzten, nahmen wir deshalb für die Lage der Peaks von ^{85}Rb einen Fehler von 1,0 μs an. Wie auch schon bei der Frequenzzeichnung berechnet sich der Fehler der Zeitdifferenz $\Delta t = |t_1 - t_2|$ durch

$$s_{\Delta t} = \sqrt{s_{t_1}^2 + s_{t_2}^2} ,$$

und pflanzt sich bei der Umrechnung $\nu = f \cdot \Delta t$ in die relative Frequenz wie folgt fort:

$$s_\nu = |\nu| \sqrt{\left(\frac{s_f}{f}\right)^2 + \left(\frac{s_{\Delta t}}{\Delta t}\right)^2}.$$

Die Ergebnisse sind für die linke Seite in Tabelle 4 und für die rechte Seite in Tabelle 5 aufgeführt. Für die linke Seite stimmen die experimentell ermittelten Frequenzen der ^{87}Rb -Übergänge sehr gut mit den Literaturwerten aus [1] überein. Der fortgepflanzte Fehler der Frequenzen erscheint bei Differenzen des experimentellen Werts zum Literaturwert von teilweise gerade mal $0,1\sigma$ etwas zu groß zu sein, jedoch konnten wir trotz wiederholter Überprüfung keine der angenommenen Unsicherheiten der Messwerte verringern. Die Frequenzen der ^{85}Rb -Peaks weichen stärker von den Literaturwerten ab, da es sich bei den Peaks um Überlagerungen von jeweils zwei Übergängen handelt, die sich leider nicht auflösen lassen. Die beiden ermittelten Frequenzen jedoch, stimmen aufgrund der größeren angenommenen Ableseungenauigkeit trotzdem noch mit einer Abweichung von unter 2σ mit den vier theoretischen Übergangsfrequenzen überein.

Peak	abgel. Zeit [μs]	Freq. [GHz]	Literatur [GHz]	Diff [GHz]
^{87}Rb , F: 1-2	$21,3 \pm 0,1$	$+4,58 \pm 0,00$	+4,58	+0,000
^{87}Rb , F: 1-1	$27,5 \pm 0,2$	$+3,77 \pm 0,03$	+3,76	+0,010
^{85}Rb , F: 2-3	$42,7 \pm 1,0$	$+1,79 \pm 0,14$	+1,92	-0,134
^{85}Rb , F: 2-2	$42,7 \pm 1,0$	$+1,79 \pm 0,14$	+1,56	+0,226
^{85}Rb , F: 3-3	$67,2 \pm 1,0$	$-1,41 \pm 0,17$	-1,12	-0,294
^{85}Rb , F: 3-2	$67,2 \pm 1,0$	$-1,41 \pm 0,17$	-1,48	+0,066
^{87}Rb , F: 2-2	$73,5 \pm 0,2$	$-2,24 \pm 0,13$	-2,25	+0,014
^{87}Rb , F: 2-1	$79,9 \pm 0,1$	$-3,07 \pm 0,15$	-3,07	-0,002

Tabelle 4: Abgelesene Zeit der Absorptionspeaks im HFS-Spektrum und die bestimmten Frequenzen relativ zur D_1 -Linie, sowie deren Literaturwerte und die Differenz zwischen Mess- und Literaturwerten (linke Seite)

Peak	abgel. Zeit [μs]	Freq. [GHz]	Literatur [GHz]	Diff [GHz]
^{87}Rb , F: 2-1	$108,9 \pm 0,1$	$-3,07 \pm 0,00$	-3,07	+0,000
^{87}Rb , F: 2-2	$118,9 \pm 0,2$	$-1,89 \pm 0,03$	-2,25	+0,361
^{85}Rb , F: 3-2	$128,0 \pm 1,0$	$-0,81 \pm 0,12$	-1,48	+0,667
^{85}Rb , F: 3-3	$128,0 \pm 1,0$	$-0,81 \pm 0,12$	-1,12	+0,307
^{85}Rb , F: 2-2	$158,7 \pm 1,0$	$+2,81 \pm 0,16$	+1,56	+1,254
^{85}Rb , F: 2-3	$158,7 \pm 1,0$	$+2,81 \pm 0,16$	+1,92	+0,894
^{87}Rb , F: 1-1	$176,5 \pm 0,2$	$+4,92 \pm 0,14$	+3,76	+1,156
^{87}Rb , F: 1-2	$183,7 \pm 0,1$	$+5,77 \pm 0,15$	+4,58	+1,187

Tabelle 5: Abgelesene Zeit der Absorptionspeaks im HFS-Spektrum und die bestimmten Frequenzen relativ zur D_1 -Linie, sowie deren Literaturwerte und die Differenz zwischen Mess- und Literaturwerten (rechte Seite)

Die in Tabelle 5 aufgeführten Ergebnisse für das HFS-Spektrum bei fallender Laserfrequenz (rechte Seite) weichen dagegen stark von den Literaturwerten ab. Die experimentellen Werte unterscheiden sich zum Teil um über 10 Standardabweichungen von den angegebenen Werten. Dass die gemessenen Peaks nicht mit der Theorie übereinstimmen können, lässt sich schon in Abbildung 20 erkennen: Man sieht deutlich, dass die Abstände zwischen den Peaks nach rechts hin kleiner werden. So sollte für ^{87}Rb der Abstand zwischen den Übergängen $F : 2 \rightarrow 1$ und $F : 2 \rightarrow 2$ gleich dem Abstand zwischen den Übergängen $F : 1 \rightarrow 1$ und $F : 1 \rightarrow 2$ sein, in der Abbildung ist dagegen der erste Abstand wesentlich größer als der zweite. Umso verwunderlicher ist dieses Verhalten, da das die Laserfrequenz modulierende Dreieckssignal einen linearen Verlauf aufweist (siehe Abbildungen 31 und 32), die Frequenz des Lasers muss sich also bei linearer Modulationsspannung nichtlinear ändern. Ein eventuell auftretender Modensprung kann jedoch ausgeschlossen werden, da der Effekt nur in einer Scanrichtung auftritt.

Wie wir bereits bei der Frequenzzeichnung geschrieben haben, haben wir keine genaue Erklärung für dieses Phänomen. Da aber hier deutlich wird, dass der nichtlineare Zusammenhang zwischen Modulationssignal und Laserfrequenz erst nach der Änderung des Vorzeichens der Steigung eintritt und es offensichtlich vorher einen linearen Zusammenhang gegeben hat (die oben zitierten Abstände der Übergänge von ^{87}Rb stimmen auf der linken Seite überein), liegt die Vermutung nahe, dass es sich hierbei um Hysterese-Effekte im Lasernetzgerät handeln könnte.

Bestimmung der Intervallkonstanten

Aufgrund der im vorangegangenen Abschnitt erläuterten Schwierigkeiten, die bei der Messung mit negativer Scanrichtung (rechte Seite von Abbildung 20) auftraten, verwendeten wir zu Bestimmung der Intervallkonstanten der Hyperfeinstruktur nur die Messwerte der positiven Scanrichtung (linke Seite). Die Intervallkonstante A eines Zustands lässt sich nach Gleichung (16) durch

$$A = \frac{h \Delta\nu}{F + 1}$$

bestimmen, wobei $\Delta\nu$ die Frequenzdifferenz zwischen zwei benachbarten HFS-Niveaus eines Zustands und F die Quantenzahl des Unteren der beiden Niveaus ist. Die Differenzfrequenz $\Delta\nu$ lässt sich durch die Differenz zweier geeigneter Übergänge im aufgenommenen HFS-Spektrum bestimmen (siehe dazu Abbildung 20, sowie Abbildungen 8 und 11 in Abschnitt 2.4). Um durch die Frequenzzeichnung hervorgerufene Korrelationen zu vermeiden, gingen wir bei der Bildung der Differenzen von den gemessenen Zeitwerten (siehe Tabelle 4) aus und berechneten erst nach Bestimmung der Zeitdifferenz Δt die Frequenzdifferenz $\Delta\nu$ mit dem Eichfaktor f . Die Fehler bestimmten wir analog zum vorangegangenen Abschnitt. Für den Fehler der Intervallkonstante ergibt sich damit:

$$s_A = \left| A \frac{s_{\Delta\nu}}{\Delta\nu} \right| .$$

Die Intervallkonstante für den Grundzustand ${}^2S_{1/2}$ von ${}^{87}\text{Rb}$ bestimmten wir zum Einen durch die Differenz der Übergänge $F : 1 \rightarrow 2$ und $F : 2 \rightarrow 2$, und zum Anderen durch $F : 1 \rightarrow 1$ und $F : 2 \rightarrow 1$. Als Ergebnis erhielten wir dabei:

$$A_{S_{1/2},1} = (14,10 \pm 0,27) \cdot 10^{-6} \text{ eV}$$

$$A_{S_{1/2},2} = (14,15 \pm 0,27) \cdot 10^{-6} \text{ eV}$$

Als gewichtetes Mittel ergibt sich daraus:

$$A_{S_{1/2}} = (14,12 \pm 0,19) \cdot 10^{-6} \text{ eV} .$$

Dieser Wert stimmt sehr gut mit dem in [1] angegebenen Literaturwert von $A_{S_{1/2}} = 14,13 \cdot 10^{-6} \text{ eV}$ überein.

Die Intervallkonstante des ersten angeregten Zustands ${}^2P_{1/2}$ von ${}^{87}\text{Rb}$ bestimmten wir mit den Übergängen $F : 1 \rightarrow 2$ und $F : 1 \rightarrow 1$, sowie $F : 2 \rightarrow 2$ und $F : 2 \rightarrow 1$. Damit erhielten wir:

$$A_{P_{1/2},1} = (1,674 \pm 0,068) \cdot 10^{-6} \text{ eV}$$

$$A_{P_{1/2},2} = (1,728 \pm 0,069) \cdot 10^{-6} \text{ eV}$$

und als gewichtetes Mittel:

$$A_{P_{1/2}} = (1,701 \pm 0,048) \cdot 10^{-6} \text{ eV} .$$

Auch dieser Wert stimmt im Bereich seines Fehlers sehr gut mit dem angegebenen Literaturwert von $A_{P_{1/2}} = 1,692 \cdot 10^{-6} \text{ eV}$ überein.

Da bei ${}^{85}\text{Rb}$ jeweils zwei der Übergänge so nah beieinander liegen, dass sie nicht einzeln aufgelöst werden können, lassen sich nicht beide Intervallkonstanten bestimmen. Es ist jedoch möglich unter Annahme eines etwas größeren Fehlers für die Lage der Peaks zumindest die Intervallkonstante des Grundzustands ${}^2S_{1/2}$ zu berechnen. Als Ergebnis erhielten wir dabei:

$$A_{S_{1/2}} = (4,410 \pm 0,268) \cdot 10^{-6} \text{ eV}$$

was mit einem relativen Fehler von 6% recht ungenau ist, aber immerhin im Bereich einer Standardabweichung mit dem Literaturwert von $A_{S_{1/2}} = 4,185 \cdot 10^{-6} \text{ eV}$ übereinstimmt.

4.3. Doppelresonanz

In diesem Versuchsteil sollte mit Hilfe der Doppelresonanz das äußere horizontale Magnetfeld B_{hor} (im Folgenden auch als horizontales Erdmagnetfeld bezeichnet) am Ort der Rubidiumzelle sowie die Kernspinquantenzahlen der Isotope ^{85}Rb und ^{87}Rb bestimmt werden.

Für schwache Magnetfelder sind die aufgespaltenen Zeeman-Niveaus in guter Näherung äquidistant. Für die beiden Rb-Isotope gilt nach Gleichung (23) für die Energiedifferenz zweier benachbarter Zeeman-Niveaus im Grundzustand mit $g_F = 2$:

$$\Delta E_B = \frac{\mu_B B}{I + \frac{1}{2}}. \quad (38)$$

Der Kernspin lässt sich also bei bekannter Magnetfeldstärke durch Messung dieses energetischen Abstands bestimmen.

Bei der Methode der Doppelresonanz wird der Laser auf die Resonanzfrequenz eines der beiden Rb-Isotope eingestellt, wodurch es zu Übergängen zwischen den HFS-Termen des Grundzustands und des angeregten Zustands kommt. Da das Licht des Lasers durch das eingesetzte $\lambda/4$ -Plättchen in eine Richtung zirkular polarisiert ist, kommt es zum optischen Zeeman-Pumpen und es wird ein bestimmter Zustand (z.B. handelt es sich bei ^{87}Rb und σ^+ -Licht um den Zustand $^2S_{1/2}$, $F = 2$, $m_F = +2$) auf Kosten der anderen Niveaus des Grundzustands bevölkert. Diese werden immer mehr entleert, bis es zu einem Gleichgewicht zwischen Pumpvorgang und Relaxationsprozessen kommt. Übergänge zwischen Zeeman-Niveaus des gleichen Terms sind nach den Auswahlregeln verboten, es können jedoch durch Einstrahlung geeigneter Radiofrequenzen Übergänge induziert werden. In Gleichung (38) sieht man, dass die Aufspaltungsenergie der Zeeman-Niveaus proportional zur Magnetfeldstärke ist. Es lässt sich also auch bei fest eingestellter Radiofrequenz ein geeignetes Magnetfeld finden, so dass Übergänge zwischen den Zeeman-Niveaus induziert werden.

Als Realisierung verwendeten wir eine Spule zur Erzeugung eines horizontalen Magnetfelds, welches mit einer Frequenz von 50 Hz sinusförmig moduliert wurde. Gleichzeitig strahlten wir mit einem RF-Sender eine feste Radiofrequenz von etwa $\nu \approx 500 \text{ kHz}$ in die Zelle ein. Dadurch wird erreicht, dass es bei der Magnetfeldstärke, die nach (38) zur Energie der Radiofrequenz passt, zu induzierten Übergängen kommt, und die durch Zeeman-Pumpen hervorgerufene Übervölkerung des einen Zustands abgebaut werden kann. Da die anderen Zeeman-Niveaus des Grundzustands dadurch wieder zu höheren Besetzungszahlen aufweisen, kommt es zu verstärktem Pumpen, in Folge dessen die Transparenz der Zelle abnimmt und an der Photodiode eine geringere Intensität gemessen werden kann.

Die Energiedifferenz $\Delta E_B = h\nu$ zwischen den Zeeman-Niveaus ist durch Festlegung der Radiofrequenz bekannt. Es fehlt aber noch die Magnetfeldstärke bei der es mit dieser RF-Energie zur Resonanz kommt. Dafür lässt sich durch eine

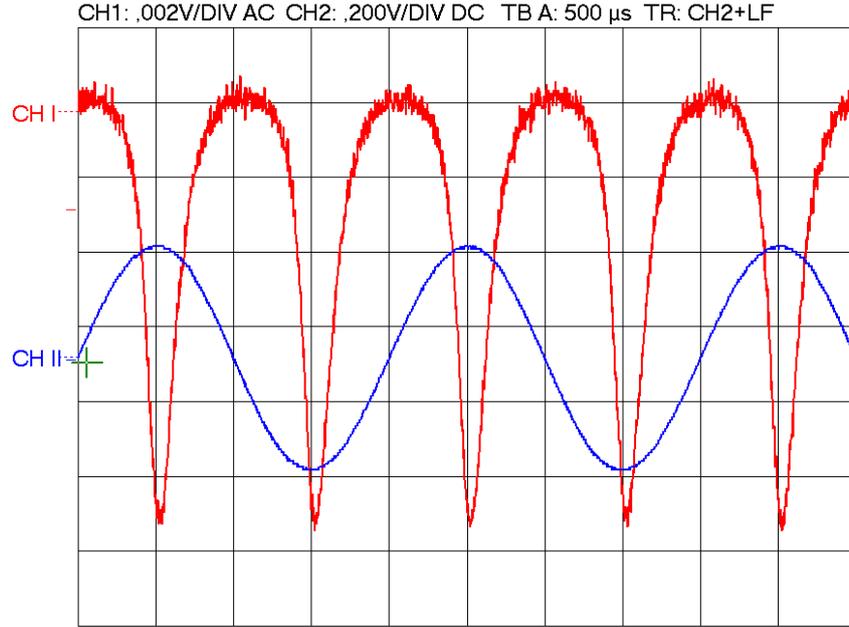


Abbildung 21: Äquidistante Absorptionspeaks und sinusförmiges Modulationssignal der ersten Spule (Anm.: das Modulationssignal und das dadurch erzeugte Magnetfeld sind Phasenverschoben; die Absorptionspeaks und der Nulldurchgang des Sinussignals erscheinen also nicht an der selben Stelle)

weitere Spule ein konstantes horizontales Magnetfeld B_{sp} am Ort der Rb-Zelle erzeugen, welches sich mit dem äußeren horizontalen (Erd-)Magnetfeld B_{hor} überlagert. Durch geeignete Wahl von B_{sp} lässt sich erreichen, dass das resultierende horizontale Magnetfeld B am Ort der Zelle genau dann zur Radiofrequenz passt, wenn beim sinusförmig modulierten Magnetfeld der ersten Spule ein Nulldurchgang vorliegt. In diesem Fall erscheinen die aufgenommenen Absorptionspeaks äquidistant (siehe Abbildung 21). Die passende Magnetfeldstärke B am Zellenort lässt sich durch zwei verschiedene Werte von B_{sp} erreichen. Zum Einen, indem man ein Magnetfeld erzeugt, das in die selbe Richtung wie das äußere Magnetfeld B_{hor} zeigt, und zum Anderen indem man den Spulenstrom umpolt und dadurch ein B_{hor} entgegen gerichtetes Magnetfeld erzeugt. Für die Magnetfeldstärke B am Ort der Zelle erhält man mit den beiden zu äquidistanten Absorptionspeaks führenden konstanten Spulenströmen I_1 und I_2 somit:

$$B = B_{\text{sp}}(I_1) + B_{\text{hor}} \quad \text{und} \quad B = B_{\text{sp}}(I_2) - B_{\text{hor}} .$$

Mit diesen beiden Gleichungen lässt sich das zur Berechnung des Kernspins benötigte Magnetfeld B am Ort der Zelle berechnen:

$$\Leftrightarrow \quad B = \frac{B_{\text{sp}}(I_1) + B_{\text{sp}}(I_2)}{2} = \frac{a (I_1 + I_2)}{2} . \quad (39)$$

Außerdem lässt sich analog dazu auch das horizontale äußere (Erd-)Magnetfeld B_{hor} bestimmen:

$$\begin{aligned} 0 &= (B_{\text{sp}}(I_1) + B_{\text{hor}}) - (B_{\text{sp}}(I_2) - B_{\text{hor}}) \\ \Leftrightarrow B_{\text{hor}} &= \frac{B_{\text{sp}}(I_1) - B_{\text{sp}}(I_2)}{2} = \frac{a(I_1 - I_2)}{2}. \end{aligned} \quad (40)$$

Bei dem in Gleichungen (39) und (40) eingeführte Faktor a handelt es sich um den in [1] angegebenen Eichfaktor für die zur Einstellung des konstanten Magnetfelds verwendete Spule (Spule 1). Er beträgt:

$$a = (7,99 \pm 0,01) \cdot 10^{-4} \text{ T/A}. \quad (41)$$

Die Messung wurde von uns insgesamt zweimal an zwei verschiedenen Tagen durchgeführt. Die Messwerte sind in Tabelle 6 aufgeführt.

Messung	Isotop	Sp. Strom I_1 [A]	Sp. Strom I_2 [A]	Frequenz ν [kHz]
1	^{85}Rb	$0,155 \pm 0,001$	$0,116 \pm 0,001$	$500,22 \pm 0,02$
	^{87}Rb	$0,110 \pm 0,001$	$0,071 \pm 0,001$	$500,20 \pm 0,02$
2	^{85}Rb	$0,150 \pm 0,001$	$0,121 \pm 0,001$	$500,26 \pm 0,02$
	^{87}Rb	$0,105 \pm 0,001$	$0,076 \pm 0,001$	$500,25 \pm 0,02$

Tabelle 6: Die bei den beiden Messungen aufgenommenen Messwerte

Die daraus mit Gleichung (40) bestimmten horizontalen äußeren (Erd-)Magnetfeldstärken zeigt Tabelle 7. Die Fehler $s_{B_{\text{hor}}}$ wurden dabei aus den Messungenauigkeiten s_{I_1} und s_{I_2} der Spulenströme und dem angegebenen Fehler s_a des Eichfaktors a wie folgt berechnet:

$$\begin{aligned} s_{B_{\text{hor}}} &= \sqrt{\left(\frac{\partial B_{\text{hor}}}{\partial a}\right)^2 s_a^2 + \left(\frac{\partial B_{\text{hor}}}{\partial I_1}\right)^2 s_{I_1}^2 + \left(\frac{\partial B_{\text{hor}}}{\partial I_2}\right)^2 s_{I_2}^2} \\ &= |B_{\text{hor}}| \sqrt{\left(\frac{s_a}{a}\right)^2 + \frac{s_{I_1}^2 + s_{I_2}^2}{(I_1 - I_2)^2}}. \end{aligned} \quad (42)$$

Die berechneten Kernspinquantenzahlen I für die beiden Rb-Isotope sind in Tabelle 8 angegeben. Zur Berechnung der Kernspinquantenzahlen wurde zunächst nach Gleichung (39) das Magnetfelds am Ort der Rb-Zelle berechnet, der Fehler dafür lässt sich analog zu (42) bestimmen:

$$s_B = |B| \sqrt{\left(\frac{s_a}{a}\right)^2 + \frac{s_{I_1}^2 + s_{I_2}^2}{(I_1 + I_2)^2}}. \quad (43)$$

Danach ließ sich mit Gleichung (38) die Kernspinquantenzahl berechnen:

$$I = \frac{\mu_B B}{h\nu} - \frac{1}{2}.$$

Für den Fehler erhält man durch gaußsche Fehlerfortpflanzung:

$$s_I = \sqrt{\left(\frac{\partial I}{\partial B}\right)^2 s_B^2 + \left(\frac{\partial I}{\partial \nu}\right)^2 s_\nu^2} = \left(|I| + \frac{1}{2}\right) \sqrt{\left(\frac{s_B}{B}\right)^2 + \left(\frac{s_\nu}{\nu}\right)^2}.$$

Die an beiden Tagen bestimmten äußeren Magnetfelder (Tabelle 7) weisen einen signifikanten Unterschied auf. Der Grund dafür sind vermutlich Störfelder im Gebäude, die zum Beispiel durch andere Geräte hervorgerufen werden können.

Messung	^{85}Rb : B_{hor} [μT]	^{87}Rb : B_{hor} [μT]
1	$15,78 \pm 0,57$	$15,58 \pm 0,57$
2	$11,59 \pm 0,57$	$11,59 \pm 0,57$

Tabelle 7: Das bestimmte horizontale äußere (Erd-)Magnetfeld

Messung	^{85}Rb : Kernspin I	^{87}Rb : Kernspin I
1	$2,524 \pm 0,016$	$1,523 \pm 0,016$
2	$2,529 \pm 0,016$	$1,523 \pm 0,016$

Tabelle 8: Die bestimmten Kernspinquantenzahlen der beiden Rb-Isotope

Die ermittelten Kernspinquantenzahlen (Tabelle 8) stimmen im Bereich ihrer Fehler relativ gut mit den theoretischen Werten

$$I(^{85}\text{Rb}) = \frac{5}{2} \quad \text{und} \quad I(^{87}\text{Rb}) = \frac{3}{2}$$

überein. Es fällt jedoch auf, dass alle Werte etwa gleich weit über den theoretischen Werten liegen, und dass die Übereinstimmung zwischen den Messungen der verschiedenen Tagen sehr genau ist. Dies liegt vermutlich an einem systematischen Fehler des Eichfaktors a von Spule 2; würde dieser um etwa $0,07 \cdot 10^{-4} \text{ T/A}$ (0,9%) niedriger liegen, so würden alle von uns bestimmten Kernspinquantenzahlen wesentlich genauer mit den Theoriewerten übereinstimmen und im schlechtesten Fall um gerade mal 0,006 (0,4%) von der Theorie abweichen.

4.4. Spinpräzession

In diesem Versuchsteil wurde die Spinpräzession für verschiedene in vertikale Magnetfeldstärken gemessen. Dazu wurde dem vertikalen Erdfeld B_{vert} mit Spule 4 ein weiteres vertikales Magnetfeld B überlagert. Ein horizontales Magnetfeld wurde so schnell ein- und wieder ausgeschaltet, dass das Ensemble der Rb-Atome nicht dem schnellen Wechsel folgen konnte, sondern dass sich eine Präzessionsbewegung um die Richtung des Vertikalfeldes einstellte. Für die Kreisfrequenz der Präzession gilt:

$$\omega_L = \frac{g_F \mu_B}{\hbar} (B + B_{\text{vert}}) .$$

Da der Polarisationsvektor dieser Präzession folgt, ändert sich die Transmission des Laserstrahls durch die Rb-Zelle und damit die beobachtete Intensität. Aus der Frequenz ν des aufgenommenen sinusförmigen Absorptionssignals kann das vertikale Erdfeld B_{vert} bestimmt werden:

$$B = \frac{h\nu}{g_F \mu_B} - B_{\text{vert}} . \quad (44)$$

Durch Auftragung des eingestellten Spulenstroms I des vertikalen Kompensationsfelds B als Funktion der gemessenen Präzessionsfrequenz ν lässt sich direkt aus dem Y-Achsenabschnitt I_0 das vertikale Erdfeld B_{vert} berechnen:

$$B_{\text{vert}} = a \cdot I_0 , \quad (45)$$

dabei ist a der angegebene Eichfaktor von Spule 4:

$$a = (4,76 \pm 0,01) \cdot 10^{-4} \text{ T/A} .$$

Der Fehler $s_{B_{\text{vert}}}$ lässt sich aus den Fehlern s_{I_0} des Fits und dem Fehler s_a des Eichfaktors wie folgt berechnen:

$$s_{B_{\text{vert}}} = |s_{B_{\text{vert}}}| \sqrt{\left(\frac{s_{I_0}}{I_0}\right)^2 + \left(\frac{s_a}{a}\right)^2} .$$

Wir führten die Messung sowohl für das ^{85}Rb - als auch das ^{87}Rb -Isotop durch. Eine Auswahl der mit dem Oszilloskop aufgenommenen Kurven ist in Anhang A.2 dargestellt. Die gemessenen Frequenzen und die jeweils eingestellten Spulenströme des vertikalen Kompensationsfelds sind in den Tabellen 9 und 10 aufgeführt. Mit den durch die Fits (siehe Abbildungen 22 und 23) gelieferten Y-Achsenabschnitten erhielten wir mit Gleichung (45) folgende Werte für das vertikale Erdmagnetfeld:

$$\begin{aligned} ^{85}\text{Rb}: \quad B_{\text{vert}} &= (34,33 \pm 0,14) \cdot 10^{-6} \text{ T} \\ ^{87}\text{Rb}: \quad B_{\text{vert}} &= (34,45 \pm 0,22) \cdot 10^{-6} \text{ T} . \end{aligned}$$

Die Ergebnisse der beiden Isotope stimmen sehr gut überein.

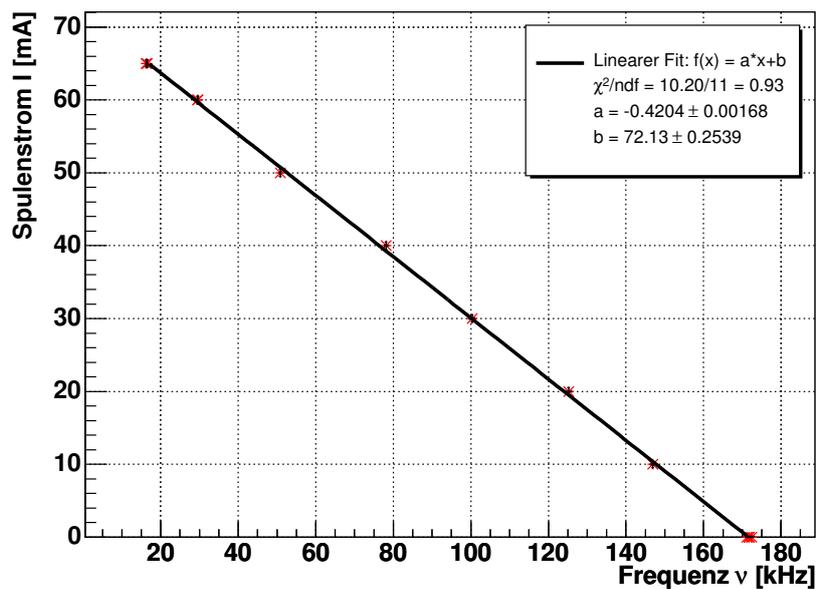


Abbildung 22: Geradenfit durch die Messwerte für ^{85}Rb

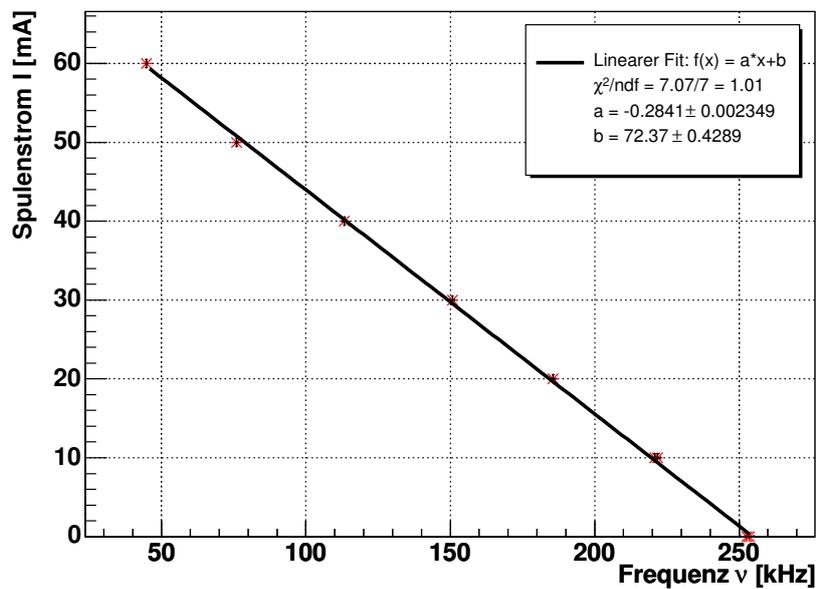


Abbildung 23: Geradenfit durch die Messwerte für ^{87}Rb

Nr	Spulenstrom I [μA]	Zeit t [μs]	Anzahl n	Frequenz ν [kHz]
1	$0,0 \pm 0,0$	$46,75 \pm 0,10$	8	$171,12 \pm 0,37$
2	$10,0 \pm 0,5$	$40,85 \pm 0,10$	6	$146,88 \pm 0,36$
3	$20,0 \pm 0,5$	$31,95 \pm 0,20$	4	$125,20 \pm 0,78$
4	$30,0 \pm 0,5$	$29,90 \pm 0,20$	3	$100,33 \pm 0,67$
5	$40,0 \pm 0,5$	$38,40 \pm 0,20$	3	$78,12 \pm 0,41$
6	$50,0 \pm 0,5$	$59,00 \pm 0,20$	3	$50,85 \pm 0,17$
7	$60,0 \pm 0,5$	$101,50 \pm 0,20$	3	$29,56 \pm 0,06$
8	$60,0 \pm 0,5$	$101,30 \pm 0,20$	3	$29,62 \pm 0,06$
9	$65,0 \pm 0,5$	$184,50 \pm 0,20$	3	$16,26 \pm 0,02$
10	$65,0 \pm 0,5$	$120,50 \pm 0,50$	2	$16,60 \pm 0,07$
11	$0,0 \pm 0,0$	$23,30 \pm 0,10$	4	$171,67 \pm 0,74$
12	$0,0 \pm 0,0$	$29,10 \pm 0,10$	5	$171,82 \pm 0,59$
13	$0,0 \pm 0,0$	$29,00 \pm 0,10$	5	$172,41 \pm 0,59$

Tabelle 9: Messwerte für ^{85}Rb

Nr	Spulenstrom I [μA]	Zeit t [μs]	Anzahl n	Frequenz ν [kHz]
1	$0,0 \pm 0,0$	$15,78 \pm 0,10$	4	$253,57 \pm 1,61$
2	$10,0 \pm 0,5$	$22,55 \pm 0,10$	5	$221,73 \pm 0,98$
3	$10,0 \pm 0,5$	$22,65 \pm 0,10$	5	$220,75 \pm 0,97$
4	$20,0 \pm 0,5$	$21,55 \pm 0,10$	4	$185,61 \pm 0,86$
5	$30,0 \pm 0,5$	$19,90 \pm 0,10$	3	$150,75 \pm 0,76$
6	$40,0 \pm 0,5$	$17,65 \pm 0,10$	2	$113,31 \pm 0,64$
7	$50,0 \pm 0,5$	$52,60 \pm 0,20$	4	$76,05 \pm 0,29$
8	$60,0 \pm 0,5$	$67,00 \pm 0,20$	3	$44,78 \pm 0,13$
9	$0,0 \pm 0,0$	$23,75 \pm 0,20$	6	$252,63 \pm 2,13$

Tabelle 10: Messwerte für ^{87}Rb

4.5. Bestimmung der Relaxationszeit nach Dehmelt

Bei der Bestimmung der Relaxationszeit durch Magnetfeldumkehr lässt sich vom zeitlichen Verlauf des Besetzungsunterschieds in Gleichung (35) ausgehen:

$$n(t) = (n_0 - n_{\max}) \exp\left(-\frac{t}{\tau}\right) + n_{\max} .$$

Zwischen der transmittierten Intensität I und $n(t)$ lässt sich folgender Zusammenhang annehmen:

$$I(t) = I_{\min} + \alpha n(t) ,$$

wobei α eine Proportionalitätskonstante und I_{\min} die minimale transmittierte Intensität ist. Mit Gleichung (35) erhält man damit:

$$I(t) = (I_0 - I_{\max}) \exp\left(-\frac{t}{\tau}\right) + I_{\max} = I_{\max} - \Delta I \exp\left(-\frac{t}{\tau}\right)$$

es ergibt sich also ein Zusammenhang zwischen der gemessenen Intensität und der Orientierungszeit τ . Dabei ist I_{\max} die maximal messbare Intensität, die im Fall des Gleichgewichts zwischen Relaxations- und Pumpvorgang, also bei einem maximalen Besetzungsunterschied n_{\max} gemessen werden kann, und ΔI die Differenz aus I_{\max} und der Intensität I_0 bei Magnetfeldumkehr. Mit dieser Funktion führten wir, für die bei unterschiedlichen eingestrahelten Intensitäten aufgenommenen Messungen, Fits durch (siehe Anhang A.3). Da die Orientierungszeit τ über Gleichung (33) mit der Pumpzeit T_P verknüpft ist und diese nach Gleichung (28) über $T_P = 1/(\alpha I)$ mit der eingestrahelten Intensität zusammenhängt, geht bei abnehmender Intensität die reziproke Pumpzeit $1/T_P$ gegen Null. Dieser Zusammenhang lässt sich ausnutzen um die Relaxationszeit T_R zu bestimmen. Nach Gleichung (33) gilt für die reziproke Orientierungszeit:

$$\lim_{I \rightarrow 0} \frac{1}{\tau} = \lim_{I \rightarrow 0} \left(\frac{1}{T_P} + \frac{1}{T_R} \right) = \frac{1}{T_R}$$

Durch Auftragen der reziproken Orientierungszeit $1/\tau$ gegen die eingestrahelte Intensität lässt sich also aus dem Y-Achsenabschnitt eines Geradenfits die reziproke Relaxationszeit $1/T_R$ ablesen.

Um bei den Fits der einzelnen Messungen ein einigermaßen aussagekräftiges χ^2 zu erhalten standen wir vor dem Problem einen Fehler auf die Spannungswerte abzuschätzen. Die aus der Oszilloskop-Software exportierten Daten haben die Form von drei Dezimalstellen gefolgt von einer Zehnerpotenz, jedoch fiel uns auf, dass dabei nicht alle Zwischenwerte angenommen werden. Bei Betrachtung der ursprünglichen Binärdaten der Messungen stellte sich heraus, dass zur Speicherung jedes Messpunkts nur ein Byte verwendet wird, die Spannungsauflösung mit maximal 256 verschiedenen Werten also noch geringer ist, als wir vermutet hatten. Aufgrund der geringen Auflösung ließ sich der Fehler bei allen aufgenommenen Messungen durch den minimalen Abstand zweier Messpunkte einer Messreihe ermitteln (siehe dazu auch `oszi_messung.py` in Anhang B.12).

Neutralfilter	rel. Intensität I_{rel}	Orientierungszeit τ [ms]	τ^{-1} [(ms) $^{-1}$]
leer	$1,000 \pm 0,000$	$0,152 \pm 0,003$	$6,598 \pm 0,123$
(-)	$0,827 \pm 0,007$	$0,191 \pm 0,002$	$5,246 \pm 0,060$
(1)	$0,503 \pm 0,006$	$0,276 \pm 0,002$	$3,625 \pm 0,021$
-0,37	$0,399 \pm 0,006$	$0,315 \pm 0,002$	$3,175 \pm 0,021$
-0,48	$0,370 \pm 0,006$	$0,372 \pm 0,003$	$2,690 \pm 0,020$
-1,03	$0,087 \pm 0,006$	$1,013 \pm 0,006$	$0,987 \pm 0,006$
(1+48)	$0,185 \pm 0,006$	$0,607 \pm 0,003$	$1,646 \pm 0,008$
1,6	$0,098 \pm 0,006$	$1,044 \pm 0,006$	$0,958 \pm 0,005$

Tabelle 11: Aus den Exponentialfits bestimmte Orientierungszeiten τ und die reziproken Orientierungszeiten τ^{-1} bei relativen Intensitäten I_{rel}

Die Ergebnisse der Fits für verschiedene Neutralfilter sind in Tabelle 11 aufgeführt. Die zu den Neutralfiltern verwendeten relativen Intensitäten stammen aus der von uns durchgeführten Eichung in Abschnitt 4.1 (siehe Tabelle 3). Die gegen die relative Intensität aufgetragene reziproke Orientierungszeit in Abbildung 24 dargestellt. Der linearen Zusammenhang lässt sich erkennen; das recht

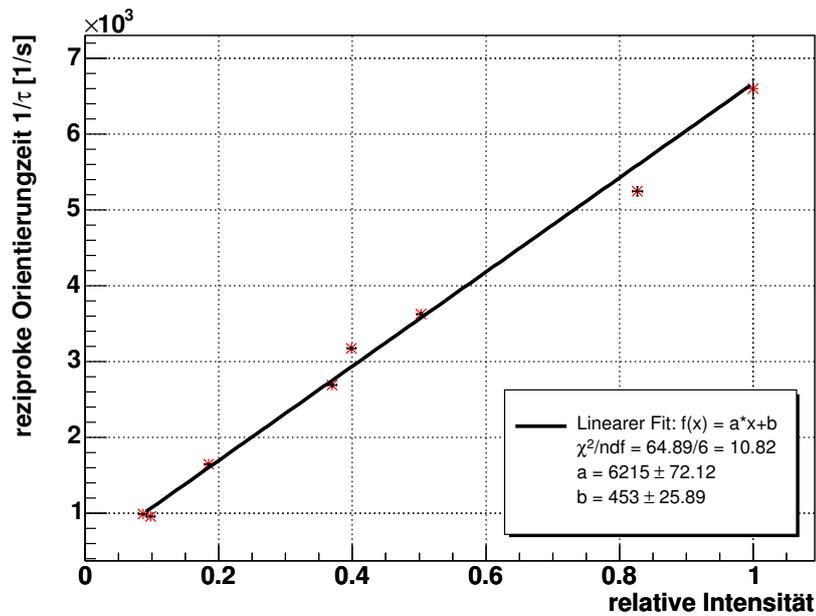


Abbildung 24: Reziproke Orientierungszeit als Funktion der relativen Intensität

hohe reduzierte χ^2 von über 10 liegt an den beiden Ausreißern, die vermutlich auf eine ungenaue Eichung oder auf Temperaturschwankungen zurückzuführen sind. Aus dem Y-Achsenabschnitt b des linearen Fits erhielten wir als Ergebnis für die Relaxationszeit:

$$T_R = (2,21 \pm 0,13) \text{ ms} .$$

4.6. Bestimmung der Relaxationszeit nach Franzen

Die Relaxationszeit T_R lässt sich bei Franzens Methode der Beobachtung der Relaxation "im Dunkeln" direkt bestimmen, indem man die Dunkelzeit variiert und dann ermittelt wie weit das Ensemble nach dieser Zeit relaxiert ist. Ist das Pumplicht für eine Zeit Δt abgeschaltet, so laufen nur noch Relaxationsprozesse ab und der Besetzungsunterschied n folgt dem in Gleichung (30) angegebenen Verlauf:

$$n(t) = n_0 \exp\left(-\frac{t}{T_R}\right).$$

Für die Intensität $I(\Delta t)$ nach einer Dunkelzeit von Δt gilt:

$$I(\Delta t) = I_{\min} + (I_{\max} - I_{\min}) \exp\left(-\frac{\Delta t}{T_R}\right),$$

wobei I_{\max} die gemessene Intensität bei maximalem Besetzungsunterschied ist und I_{\min} die gemessene Intensität bei vollständiger Relaxation, also $n = 0$, ist (siehe Abbildung 25). Mit zunehmender Dunkelzeit erreicht man so, dass sich,

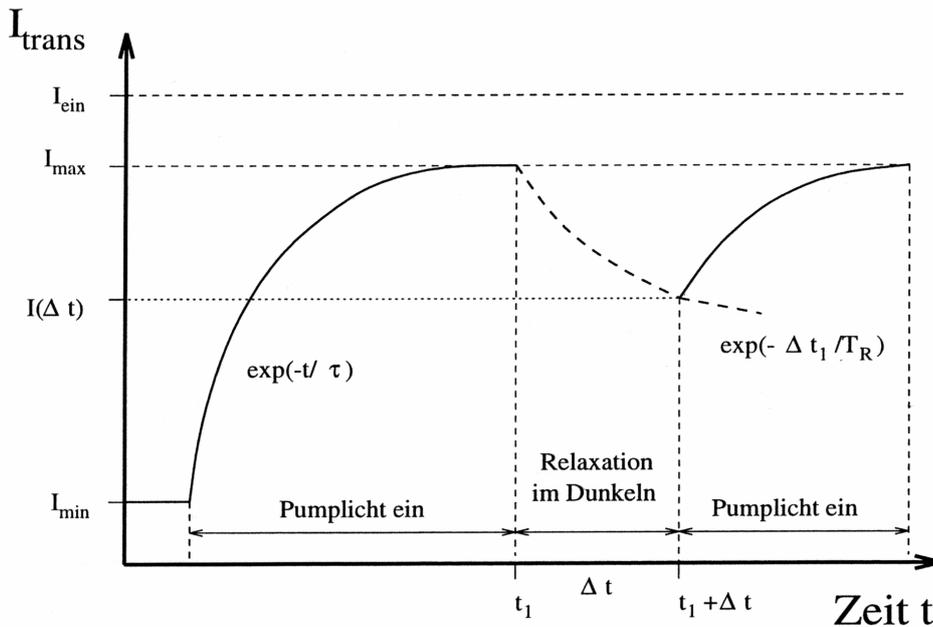


Abbildung 25: Relaxation nach Franzen [1]

bei wieder einsetzendem Pumpen durch erneut eingestrahktes Pumplicht, der Besetzungsunterschied immer weiter ausgleicht und dass dadurch der Orientierungsprozess bei kleiner werdenden Besetzungsunterschieden, also niedrigeren gemessenen Intensitäten $I(\Delta t)$ startet. Lässt sich der Abstand zwischen I_{\max} und $I(\Delta t)$ nach verschiedenen Dunkelzeiten messen, so kann man bei Auftragung dieser Abstände gegen die Dunkelzeit aus dem exponentiell abfallenden

Verlauf direkt die Zeitkonstante, also die Relaxationszeit T_R bestimmen (siehe Abbildung 26).

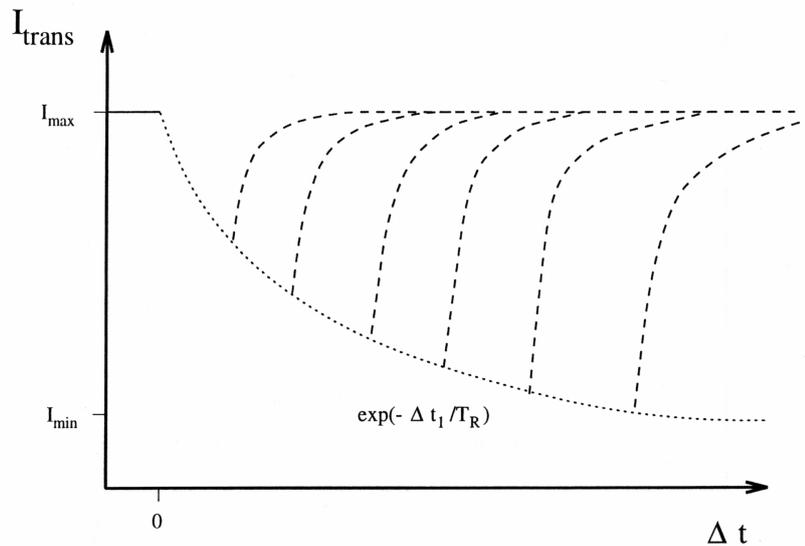


Abbildung 26: Absorptionssignale nach verschiedenen Dunkelzeiten [1]

Es stellte sich jedoch heraus, dass sich das Ablesen der Intensität $I(\Delta t)$, bei der der Orientierungsprozess nach der Dunkelzeit wieder einsetzt, als problematisch gestaltete. In den Abbildungen 27 und 28 sind die Absorptionssignale für einen Teil der insgesamt 15 aufgenommenen Messungen bei verschiedenen Dunkelzeiten dargestellt. Wie man sieht kann man keine sinnvollen Startzeiten für den Orientierungsprozess erkennen, geschweige denn einen mit zunehmender Dunkelzeit exponentiell abfallenden Verlauf der Startintensitäten $I(\Delta t)$. Es scheint sogar, dass der Orientierungsprozess, im Gegensatz zur Theorie, mit zunehmender Dunkelzeit bei immer höheren Werten von $I(\Delta t)$ beginnt.

Da es uns also nicht möglich war, vernünftige Werte von $I(\Delta t)$ zu ermitteln, konnten wir die Relaxationszeit nach dieser Methode leider nicht bestimmen.

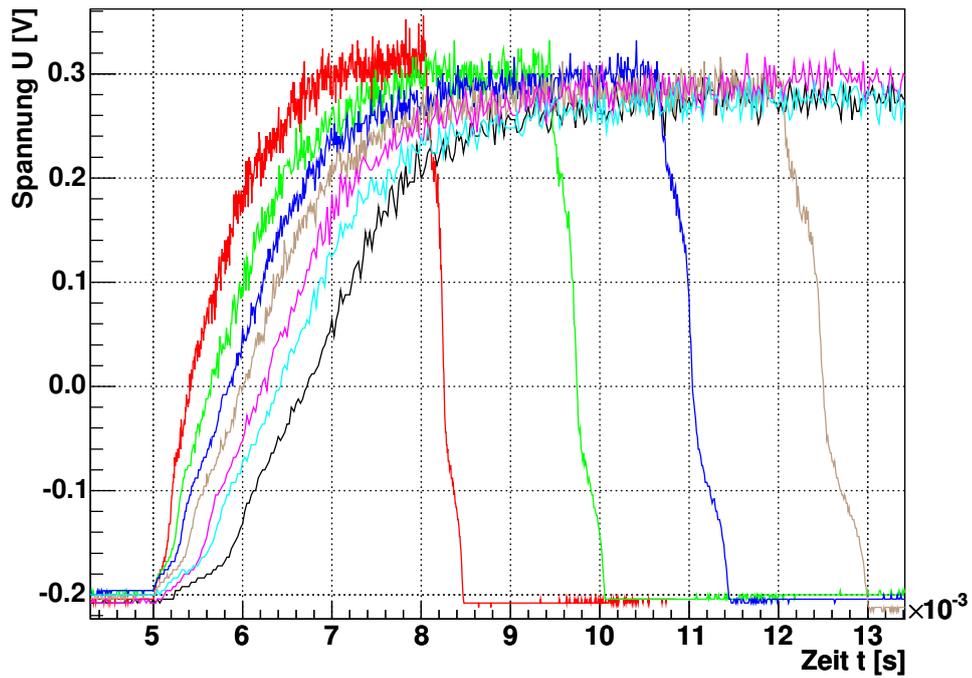


Abbildung 27: Verlauf der Intensität nach verschiedenen Dunkelzeiten

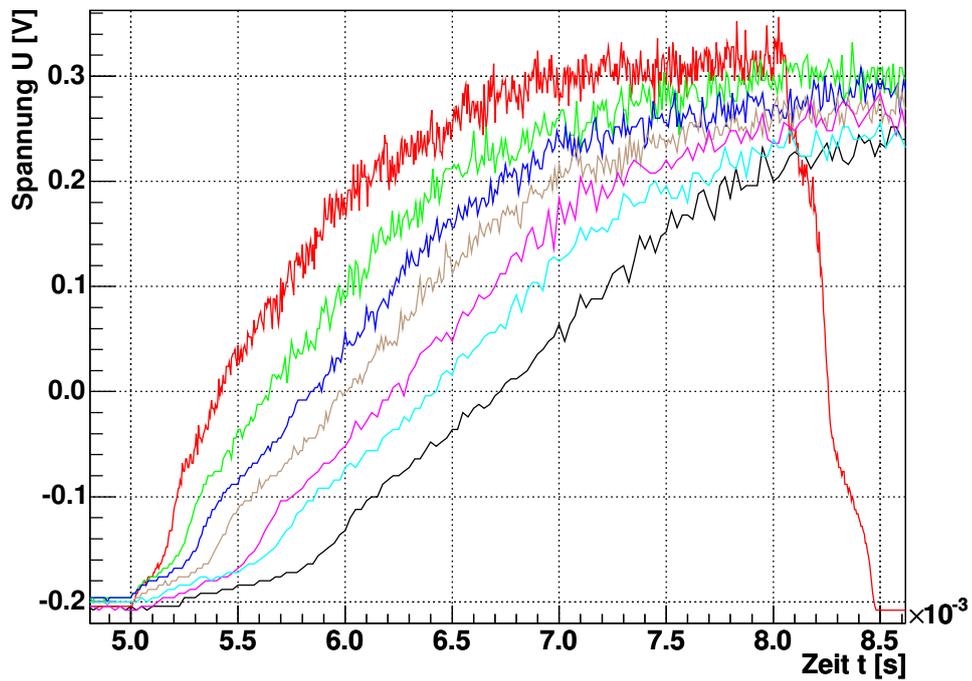


Abbildung 28: Intensität nach verschiedenen Dunkelzeiten (Vergrößerung)

5. Zusammenfassung

Neutralfilter

Da wir die angegebenen Abschwächwerte der Neutralfilter leider nicht bestätigen konnten, führten wir eine zusätzliche Bestimmung der relativen Intensitäten und damit der Abschwächwerte durch. Für die verschiedenen Neutralfilter ergaben sich folgende Werte:

Bezeichnung	Intensität $I[V]$	relative Intensität I_{rel}	Filter x
leer	$1,384 \pm 0,008$	$1,0000 \pm 0,0000$	$0,000 \pm 0,000$
(-)	$1,144 \pm 0,008$	$0,8266 \pm 0,0075$	$-0,190 \pm 0,009$
(1)	$0,696 \pm 0,008$	$0,5029 \pm 0,0065$	$-0,687 \pm 0,013$
-0,37	$0,552 \pm 0,008$	$0,3988 \pm 0,0062$	$-0,919 \pm 0,016$
-0,48	$0,512 \pm 0,008$	$0,3699 \pm 0,0062$	$-0,994 \pm 0,017$
-1,03	$0,120 \pm 0,008$	$0,0867 \pm 0,0058$	$-2,445 \pm 0,067$
1,6	$0,136 \pm 0,008$	$0,0983 \pm 0,0058$	$-2,320 \pm 0,059$
(1+48)	$0,256 \pm 0,008$	$0,1850 \pm 0,0059$	$-1,688 \pm 0,032$

Hyperfeinstruktur der beiden Rubidium-Isotope

Die Hyperfeinstruktur der beiden Isotope ließ sich gut aufnehmen. Allerdings waren nur 6 anstelle der 8 theoretischen Linien sichtbar, da die Linien von ^{85}Rb nicht einzeln auflösbar waren. So überlagerten sich jeweils zwei der vier Linien und es ließ sich daher bei ^{85}Rb nur die Intervallkonstante des Grundzustands ermitteln. Die für beide Isotope bestimmten Intervallkonstanten sind:

$$\text{Für } ^{85}\text{Rb: } A_{S_{1/2}} = (4,410 \pm 0,268) \cdot 10^{-6} \text{ eV}$$

$$\begin{aligned} \text{Für } ^{87}\text{Rb: } A_{S_{1/2}} &= (14,12 \pm 0,19) \cdot 10^{-6} \text{ eV} \\ \text{und } A_{P_{1/2}} &= (1,701 \pm 0,048) \cdot 10^{-6} \text{ eV} \end{aligned}$$

Diese Werte stimmen gut mit den in [1] angegebenen Literaturwerten überein.

Doppelresonanz

Mit der Methode der Doppelresonanz bestimmten wir die Horizontalkomponente des Erdmagnetfelds und den Kernspin der beiden Rb-Isotope. Diese Messung führten wir an zwei verschiedenen Tagen durch. Für das Magnetfeld erhielten wir:

Messung	$^{85}\text{Rb: } B_{\text{hor}} [\mu\text{T}]$	$^{87}\text{Rb: } B_{\text{hor}} [\mu\text{T}]$
1	$15,78 \pm 0,57$	$15,58 \pm 0,57$
2	$11,59 \pm 0,57$	$11,59 \pm 0,57$

Der Grund für den signifikanten Unterschied der an verschiedenen Tagen gemessenen Magnetfelder sind vermutlich Störfelder im Gebäude, die z.B. durch andere Geräte hervorgerufen werden können.

Für den Kernspin der beiden Isotope ergab sich aus den beiden Messungen:

Messung	^{85}Rb : Kernspin I	^{87}Rb : Kernspin I
1	$2,524 \pm 0,016$	$1,523 \pm 0,016$
2	$2,529 \pm 0,016$	$1,523 \pm 0,016$

Diese Werte stimmen im Bereich ihrer Fehler relativ gut mit den theoretischen Werten für ^{85}Rb und ^{87}Rb von $5/2$ bzw. $3/2$ überein. Da alle bestimmten Werte etwa gleich weit über den theoretischen Werten liegen, liegt hier vermutlich ein systematischer Fehler vor, der höchstwahrscheinlich auf den Eichfaktor der Spule zurückzuführen ist.

Spinpräzession

Durch Beobachtung der Spinpräzession beider Isotope konnte die vertikale Komponente des Erdmagnetfelds ermittelt werden. Dabei erhielten wir folgende Ergebnisse:

$$\begin{aligned} ^{85}\text{Rb}: \quad B_{\text{vert}} &= (34,33 \pm 0,14) \cdot 10^{-6} \text{ T} \\ ^{87}\text{Rb}: \quad B_{\text{vert}} &= (34,45 \pm 0,22) \cdot 10^{-6} \text{ T} . \end{aligned}$$

Die Ergebnisse der beiden Messreihen stimmen sehr gut überein.

Relaxationszeiten

Wir führten die Messung der Relaxationszeit mit Dehmelts Methode der Magnetfeldumkehr und mit Franzens Methode der Relaxation im Dunkeln durch.

Bei der Methode nach Dehmelt erhielten wir als Relaxationszeit:

$$T_R = (2,21 \pm 0,13) \text{ ms} .$$

Mit der Methode nach Franzens war es uns leider nicht möglich die Relaxationszeit zu bestimmen, da sich aus den Messwerten keine sinnvollen Startintensitäten für den Orientierungsprozess ablesen ließen.

Literatur

- [1] Clemens Baur. Einrichtung des Versuchs "Optisches Pumpen mit Laserdioden". Albert-Ludwigs-Universität, Freiburg im Breisgau, August 1997.

A. Weitere Grafiken

A.1. Spektroskopie der Hyperfeinstruktur

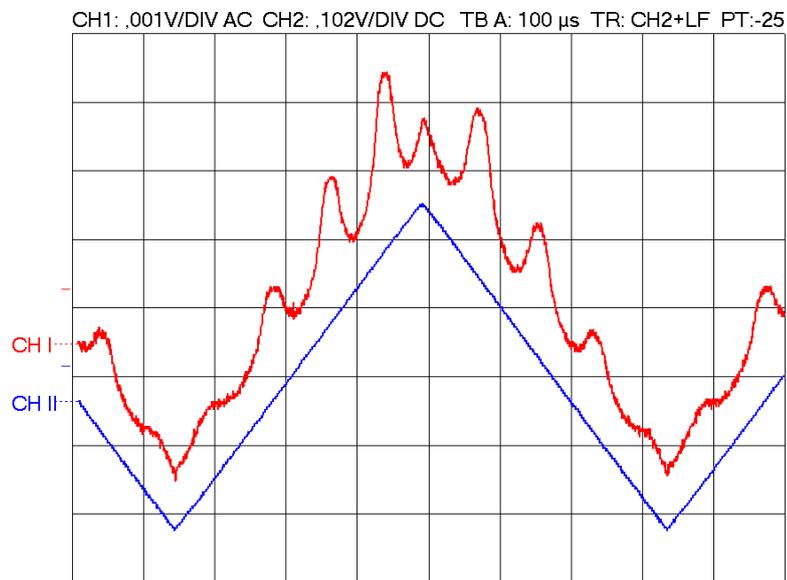


Abbildung 29: Interferenzmaxima des Etalons und die zur Lasermodulation verwendete Dreiecksspannung

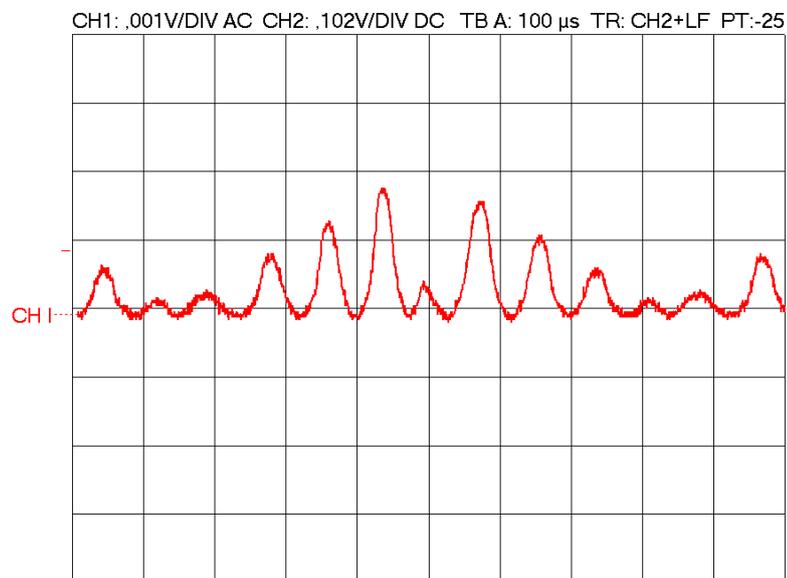


Abbildung 30: Interferenzmaxima des Etalons mit abzogener Dreiecksspannung

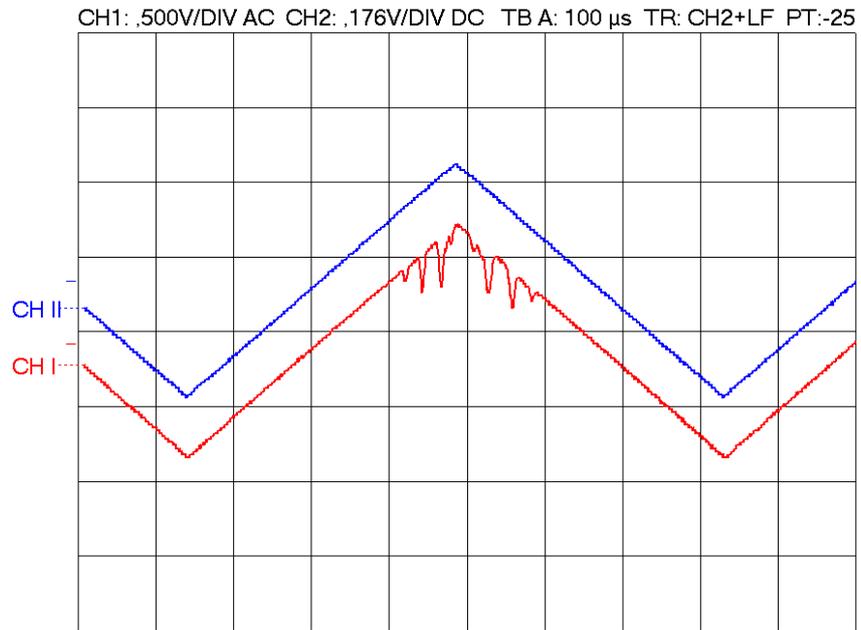


Abbildung 31: Das HFS-Spektrum und das volle zur Lasermodulation verwendete Dreiecksignal

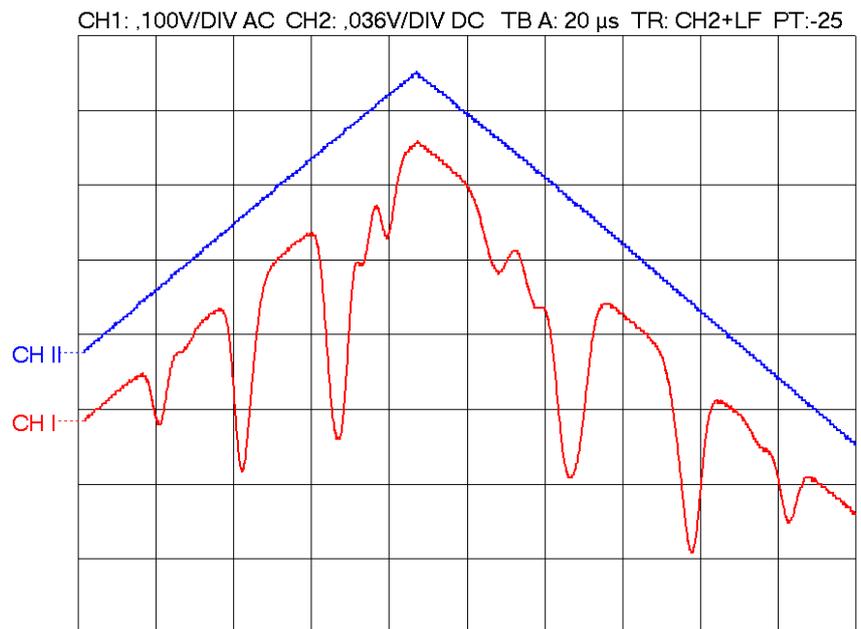


Abbildung 32: Vergrößerter Ausschnitt des HFS-Spektrum

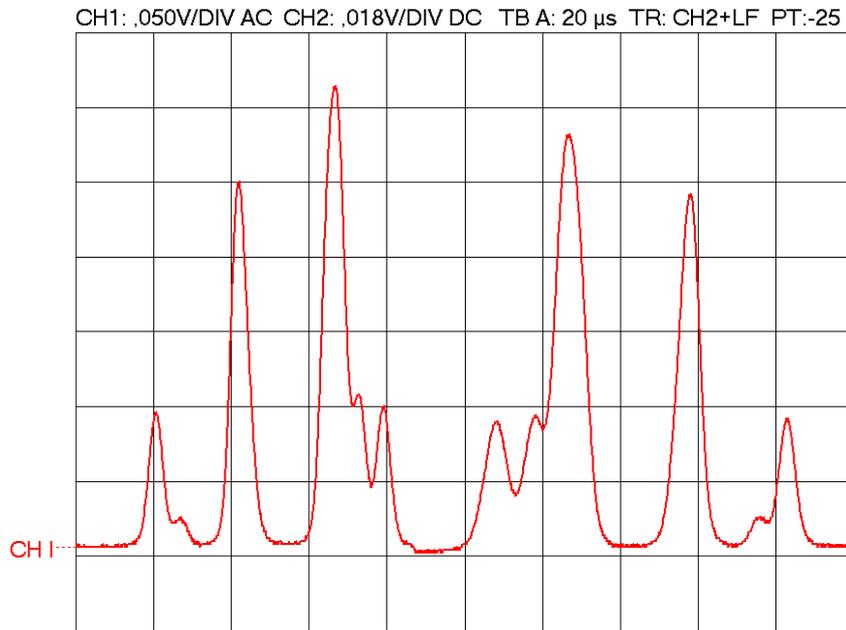


Abbildung 33: Das invertierte HFS-Spektrum mit abgezogener Dreieckspannung

A.2. Ausgewählte Spinpräzessions-Aufnahmen

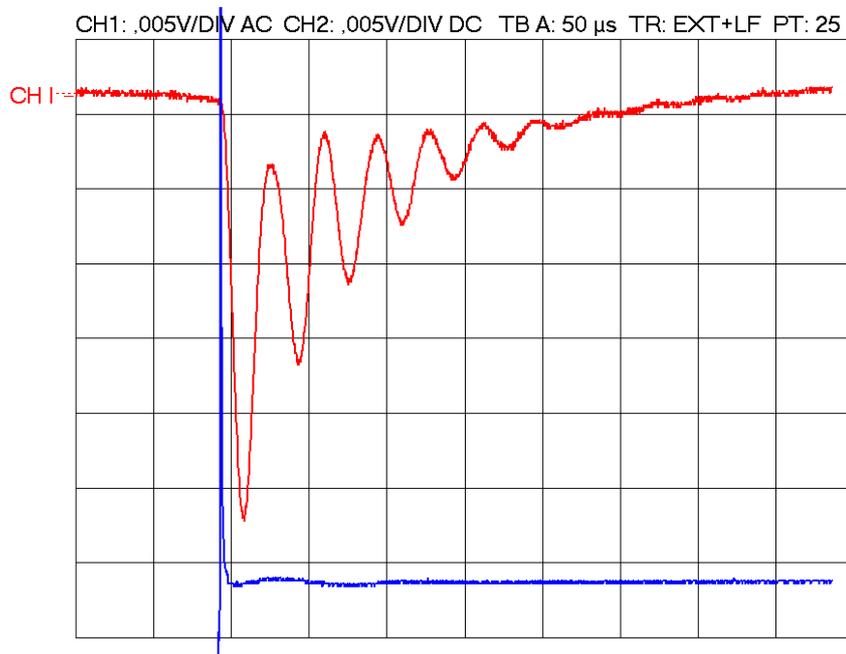


Abbildung 34: Spinpräzession bei ^{85}Rb , Messung 7

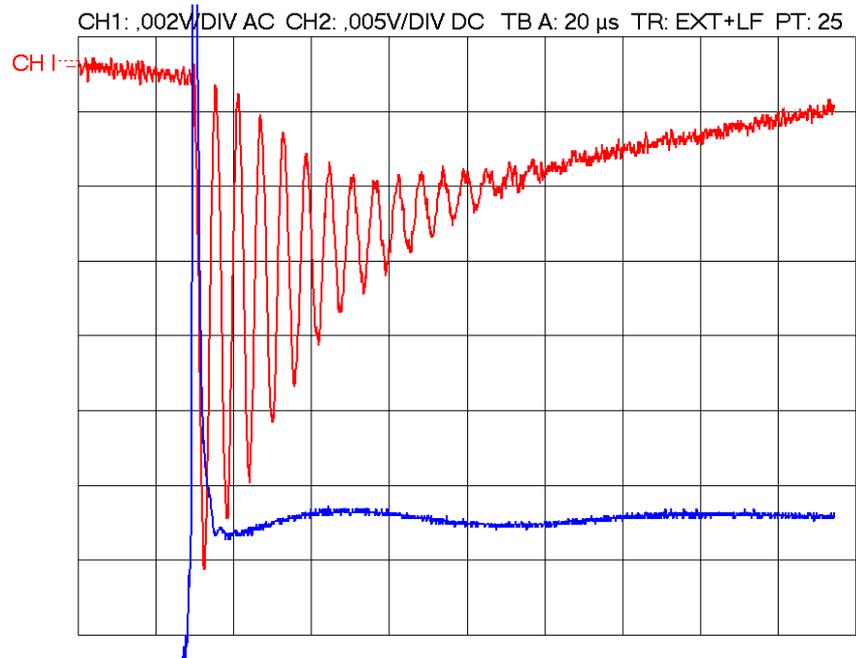


Abbildung 35: Spinpräzession bei ^{85}Rb , Messung 11

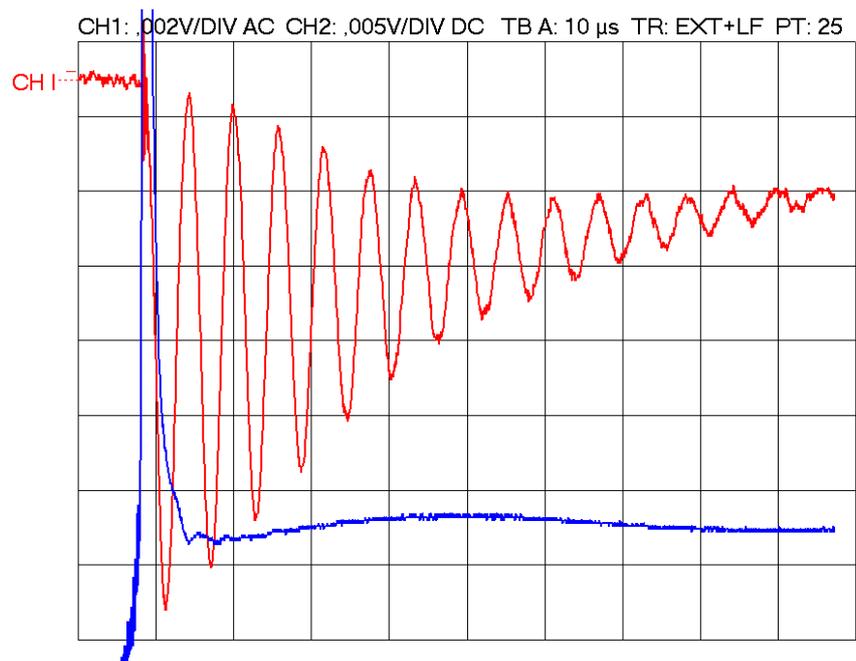


Abbildung 36: Spinpräzession bei ^{85}Rb , Messung 13

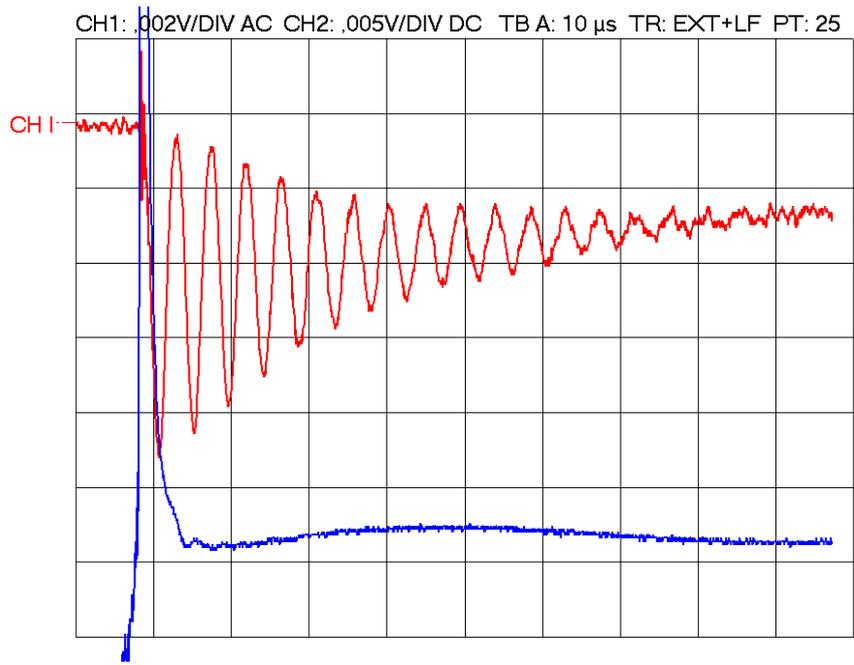


Abbildung 37: Spinpräzession bei ^{87}Rb , Messung 3

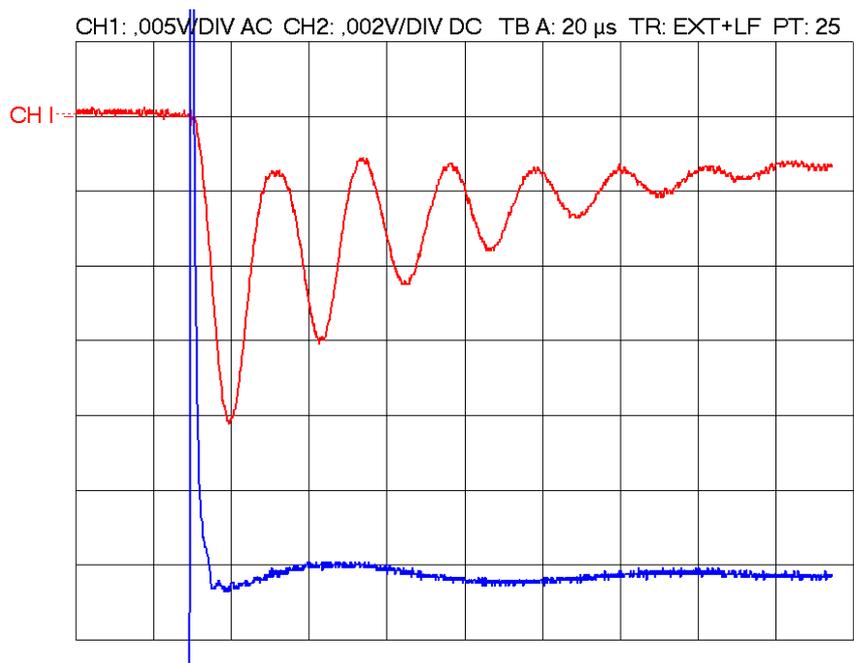


Abbildung 38: Spinpräzession bei ^{87}Rb , Messung 8

A.3. Fits zur Relaxationszeit nach Dehmelt

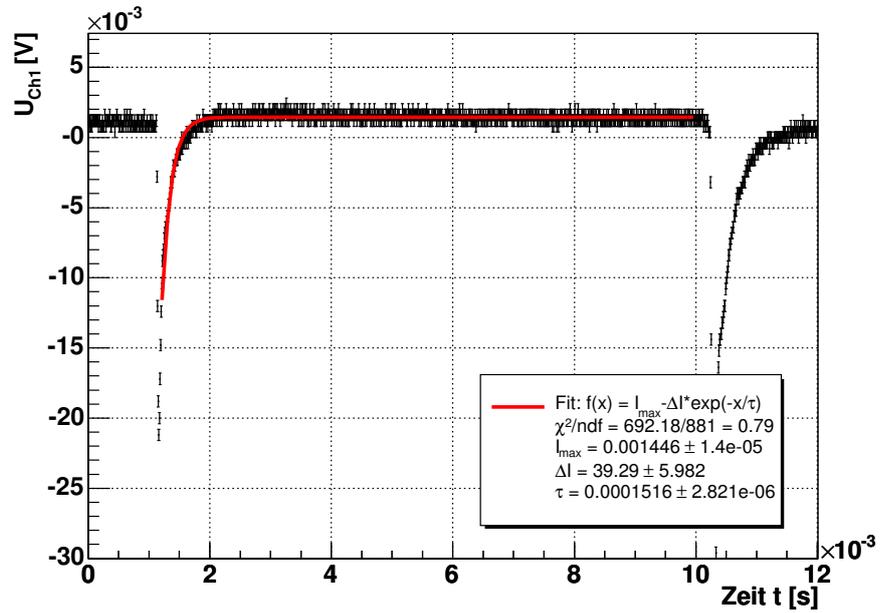


Abbildung 39: Fit zur Messung ohne Neutralfilter

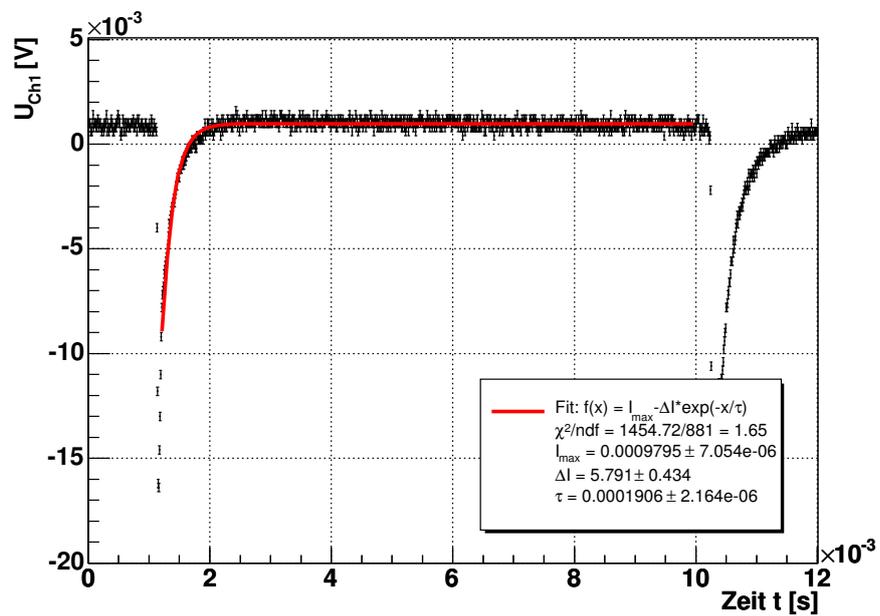


Abbildung 40: Fit zur Messung mit Neutralfilter 1

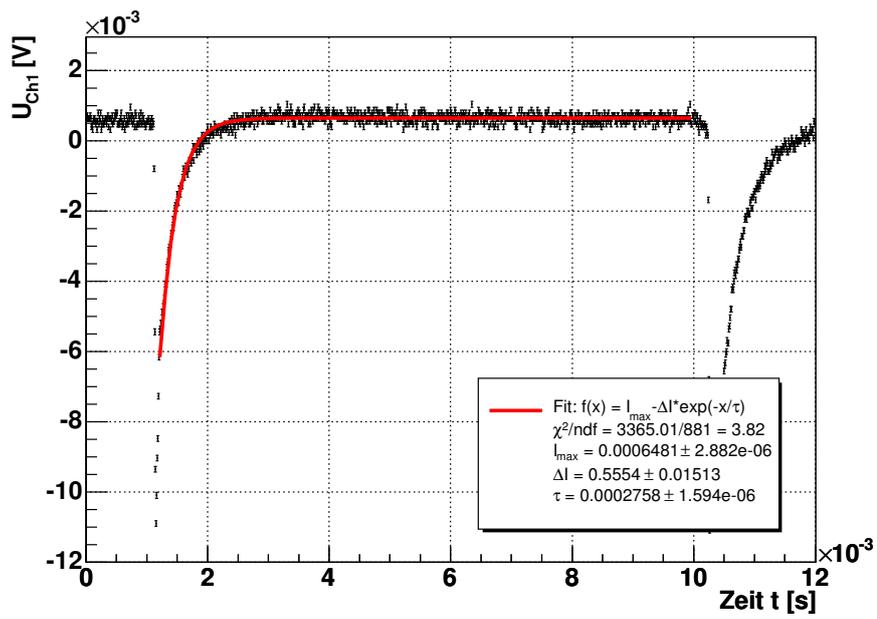


Abbildung 41: Fit zur Messung mit Neutralfilter 2

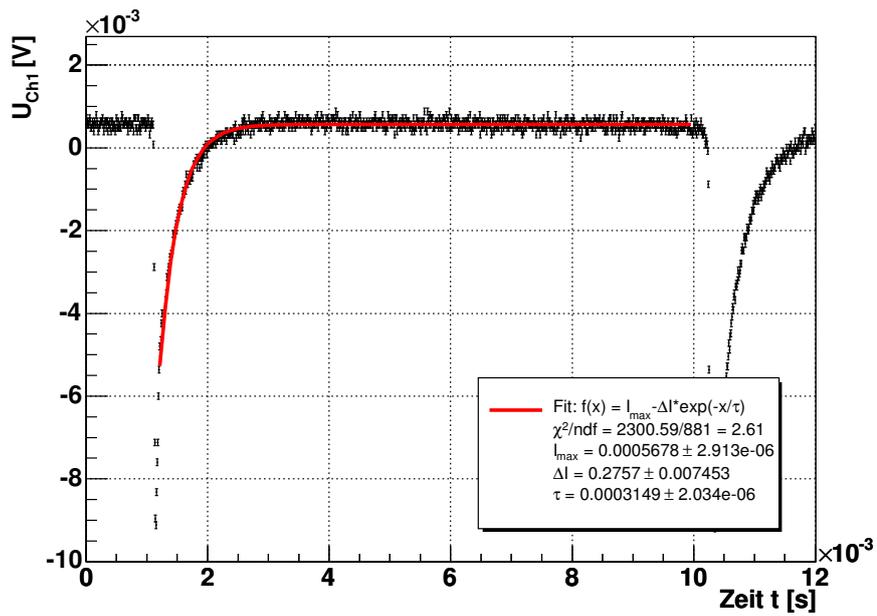


Abbildung 42: Fit zur Messung mit Neutralfilter 3

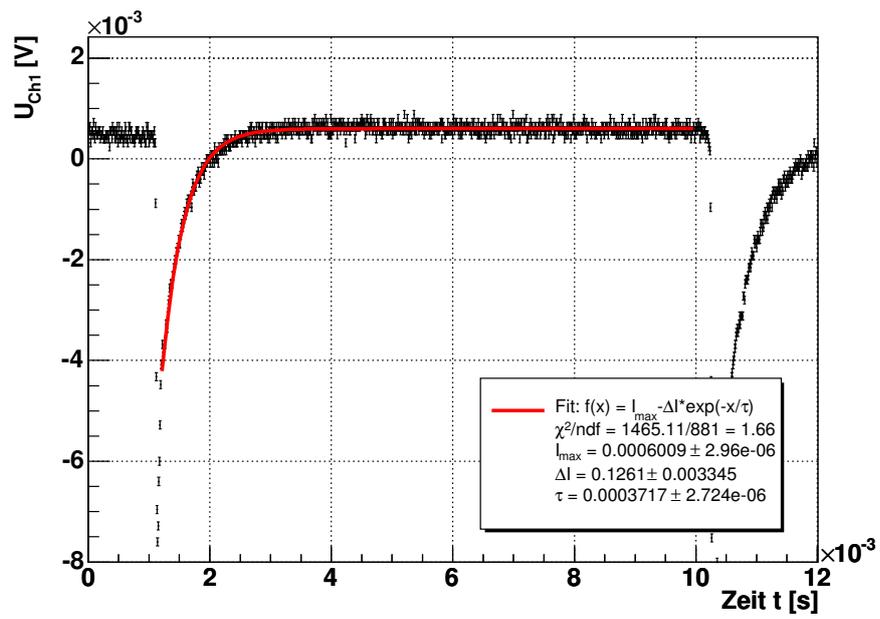


Abbildung 43: Fit zur Messung mit Neutralfilter 4

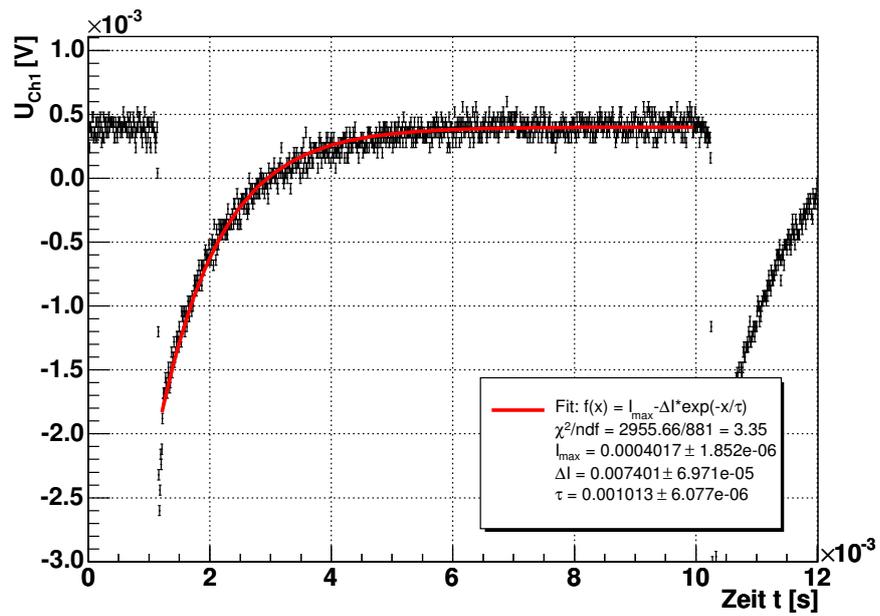


Abbildung 44: Fit zur Messung mit Neutralfilter 5

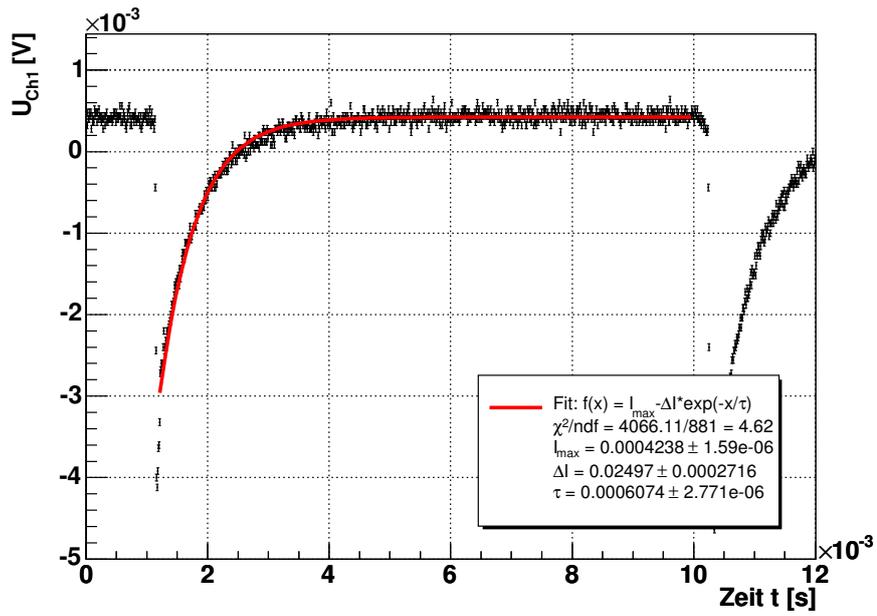


Abbildung 45: Fit zur Messung mit Neutralfilter 6

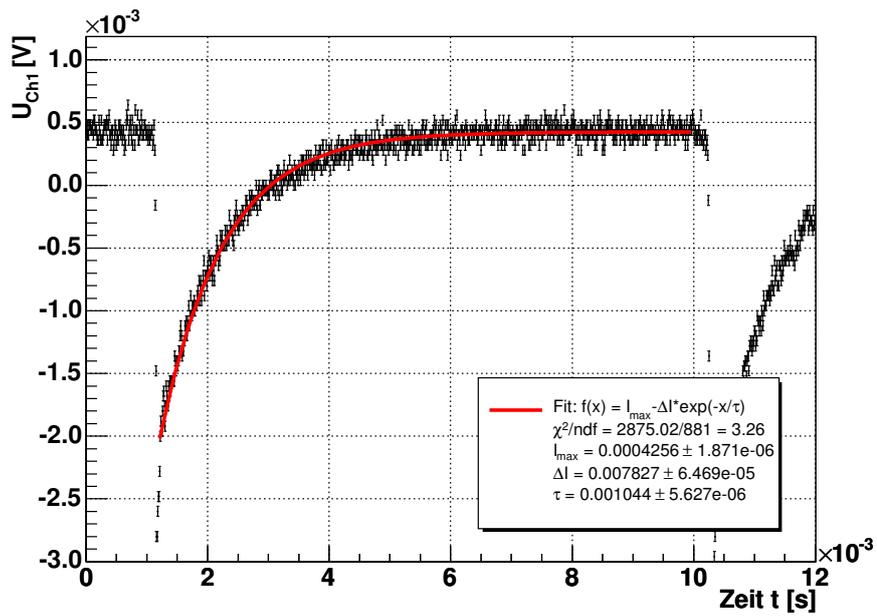


Abbildung 46: Fit zur Messung mit Neutralfilter 7

B. Quelltexte¹

Wir verwendeten zur Auswertung die Programmiersprache Python mit dem Datenanalyse Framework ROOT, sowie die Oszilloskop-Software des Oszilloskops HM1505 der Firma Hameg.

B.1. Laserdiodenkennlinie (laser_kennlinie.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from array import array
5  from fit_tools import create_fit_legend, print_fit_result
6  from ROOT import gROOT, TCanvas, TGraphErrors, TF1, TLegend
7
8  gROOT.SetStyle("Plain")
9
10 # Lese Daten aus Datei
11 data = []
12 for line in file('messdaten/laser_kennlinie.dat', 'r'):
13     # Laserdiodenstrom [mA], Photodiodenspannung, Einheit [mV]
14     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
15         vals = map(float, line.split())
16         assert len(vals) == 3
17         data += [vals]
18
19 # Extrahiere Laserdiodenstrom [mA]
20 I = [z[0] for z in data]
21 sI = [0.1]*len(data)
22
23 # Extrahiere Photodiodenspannung [mV]
24 U = [z[1]*z[2] for z in data]
25 sU = [z[2]*0.02 for z in data]
26
27 # Zeichne Messdaten
28 c = TCanvas('c_laserkennlinie', 'Laser-Kennlinie')
29 g = TGraphErrors(len(data), array('d', I), array('d', U),
30                 array('d', sI), array('d', sU))
31 g.SetTitle(';Laserdiodenstrom I_{L} [mA];Photodiodenspannung U [mV]')
32 g.GetYaxis().SetTitleOffset(1.2)
33 g.SetMarkerStyle(3)
34 g.SetMarkerColor(2)
35 g.Draw('AP')
36
37 # Linearer Fit
38 f = TF1('f', '[0]*x + [1]', 52, 78)
39 for i, pn in enumerate(['a', 'b']): f.SetParName(i, pn)
```

¹Siehe <http://www.physik.uni-freiburg.de/~kolja/fp2/optpump/>

```

40
41 g.Fit(f, 'RQ')
42 print_fit_result(f)
43
44 lg = create_fit_legend(f, fit_name = 'Linearer Fit', fstr_nl = False,
45                       lpos = (0.14, 0.65, 0.45, 0.86))
46 lg.Draw()

```

B.2. Photodiodenkennlinie (photo_kennlinie.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from math import exp
5  from array import array
6  from fit_tools import create_fit_legend, print_fit_result
7  from ROOT import gROOT, TCanvas, TGraphErrors, TF1, TLegend
8
9  gROOT.SetStyle("Plain")
10
11 # Lese Daten aus Datei
12 data = []
13 for line in file('messdaten/photo_kennlinie.dat', 'r'):
14     # Filter, Filter (neu), Spannung [V], Fehler [V]
15     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
16         vals = map(float, line.split())
17         assert len(vals) == 4
18         data += [vals]
19
20 # relative Intensitaeten aus angegebene Neutralfilterwerte
21 Ia = [exp(-z[0]) for z in data]
22 Ib = [exp(-z[1]) for z in data]
23 sI = [0]*len(data)
24
25 # Photodiodenspannungen [V]
26 U = [z[2] for z in data]
27 sU = [z[3] for z in data]
28
29 # trage "alte" Filterwerte auf
30 ca = TCanvas('ca', 'Photodiodenkennlinie, alt')
31 ca.SetGrid()
32 ga = TGraphErrors(len(U), array('d', U), array('d', Ia),
33                  array('d', sU), array('d', sI))
34 ga.SetTitle('"Alte" Werte;Photodiodenspannung [V];relative Intensit#ddot{a}t')
35 ga.SetMarkerStyle(3); ga.SetMarkerColor(2)
36 ga.GetAxis().SetDecimals()
37 ga.SetMinimum(0); ga.SetMaximum(1.6)
38 ga.Draw('AP')
39
40 fa = TF1('f', '[0]*x + [1]')

```

```

41 for i,pn in enumerate(['a','b']): fa.SetParName(i,pn); fa.SetParameter(i,1)
42 ga.Fit(fa, 'Q'); print_fit_result(fa, show_fstr = False)
43 la = create_fit_legend(fa, fit_name = 'Linearer Fit', fstr_nl = False,
44                       lpos = (0.14, 0.65, 0.50, 0.86))
45 la.Draw()
46
47 # trage "neue" Filterwerte auf
48 cb = TCanvas('cb', 'Photodiodenkennlinie, neu')
49 cb.SetGrid()
50 gb = TGraphErrors(len(U), array('d', U), array('d', Ib),
51                  array('d', sU), array('d', sI))
52 gb.SetTitle('"Neue" Werte;Photodiodenspannung [V];relative Intensit#ddot{a}t')
53 gb.SetMarkerStyle(3); gb.SetMarkerColor(2)
54 gb.SetMinimum(0); gb.SetMaximum(1.6)
55 gb.GetXaxis().SetDecimals()
56 gb.Draw('AP')
57
58 fb = TF1('f', '[0]*x + [1]')
59 for i,pn in enumerate(['a','b']): fb.SetParName(i,pn); fb.SetParameter(i,1)
60 gb.Fit(fb, 'Q'); print; print_fit_result(fb, show_fstr = False)
61 lb = create_fit_legend(fb, fit_name = 'Linearer Fit', fstr_nl = False,
62                       lpos = (0.14, 0.65, 0.50, 0.86))
63 lb.Draw()

```

B.3. Neutralfilter (neutral_filter.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from math import exp, log
5  from konst import Q
6
7  # Sollen LaTeX-Tabellen erzeugt werden?
8  gen_tabellen = False
9
10 # Sollen Ergebnisse gespeichert werden?
11 speichere_ergebnisse = True
12
13 # Lese Daten aus Datei
14 data = []
15 for line in file('messdaten/neutral_filter.dat', 'r'):
16     # Bezeichnung, Filter (alt), Filter (neu), Spannung [V], Fehler [V]
17     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
18         toks = line.split()[1:]
19         assert len(toks) == 5
20         vals = toks[:3] + map(float, toks[3:])
21         data += [vals]
22
23 # Bezeichnungen
24 bez = [z[0] for z in data]

```

```

25
26 # extrahiere Intensitaeten
27 I, sI = [z[3] for z in data], [z[4] for z in data]
28
29 # relative Intensitaeten
30 Ir = [Ii/I[0] for Ii in I]
31 sIr = [Iri*((sI[i]/I[i])**2+(sI[0]/I[0])**2)**0.5 for i,Iri in enumerate(Ir)]
32 sIr[0] = 0.0
33
34 # berechne Abschwaechung
35 x = [log(Iri) for Iri in Ir]
36 sx = [abs(sIri/Iri) for Iri,sIri in zip(Ir,sIr)]
37
38 # gebe Ergebnis aus
39 for i,d in enumerate(data):
40     print '%6s: I = %.3f +- %.3f, Ir = %.4f +- %.4f, x = %6.3f +- %.3f' % (
41         d[0], I[i], sI[i], Ir[i], sIr[i], x[i], sx[i])
42
43 # erzeuge LaTeX-Tabelle
44 if gen_tabellen:
45     import texgen
46     ftab = file('tables/neutral_filter.out.tex', 'w')
47     s = texgen.table_neutral_filter(data)
48     ftab.write(s); ftab.close()
49     ftab = file('tables/neutral_filter_eichung.out.tex', 'w')
50     s = texgen.table_neutral_filter_eichung(data, I, sI, Ir, sIr, x, sx)
51     ftab.write(s); ftab.close()
52
53 # Ergebnisse speichern
54 if speichere_ergebnisse:
55     import shelve
56     d = shelve.open('results.out')
57     d['neutral_filter.bez'] = bez
58     d['neutral_filter.Ir'] = Ir
59     d['neutral_filter.sIr'] = sIr
60     d['neutral_filter.x'] = x
61     d['neutral_filter.sx'] = sx
62     d.close()

```

B.4. Frequenzzeichnung der Hyperfeinstruktur (freq_eich.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from konst import Q, FSR, sFSR, mus, sec
5
6  # Maximum 1 und 2, links
7  p1l, sp1l = 429.0*mus, 1.0*mus
8  p2l, sp2l = 353.0*mus, 1.0*mus
9  p3l, sp3l = 272.0*mus, 1.0*mus

```

```

10 p4l, sp4l = 181.0*mus, 1.0*mus
11
12 # Differenz zwischen den jeweiligen Maxima, links
13 dl, sdl = p1l-p2l, (sp1l**2 + sp2l**2)**0.5
14 dl2, sdl2 = p2l-p3l, (sp2l**2 + sp3l**2)**0.5
15 dl3, sdl3 = p3l-p4l, (sp3l**2 + sp4l**2)**0.5
16
17 # Maximum 1 und 2, rechts
18 p1r, sp1r = 565.0*mus, 1.0*mus
19 p2r, sp2r = 649.0*mus, 1.0*mus
20 p3r, sp3r = 729.0*mus, 1.0*mus
21 p4r, sp4r = 802.0*mus, 1.0*mus
22
23 # Differenz zwischen den jeweiligen Maxima, rechts
24 dr, sdr = p2r-p1r, (sp2r**2 + sp1r**2)**0.5
25 dr2, sdr2 = p3r-p2r, (sp3r**2 + sp2r**2)**0.5
26 dr3, sdr3 = p4r-p3r, (sp4r**2 + sp3r**2)**0.5
27
28 # Eich-Koeffizienten (linke und rechte Seite)
29 fel = FSR/dl; sfel = fel * ( (sFSR/FSR)**2 + (sdl/dl)**2 )**0.5
30 fer = FSR/dr; sfer = fer * ( (sFSR/FSR)**2 + (sdr/dr)**2 )**0.5
31
32 # Eichung, linke Seite
33 def feich_l(dt, sdt):
34     f = fel*dt; sf = abs(f) * ( (sfel/fel)**2 + (sdt/dt)**2 )**0.5
35     return f.inUnitsOf('GHz'), sf.inUnitsOf('GHz')
36
37 # Eichung, rechte Seite
38 def feich_r(dt, sdt):
39     f = fer*dt; sf = abs(f) * ( (sfer/fer)**2 + (sdt/dt)**2 )**0.5
40     return f.inUnitsOf('GHz'), sf.inUnitsOf('GHz')
41
42 # Erstellung der Grafik
43 if __name__ == '__main__':
44     print 'Eichung:'
45     print 'links: %.2f +- %.2f MHz/mus' % (
46         fel/Q('1MHz/mus'), sfel/Q('1MHz/mus'))
47     print 'rechts: %.2f +- %.2f MHz/mus' % (
48         fer/Q('1MHz/mus'), sfer/Q('1MHz/mus'))
49
50     from oszi_messung import Messung
51     from ROOT import gROOT, TCanvas, TGraph, TGaxis, TLine, TArrow, TLatex
52
53     gROOT.SetStyle("Plain")
54     TGaxis.SetMaxDigits(3)
55
56     # Lade Messdaten
57     m = Messung('hfs/nochmal/etl64_2.tab')
58     m.draw(show_grid = False, draw_lines = True, x_range = (1.4e-4, 8.4e-4))
59
60     ymin = m.g.GetHistogram().GetMinimum()
61     ymax = m.g.GetHistogram().GetMaximum()

```

```

62
63 # Zeichne Linien
64 l1l = TLine(p1l/sec, ymin, p1l/sec, ymax*0.94)
65 l1r = TLine(p1r/sec, ymin, p1r/sec, ymax*0.94)
66 l2l = TLine(p2l/sec, ymin, p2l/sec, ymax*0.94)
67 l2r = TLine(p2r/sec, ymin, p2r/sec, ymax*0.94)
68 l3l = TLine(p3l/sec, ymin, p3l/sec, ymax*0.54)
69 l3r = TLine(p3r/sec, ymin, p3r/sec, ymax*0.54)
70 l4l = TLine(p4l/sec, ymin, p4l/sec, ymax*0.14)
71 l4r = TLine(p4r/sec, ymin, p4r/sec, ymax*0.14)
72 for l in [l1l, l2l, l3l, l4l, l1r, l2r, l3r, l4r]:
73     l.SetLineWidth(2); l.SetLineStyle(2); l.SetLineColor(2); l.Draw()
74
75 # Zeichne Abstaende
76 a1 = TArrow(p1l/sec, ymax*0.9, p2l/sec, ymax*0.9)
77 ar = TArrow(p1r/sec, ymax*0.9, p2r/sec, ymax*0.9)
78 a12 = TArrow(p2l/sec, ymax*0.5, p3l/sec, ymax*0.5)
79 ar2 = TArrow(p2r/sec, ymax*0.5, p3r/sec, ymax*0.5)
80 a13 = TArrow(p3l/sec, ymax*0.1, p4l/sec, ymax*0.1)
81 ar3 = TArrow(p3r/sec, ymax*0.1, p4r/sec, ymax*0.1)
82 for a in [a1, ar, a12, ar2, a13, ar3]:
83     a.SetFillColor(1); a.SetArrowSize(0.02); a.Draw('<|>')
84
85 t1 = TLatex((p1l+p2l)/2/sec, ymax*0.86, '%.0f #mus' % (dl/mus))
86 tr = TLatex((p1r+p2r)/2/sec, ymax*0.86, '%.0f #mus' % (dr/mus))
87 t12 = TLatex((p2l+p3l)/2/sec, ymax*0.46, '%.0f #mus' % (dl2/mus))
88 tr2 = TLatex((p2r+p3r)/2/sec, ymax*0.46, '%.0f #mus' % (dr2/mus))
89 t13 = TLatex((p3l+p4l)/2/sec, ymax*0.06, '%.0f #mus' % (dl3/mus))
90 tr3 = TLatex((p3r+p4r)/2/sec, ymax*0.06, '%.0f #mus' % (dr3/mus))
91 for t in [t1, tr, t12, tr2, t13, tr3]:
92     t.SetTextAlign(23); t.SetTextSize(0.04); t.Draw()

```

B.5. Hyperfeinstrukturspektrum (hfs_spektrum.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from konst import Q, mus, sec, GHz, h
5  from oszi_messung import Messung
6  from freq_eich import feich_l, feich_r
7  from texgen import table_hfs_spek
8  from ROOT import gROOT, TCanvas, TGraph, TGraph, TAxis, TLine, TLatex
9
10 gROOT.SetStyle("Plain")
11 TAxis.SetMaxDigits(3)
12
13 # Sollen LaTeX-Tabellen erzeugt werden?
14 gen_tabellen = False
15
16 # Linke Seite -----

```

```

17
18 # lese Daten aus Datei (linke Seite)
19 data_l = []
20 for line in file('messdaten/hfs_peaks_l.dat', 'r'):
21     # Peak, rel. theor. Frequenz [GHz], Zeit [mus], Fehler [mus]
22     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
23         toks = line.split()
24         assert len(toks) == 4
25         vals = toks[1:] + map(float, toks[1:])
26         data_l += [vals]
27
28 # Peak-Bezeichnungen
29 peaks_tex_l = [z[0] for z in data_l]
30 peaks_l = [z.replace('$^{', '').replace('}$', '').replace('~F:~', ' ')
31            for z in peaks_tex_l]
32
33 # theoretische Frequenz [GHz] (relativ zur D1-Linie)
34 nu_th_l = [Q(z[1], 'GHz') for z in data_l]
35
36 # gemessene Zeit [mus] (relativ zum Trigger)
37 t_l = [z[2]*mus for z in data_l]
38 st_l = [z[3]*mus for z in data_l]
39
40 # Zeit relativ zum ersten Peak [mus]
41 rt_l = [ti-t_l[0] for ti in t_l]
42 srt_l = [(sti**2 + st_l[0]**2)**0.5 for sti in st_l]
43 srt_l[0] = 0*mus
44
45 # Frequenzen relativ zum ersten Peak [GHz]
46 nu_eich_l = [(0*GHz,0*GHz)] + [
47     feich_l(rti, srti) for rti,srti in zip(rt_l, srt_l)[1:]]
48
49 # Frequenzen [GHz] (relativ zur D1-Linie, verwende nu_th[0])
50 rnu_l = [nu_th_l[0]-z[0] for z in nu_eich_l]
51 srnu_l = [z[1] for z in nu_eich_l]
52
53 # Differenz zum Literaturwert
54 diff_l = [rnui - nuthi for rnui, nuthi in zip(rnu_l, nu_th_l)]
55
56 print 'relative Frequenzen, linke Seite:'
57 for i, rnui in enumerate(rnu_l):
58     print '%d: %s, %6.3f +- %.3f GHz, th: %5.2f GHz, diff: %6.3f' % (
59         i, peaks_l[i], rnui/GHz, srnu_l[i]/GHz, nu_th_l[i]/GHz, diff_l[i]/GHz )
60
61 # erzeuge LaTeX-Tabelle
62 if gen_tabellen:
63     ftab = file('tables/hfs_spek_l.out.tex', 'w')
64     s = table_hfs_spek(peaks_tex_l, t_l, st_l, rnu_l, srnu_l, nu_th_l, diff_l)
65     ftab.write(s); ftab.close()
66
67 # Rechte Seite -----
68

```

```

69 # lese Daten aus Datei (rechte Seite)
70 data_r = []
71 for line in file('messdaten/hfs_peaks_r.dat', 'r'):
72     # Peak, rel. theor. Frequenz [GHz], Zeit [mus], Fehler [mus]
73     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
74         toks = line.split()
75         assert len(toks) == 4
76         vals = toks[1:] + map(float, toks[1:])
77         data_r += [vals]
78
79 # Peak-Bezeichnungen
80 peaks_tex_r = [z[0] for z in data_r]
81 peaks_r = [z.replace('$^{', '').replace('}$', '').replace(',~F:~', ' ')
82            for z in peaks_tex_r]
83
84 # theoretische Frequenz [GHz] (relativ zur D1-Linie)
85 nu_th_r = [Q(z[1], 'GHz') for z in data_r]
86
87 # gemessene Zeit [mus] (relativ zum Trigger)
88 t_r = [z[2]*mus for z in data_r]
89 st_r = [z[3]*mus for z in data_r]
90
91 # Zeit relativ zum ersten Peak [mus]
92 rt_r = [ti-t_r[0] for ti in t_r]
93 srt_r = [(sti**2 + st_r[0]**2)**0.5 for sti in st_r]
94 srt_r[0] = 0*mus
95
96 # Frequenzen relativ zum ersten Peak [GHz]
97 nu_eich_r = [(0*GHz,0*GHz)] + [
98     feich_r(rti, srti) for rti,srti in zip(rt_r, srt_r)[1:]]
99
100 # Frequenzen [GHz] (relativ zur D1-Linie, verwende nu_th[0])
101 rnu_r = [nu_th_r[0]+z[0] for z in nu_eich_r]
102 srnu_r = [z[1] for z in nu_eich_r]
103
104 # Differenz zum Literaturwert
105 diff_r = [rnui - nuthi for rnui, nuthi in zip(rnu_r, nu_th_r)]
106
107 print '\nrelative Frequenzen, rechte Seite:'
108 for i, rnui in enumerate(rnu_r):
109     print '%d: %s, %6.3f +- %.3f GHz, th: %5.2f GHz, diff: %6.3f' % (
110         i, peaks_r[i], rnui/GHz, srnu_r[i]/GHz, nu_th_r[i]/GHz, diff_r[i]/GHz )
111
112 # erzeuge LaTeX-Tabelle
113 if gen_tabellen:
114     ftab = file('tables/hfs_spek_r.out.tex', 'w')
115     s = table_hfs_spek(peaks_tex_r, t_r, st_r, rnu_r, srnu_r, nu_th_r, diff_r)
116     ftab.write(s); ftab.close()
117
118 # Grafik -----
119
120 m = Messung('hfs/nochmal/hfs2.tab')

```

```

121 m.draw(show_grid = False, draw_lines = True,
122         y_title_offs = 1.2, y_title = 'Absorption',
123         y_min = -0.45, y_max = -0.05)#, x_range=(0.09e-3,0.2e-3))
124
125 ymin = m.g.GetHistogram().GetMinimum()
126 ymax = m.g.GetHistogram().GetMaximum()
127 hy = ymax - ymin
128
129 # Hoehe von Linien und Texten, linke Seite
130 al = [0.46, 0.34, 0.71, 0.71, 0.88, 0.88, 0.50, 0.34]
131 tal = [0.47, 0.35, 0.76, 0.72, 0.93, 0.89, 0.51, 0.35]
132
133 # zeichne Linien, linke Seite
134 lines_l = [TLine(ti/sec, ymin, ti/sec, ymin + hy*al[i])
135            for i,ti in enumerate(t_l)]
136 for l in lines_l:
137     l.SetLineWidth(2); l.SetLineStyle(2); l.SetLineColor(2); l.Draw()
138
139 # zeichne Bezeichnungen, linke Seite
140 texts_l = [
141     TLatex(ti/sec, ymin + hy*tal[i],
142           peaks_tex_l[i].replace('$','').replace('~F:~',' '))
143     for i,ti in enumerate(t_l) ]
144 for tx in texts_l:
145     tx.SetTextAlign(21); tx.SetTextSize(0.03); tx.Draw()
146
147 # Hoehe von Linien und Texten, rechte Seite
148 ar = [0.50, 0.34, 0.80, 0.80, 0.70, 0.70, 0.44, 0.34]
149 tar = [0.51, 0.35, 0.85, 0.81, 0.75, 0.71, 0.45, 0.35]
150
151 # zeichne Linien, rechte Seite
152 lines_r = [TLine(ti/sec, ymin, ti/sec, ymin + hy*ar[i])
153            for i,ti in enumerate(t_r)]
154 for l in lines_r:
155     l.SetLineWidth(2); l.SetLineStyle(2); l.SetLineColor(2); l.Draw()
156
157 # zeichne Bezeichnungen, rechte Seite
158 texts_r = [
159     TLatex(ti/sec, ymin + hy*tar[i],
160           peaks_tex_r[i].replace('$','').replace('~F:~',' '))
161     for i,ti in enumerate(t_r) ]
162 for tx in texts_r:
163     tx.SetTextAlign(21); tx.SetTextSize(0.03); tx.Draw()

```

B.6. Berechnung der Intervallkonstanten (intervall_konst.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from tools import gew_mittel

```

```

5 from konst import Q, mus, sec, GHz, h
6 from oszi_messung import Messung
7 from freq_eich import feich_l, feich_r
8
9 # lese Daten aus Datei (linke Seite)
10 bez_tex, t, st = [], [], []
11 for line in file('messdaten/hfs_peaks_l.dat', 'r'):
12     # Peak, rel. theor. Frequenz [GHz], Zeit [mus], Fehler [mus]
13     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
14         toks = line.split(); assert len(toks) == 4
15         bez_tex += [toks[0]] # Peak Bezeichnung (TeX)
16         t += [float(toks[2]) * mus] # gemessene Zeit [mus]
17         st += [float(toks[3]) * mus] # Fehler der Zeitmessung [mus]
18
19 # Kurzversion der Peak-Bezeichnungen
20 bez = [z.replace('$^{', '').replace('$', '').replace('~F:~', ' ')]
21     for z in bez_tex]
22
23 # Intervallkonstante Rb87, S (F=1)
24 dt_22_12 = t[6] - t[0]
25 sdt_22_12 = (st[6]**2 + st[0]**2)**0.5
26 nu_22_12, snu_22_12 = feich_l(dt_22_12, sdt_22_12)
27 AS1 = (nu_22_12 * h / 2).inUnitsOf('eV')
28 sAS1 = (snu_22_12 * h / 2).inUnitsOf('eV')
29
30 dt_21_11 = t[7] - t[1]
31 sdt_21_11 = (st[7]**2 + st[1]**2)**0.5
32 nu_21_11, snu_21_11 = feich_l(dt_21_11, sdt_21_11)
33 AS2 = (nu_21_11 * h / 2).inUnitsOf('eV')
34 sAS2 = (snu_21_11 * h / 2).inUnitsOf('eV')
35
36 # gewichtetes Mittel
37 AS, sAS = gew_mittel([(AS1, sAS1), (AS2, sAS2)])
38
39 print 'Rb87, AS:'
40 print '%.3f +- %.3f E-6 eV' % (AS1/Q('1mueV'), sAS1/Q('1mueV'))
41 print '%.3f +- %.3f E-6 eV' % (AS2/Q('1mueV'), sAS1/Q('1mueV'))
42 print 'Mittel: %.3f +- %.3f E-6 eV' % (AS/Q('1mueV'), sAS/Q('1mueV'))
43
44 # Intervallkonstante Rb87, P (F=1)
45 dt_11_12 = t[1] - t[0]
46 sdt_11_12 = (st[1]**2 + st[0]**2)**0.5
47 nu_11_12, snu_11_12 = feich_l(dt_11_12, sdt_11_12)
48 AP1 = (nu_11_12 * h / 2).inUnitsOf('eV')
49 sAP1 = (snu_11_12 * h / 2).inUnitsOf('eV')
50
51 dt_21_22 = t[7] - t[6]
52 sdt_21_22 = (st[7]**2 + st[6]**2)**0.5
53 nu_21_22, snu_21_22 = feich_l(dt_21_22, sdt_21_22)
54 AP2 = (nu_21_22 * h / 2).inUnitsOf('eV')
55 sAP2 = (snu_21_22 * h / 2).inUnitsOf('eV')
56

```

```

57 # gewichtetes Mittel
58 AP, sAP = gew_mittel([(AP1, sAP1), (AP2, sAP2)])
59
60 print '\nRb87, AP:'
61 print '%.3f +- %.3f E-6 eV' % (AP1/Q('1mueV'), sAP1/Q('1mueV'))
62 print '%.3f +- %.3f E-6 eV' % (AP2/Q('1mueV'), sAP2/Q('1mueV'))
63 print 'Mittel: %.3f +- %.3f E-6 eV' % (AP/Q('1mueV'), sAP/Q('1mueV'))
64
65 # Intervallkonstante Rb85, S (F=2)
66 dt_33_23 = t[4] - t[2]
67 sdt_33_23 = (st[4]**2 + st[2]**2)**0.5
68 nu_33_23, snu_33_23 = feich_1(dt_33_23, sdt_33_23)
69 AS_85 = (nu_33_23 * h / 3).inUnitsOf('eV')
70 sAS_85 = (snu_33_23 * h / 3).inUnitsOf('eV')
71
72 print '\nRb85, AS:'
73 print '%.3f +- %.3f E-6 eV' % (AS_85/Q('1mueV'), sAS_85/Q('1mueV'))

```

B.7. Doppelresonanz (doppelresonanz.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from konst import Q, A, T, muT, kHz, h, muB
5
6  # Sollen LaTeX-Tabellen erzeugt werden?
7  gen_tabellen = False
8
9  # Spule 1: Verhaeltnis Feldstaerke / Spulenstrom
10 a, sa = 7.99e-4*T/A, 0.01e-4*T/A
11
12 # Berechnung des horizontalen Erdmagnetfelds
13 def erdmagnetfeld(Im1, sIm1, Im2, sIm2):
14     B = a * (Im1-Im2) / 2.0
15     sB = abs(B) * ( (sa/a)**2 + (sIm1**2 + sIm2**2)/(Im1-Im2)**2 )**0.5
16     return B, sB
17
18 # Berechnung des Kerspins
19 def kernspin(Im1, sIm1, Im2, sIm2, nu, snu):
20     B = a * (Im1+Im2) / 2.0
21     sB = abs(B) * ( (sa/a)**2 + (sIm1**2 + sIm2**2)/(Im1+Im2)**2 )**0.5
22     I = muB*B/(h*nu) - 0.5
23     sI = (I+0.5) * ((sB/B)**2 + (snu/nu)**2)**(0.5)
24     return I, sI
25
26 # Listen fuer Ergebnisse
27 m, B, I = [], [], []
28
29 # Erste Messung
30 print 'Erste Messung, Rb85:'

```

```

31  Im1, sIm1 = 0.155*A, 0.001*A
32  Im2, sIm2 = 0.1155*A, 0.001*A
33  nu, snu = 500.221*kHz, 0.02*kHz
34  m += [(Im1, sIm1), (Im2, sIm2), (nu, snu)]
35
36  B85, sB85 = erdmagnetfeld(Im1, sIm1, Im2, sIm2)
37  print 'Erdmagnetfeld: %.3f +- %.3f muT' % (B85/muT, sB85/muT)
38  B += [(B85, sB85)]
39
40  I85, sI85 = kernspin(Im1, sIm1, Im2, sIm2, nu, snu)
41  print 'Kernspin: %.5f +- %.5f' % (I85, sI85)
42  I += [(I85, sI85)]
43
44  print '\nErste Messung, Rb87:'
45  Im1, sIm1 = 0.110*A, 0.001*A
46  Im2, sIm2 = 0.071*A, 0.001*A
47  nu, snu = 500.196*kHz, 0.02*kHz
48  m += [(Im1, sIm1), (Im2, sIm2), (nu, snu)]
49
50  B87, sB87 = erdmagnetfeld(Im1, sIm1, Im2, sIm2)
51  print 'Erdmagnetfeld: %.3f +- %.3f muT' % (B87/muT, sB87/muT)
52  B += [(B87, sB87)]
53
54  I87, sI87 = kernspin(Im1, sIm1, Im2, sIm2, nu, snu)
55  print 'Kernspin: %.5f +- %.5f' % (I87, sI87)
56  I += [(I87, sI87)]
57
58  # Zweite Messung
59  print '\nZweite Messung, Rb85:'
60  Im1, sIm1 = 0.150*A, 0.001*A
61  Im2, sIm2 = 0.121*A, 0.001*A
62  nu, snu = 500.260*kHz, 0.02*kHz
63  m += [(Im1, sIm1), (Im2, sIm2), (nu, snu)]
64
65  B85, sB85 = erdmagnetfeld(Im1, sIm1, Im2, sIm2)
66  print 'Erdmagnetfeld: %.3f +- %.3f muT' % (B85/muT, sB85/muT)
67  B += [(B85, sB85)]
68
69  I85, sI85 = kernspin(Im1, sIm1, Im2, sIm2, nu, snu)
70  print 'Kernspin: %.5f +- %.5f' % (I85, sI85)
71  I += [(I85, sI85)]
72
73  print '\nZweite Messung, Rb87:'
74  Im1, sIm1 = 0.105*A, 0.001*A
75  Im2, sIm2 = 0.076*A, 0.001*A
76  nu, snu = 500.249*kHz, 0.02*kHz
77  m += [(Im1, sIm1), (Im2, sIm2), (nu, snu)]
78
79  B87, sB87 = erdmagnetfeld(Im1, sIm1, Im2, sIm2)
80  print 'Erdmagnetfeld: %.3f +- %.3f muT' % (B87/muT, sB87/muT)
81  B += [(B87, sB87)]
82

```

```

83 I87, sI87 = kernspin(Im1, sIm1, Im2, sIm2, nu, snu)
84 print 'Kernspin: %.5f +- %.5f' % (I87, sI87)
85 I += [(I87, sI87)]
86
87 # erzeuge LaTeX-Tabellen
88 if gen_tabellen:
89     import texgen
90
91     ftab = file('tables/dres_messwerte.out.tex', 'w')
92     s = texgen.table_dres_messwerte(m)
93     ftab.write(s); ftab.close()
94
95     ftab = file('tables/dres_magnetfeld.out.tex', 'w')
96     s = texgen.table_dres_magnetfeld(B)
97     ftab.write(s); ftab.close()
98
99     ftab = file('tables/dres_kernspin.out.tex', 'w')
100    s = texgen.table_dres_kernspin(I)
101    ftab.write(s); ftab.close()

```

B.8. Spinpräzession (spin_praez.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  import texgen
5  from array import array
6  from konst import Q, A, mA, mus, kHz, T, muT
7  from fit_tools import print_fit_result, create_fit_legend
8  from ROOT import gROOT, TCanvas, TGraphErrors, TF1
9
10 gROOT.SetStyle("Plain")
11
12 # Sollen LaTeX-Tabellen erzeugt werden?
13 gen_tabellen = False
14
15 # Spule 4: Verhaeltnis Feldstaerke / Spulenstrom
16 a, sa = 4.76e-4*T/A, 0.01e-4*T/A
17
18 class Messung:
19     def __init__(self, filename):
20         self.filename = filename
21
22         data = []
23         # Messung, Spulenstrom I [mA], sI [mA], Zeit t [mus], st [mus], Anzahl
24         for line in file('messdaten/'+filename, 'r'):
25             if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
26                 data += [map(float, line.split())]
27
28         # Nr. der Messung

```

```

29     self.nr = [z[0] for z in data]
30
31     # Spulenstrom [mA]
32     self.I = [z[1]*mA for z in data]
33     self.sI = [z[2]*mA for z in data]
34
35     # Zeit [mus]
36     self.t = [z[3]*mus for z in data]
37     self.st = [z[4]*mus for z in data]
38
39     # Anzahl der Perioden
40     self.n = [z[5] for z in data]
41
42     # Frequenz [kHz]
43     self.nu = [(ni/ti).inUnitsOf('kHz') for ni,ti in zip(self.n,self.t)]
44     self.snu = [(abs(nui*sti/ti)).inUnitsOf('kHz')
45                 for nui,ti,sti in zip(self.nu,self.t,self.st)]
46
47     # erzeuge Graphen
48     x = array('d', [nui/kHz for nui in self.nu])
49     sx = array('d', [snui/kHz for snui in self.snu])
50     y = array('d', [Ii/mA for Ii in self.I])
51     sy = array('d', [sIi/mA for sIi in self.sI])
52     g = self.graph = TGraphErrors(len(x), x, y, sx, sy)
53     g.SetTitle(';Frequenz #nu [kHz];Spulenstrom I [mA]')
54     g.SetMarkerStyle(3); g.SetMarkerColor(2)
55
56     def draw(self, show_grid = True):
57         c = self.canvas = TCanvas('c_'+self.filename, self.filename)
58         if show_grid: c.SetGrid()
59         self.graph.Draw('AP')
60
61     def fit(self):
62         # fuehre Fit durch
63         f = self.fcn = TF1('f'+self.filename, '[0]*x + [1]')
64         for i, (pn,pv) in enumerate([('a',-1), ('b',100)]):
65             f.SetParName(i, pn)
66             f.SetParameter(i, pv)
67         self.graph.Fit(f, 'Q')
68         print_fit_result(f)
69         self.lg = create_fit_legend(f, fit_name='Linearer Fit', fstr_nl=False)
70         self.lg.Draw()
71
72         # Y-Achsenabschnitt
73         self.I0, self.sI0 = f.GetParameter(1)*mA, f.GetParError(1)*mA
74
75         # Berechnung von B_vert
76         self.Bv = a * self.I0
77         self.sBv = self.Bv * ( (sa/a)**2 + (self.sI0/self.I0)**2 )**0.5
78
79     def write_table(self):
80         tabf = file('tables/'+self.filename[:-3]+'out.tex', 'w')

```

```

81         s = texgen.table_spin_praez(self)
82         #print s
83         tabf.write(s); tabf.close()
84
85
86     print 'Rb85:'
87     m85 = Messung('spin_praez_85.dat')
88     m85.draw(); m85.fit()
89     print 'Bv = %.3f +- %.3f muT' % (m85.Bv/muT, m85.sBv/muT)
90
91     print '\nRb87:'
92     m87 = Messung('spin_praez_87.dat')
93     m87.draw(); m87.fit()
94     print 'Bv = %.3f +- %.3f muT' % (m87.Bv/muT, m87.sBv/muT)
95
96
97     if gen_tabellen:
98         m85.write_table()
99         m87.write_table()

```

B.9. Fits zur Relaxationszeit nach Dehmelt (dehmelt_fits.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  import oszi_messung
5  from konst import Q, sec, mus
6  from fit_tools import print_fit_result, create_fit_legend
7  from ROOT import gROOT, TGaxis, TCanvas, TGraphErrors, TF1
8
9  gROOT.SetStyle("Plain")
10 TGaxis.SetMaxDigits(3)
11
12 # Sollen Ergebnisse gespeichert werden?
13 speichere_ergebnisse = True
14
15 # Erweiterung der Oszi-Messung-Klasse fuer Dehmelt-Messungen
16 class Messung(oszi_messung.Messung):
17     def __init__(self, filename):
18         oszi_messung.Messung.__init__(
19             self, 'relax/dehmelt/'+filename, use_errors=True)
20         self.filename = filename
21         self.fcns, self.legend, self.tau, self.stau = [], [], [], []
22
23     def fit(self, r1, r2, params = (0.0, 0.0, 0.0),
24            line_color = 2, print_results = True):
25         f = TF1('f%d%s' % (len(self.fcns), self.filename),
26                '[0] - [1]*exp(-x/[2])', r1, r2)
27         f.SetLineColor(line_color)
28         self.fcns += [f]

```

```

29
30     pnames = [ 'I_{max}', '#DeltaI', '#tau' ]
31     for i, (pn, pv) in enumerate(zip(pnames,params)):
32         f.SetParName(i, pn)
33         f.SetParameter(i, pv)
34     self.g.Fit(f, 'QR0')
35     self.tau += [f.GetParameter(2)*sec]
36     self.stau += [f.GetParError(2)*sec]
37
38     lg = create_fit_legend(f, fstr_nl = False,
39                           lpos = (0.53, 0.15, 0.86, 0.38))
40     lg.Draw();
41     self.legend += [lg]
42
43     if print_results: print_fit_result(f, show_fstr=False)
44     if self.has_canvas: f.Draw('same')
45
46 # fuehre Fits fuer die verschiedenen Messreihe durch
47 m = []
48 nfilter = [0, 1, 2, 3, 4, 5, 7, 6]
49 ymin = [-30e-3, -20e-3, -12e-3, -10e-3, -8e-3, -3e-3, -5e-3, -3e-3]
50 for i in range(len(nfilter)):
51     filename = 'dehm_f%02d.tab' % i
52     print '\n%s:' % filename
53     mi = Messung(filename)
54     mi.draw(x_range=(0, 12e-3), y_min = ymin[i])
55     mi.fit(0.00117, 0.010, (0.0004, 0.006, 0.001))
56     m += [mi]
57
58 # die bestimmten tau-Werte
59 tau = [z.tau[0] for z in m]
60 stau = [z.stau[0] for z in m]
61
62 print '\nErgebnisse:'
63 for mi in m:
64     print '%s: tau = %.3f +- %.3f mus' % (
65         mi.filename, mi.tau[0]/mus, mi.stau[0]/mus)
66
67 # Ergebnisse speichern
68 if speichere_ergebnisse:
69     import shelve
70     d = shelve.open('results.out')
71     d['dehmelt_fits.nfilter'] = nfilter
72     d['dehmelt_fits.tau'] = tau
73     d['dehmelt_fits.stau'] = stau
74     d.close()

```

B.10. Relaxationszeit nach Dehmelt (dehmelt.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  import shelve
5  from array import array
6  from konst import Q, sec, mus, ms
7  from fit_tools import print_fit_result, create_fit_legend
8  from ROOT import gROOT, TGaxis, TCanvas, TGraphErrors, TF1
9
10 gROOT.SetStyle("Plain")
11 TGaxis.SetMaxDigits(3)
12
13 # Sollen LaTeX-Tabellen erzeugt werden?
14 gen_tabellen = False
15
16 # Oeffne Ergebniss-Datenbank
17 d = shelve.open('results.out')
18
19 # lade relative Intensitaeten der Neutralfilter
20 bez = d['neutral_filter.bez']
21 Ir = d['neutral_filter.Ir']
22 sIr = d['neutral_filter.sIr']
23
24 # lade die aus den Fits ermittelten tau-Werte und die Nr. der dafuer
25 # verwendeten Neutralfilter
26 nfilter = d['dehmelt_fits.nfilter']
27 tau = d['dehmelt_fits.tau']
28 stau = d['dehmelt_fits.stau']
29
30 # Schliesse Ergebniss-Datenbank
31 d.close()
32
33 # Bringe Filter-Bezeichnungen in die richtige Reihenfolge
34 bez = [bez[i] for i in nfilter]
35
36 # Intensitaeten der verwendeten Neutralfilter (richtige Reihenfolge)
37 Ir = [Ir[i] for i in nfilter]
38 sIr = [sIr[i] for i in nfilter]
39
40 # Kehrwert der tau-Werte [1/sec]
41 rtau = [1./z for z in tau]
42 srtau = [sz/z**2 for z, sz in zip(tau, stau)]
43
44 # Erzeuge Canvas und Graph
45 c = TCanvas('c', 'Relaxationszeit, Dehmelt'); c.SetGrid()
46 g = TGraphErrors(len(Ir), array('d', Ir), array('d', [sec*z for z in rtau]),
47                 array('d', sIr), array('d', [sec*z for z in srtau]))
48 g.SetTitle(';relative Intensit#ddot{a}t'
49           + ' ;reziproke Orientierungszeit 1/#tau [1/s]')
```

```

50 g.SetMarkerStyle(3); g.SetMarkerColor(2)
51 g.Draw('AP')
52
53 # linearer Fit
54 f = TF1('f', '[0]*x + [1]')
55 for i, pn in enumerate(['a', 'b']): f.SetParName(i, pn)
56 g.Fit(f, 'Q'); print_fit_result(f, show_fstr = False)
57 lg = create_fit_legend(f, fit_name = 'Linearer Fit', fstr_nl = False,
58                       lpos = (0.56, 0.14, 0.88, 0.36))
59 lg.Draw()
60
61 # Y-Achsen-Abschnitt der Fit-Geraden
62 b, sb = f.GetParameter(1)/sec, f.GetParError(1)/sec
63
64 # berechne Relaxationszeit
65 Tr, sTr = 1.0/b, sb/b**2
66
67 print 'Relaxationszeit: Tr = %.3f +- %.3f ms' % (Tr/ms, sTr/ms)
68
69 # erzeuge LaTeX-Tabellen
70 if gen_tabellen:
71     import texgen
72     ftab = file('tables/dehmelt.out.tex', 'w')
73     s = texgen.table_dehmelt(bez, Ir, sIr, tau, stau, rtau, srtau)
74     ftab.write(s); ftab.close()

```

B.11. Relaxation nach Franzen (franzen_mehrfach.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from array import array
5  from konst import Q, ms, sec, V
6  from oszi_messung import Messung
7  from fit_tools import print_fit_result, create_fit_legend
8  from ROOT import gROOT, TAxis, TCanvas, TGraphErrors, TF1
9
10 gROOT.SetStyle("Plain")
11 TAxis.SetMaxDigits(3)
12
13 # lese Daten aus Datei
14 nr, t0, ta = [], [], []
15 for line in file('messdaten/franzen_offs.dat', 'r'):
16     # Nr, Offset [ms], Anstieg [ms]
17     if (len(line.strip()) != 0) and (line.lstrip()[0] != '#'):
18         toks = line.split(); assert len(toks) == 3
19         nr += [int(toks[0])]
20         t0 += [float(toks[1])*ms]
21         ta += [float(toks[2])*ms]
22

```

```

23 # Dateinamen
24 filenames = ['relax/franzen/franz_%02d.tab' % i for i in nr]
25
26 # lade Messungen
27 m = []
28 for i, fn in enumerate(filenames):
29     mi = Messung(fn, use_errors=False, time_offs=-ta[i]/sec+5e-3); m += [mi]
30     #mi.draw(x_range = (0, 40e-3))
31
32 # zeichne mehrere Messungen in ein Diagramm
33 mx = m[-1]
34 mx.draw(x_range = (0, 35e-3), draw_lines = True)
35 for i, mi in enumerate(m[:-1]):
36     mi.g.SetMarkerStyle(6)
37     mi.g.SetMarkerColor(i+2)
38     mi.g.SetLineColor(i+2)
39     mi.g.Draw('sameL')

```

B.12. Routinen für Oszilloskop-Daten (oszi_messung.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from sets import Set
5  from array import array
6  from ROOT import TCanvas, TGraphErrors, TAxis
7
8  class Messung:
9      def __init__(self, filename, nch = 1, use_errors = False, time_offs = 0.0):
10         self.filename = filename
11         self.has_canvas = False
12
13         # oeffne Datei und ueberspringe erste Zeile
14         f = file('messdaten/'+filename, 'r'); f.readline()
15
16         # Zeit [s], CH1 [V], CH2[V]
17         self.data = [line.replace(',','.') for line in f.readlines()]
18         count = self.count = len(self.data)
19
20         # extrahiere Zeit und Kanäle
21         self.t = [float(z[0]) + time_offs for z in self.data]
22         if nch == 1: self.ch = [float(z[1]) for z in self.data]
23         else: self.ch = [float(z[2]) for z in self.data]
24         self.st, self.sch = 0.0, 0.0
25
26         # beruecksichtige die seltsamen Fehler der Oszi-Daten
27         if use_errors:
28             l = list(Set(self.ch)); l.sort()
29             dl = [l[i+1]-l[i] for i in range(len(l)-1)]
30             if dl: self.sch = min(dl)

```

```

31
32     # erzeuge Graphen
33     self.g = TGraphErrors(
34         count, array('d', self.t), array('d', self.ch),
35         array('d', [self.st]*count), array('d', [self.sch]*count))
36     self.g.SetTitle(';Zeit t [s];Spannung U [V]')
37     self.g.GetAxis().SetRangeUser(self.t[0], self.t[-1])
38     self.g.GetAxis().SetDecimals()
39     self.g.GetAxis().SetDecimals()
40
41     def draw(self, show_grid = True, draw_lines = False,
42             x_title_offs = 1.0, y_title_offs = 1.0,
43             x_title = None, y_title = None,
44             y_min = None, y_max = None, x_range = None):
45         c = self.canvas = TCanvas('c_'+self.filename, self.filename)
46         self.has_canvas = True
47         if show_grid: c.SetGrid()
48
49         g = self.g
50         g.GetAxis().SetTitleOffset(x_title_offs)
51         g.GetAxis().SetTitleOffset(y_title_offs)
52
53         if x_title != None: g.GetAxis().SetTitle(x_title)
54         if y_title != None: g.GetAxis().SetTitle(y_title)
55
56         if y_min != None: g.GetHistogram().SetMinimum(y_min)
57         if y_max != None: g.GetHistogram().SetMaximum(y_max)
58         if x_range != None: g.GetAxis().SetRangeUser(x_range[0], x_range[1])
59
60         if draw_lines: g.Draw('AL')
61         else: g.SetMarkerStyle(6); g.Draw('AP')

```

B.13. Konstanten und angegebene Werte (konst.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from math import cos, pi
5  from Scientific.Physics.PhysicalQuantities import PhysicalQuantity as Q
6
7  # Physikalische Konstanten
8  c    = Q('1 c')      # Lichtgeschwindigkeit
9  me   = Q('1 me')    # Ruhemasse des Elektrons
10 h    = Q('1 hplanck') # Plancksche Konstante
11 hbar = Q('1 hbar')   # Plancksches Wirkungsquantum
12 muB  = Q('1 e*hbar/(2.*me)').inUnitsOf('J/T') # Bohrsches Magneton
13
14 # Einheiten
15 keV  = Q('1 keV')   # keV
16 A    = Q('1 A')     # Ampere

```

```

17 mA = Q('1 mA') # Milliampere
18 V = Q('1 V') # Volt
19 T = Q('1 T') # Tesla
20 muT = Q('1 muT') # Mikrottesla
21 sec = Q('1 s') # Sekunden
22 ms = Q('1 ms') # Milisekunden
23 mus = Q('1 mus') # Mikrosekunden
24 GHz = Q('1 GHz') # GHz-Einheit
25 kHz = Q('1 kHz') # kHz-Einheit
26
27 # Freier Spektralbereich des Etalons
28 FSR, sFSR = Q(9924.0, 'MHz'), Q(30.0, 'MHz')

```

B.14. Hilfsroutinen für Fits (fit_tools.py)

```

1 #!/usr/bin/python
2 # -*- coding: iso-8859-1 -*-
3
4 '''
5 Fit-Hilfsroutinen
6 '''
7
8 from ROOT import TLegend, TF1
9
10 def create_fit_legend(f, fcn_name = 'f(x)', fit_name = 'Fit',
11                      show_fstr = True, fstr_nl = True, fstr = [],
12                      lpos = (0.59, 0.65, 0.89, 0.88)):
13     '''
14     Erzeugt eine Legende fuer die uebergebene Fitfunktion.
15     f: Die Fitfunktion
16     fcn_name: Bezeichnung der Fitfunktion
17     fit_name: Bezeichnung des Fits
18     show_fstr: Fitfunktion mit ausgeben
19     fstr_nle: Fitfunktion in neuer Zeile ausgeben
20     fstr: Liste zum Ueberschreiben der Funktionsformeldarstellung
21     lpos: Die Position der Legende
22     '''
23     lg = TLegend(lpos[0], lpos[1], lpos[2], lpos[3])
24     lg.SetFillColor(0)
25     lg.SetHeader('')
26
27     par_count = f.GetNpar()
28     if show_fstr and not fstr:
29         ft = f.GetTitle()
30         for i in range(par_count):
31             ft = ft.replace('[%d]' % i, f.GetParName(i))
32         if fstr_nl:
33             lg.AddEntry(f, '%s: %s' % (fit_name, fcn_name), 'l')
34             lg.AddEntry(f, '%s = %s' % (fcn_name, ft), '')
35     else:

```

```

36         lg.AddEntry(f, '%s: %s = %s' % (fit_name, fcn_name, ft), 'l')
37     else:
38         lg.AddEntry(f, '%s: %s' % (fit_name, fcn_name), 'l')
39
40     if fstr:
41         for fstri in fstr:
42             lg.AddEntry(f, fstri, '')
43
44     ndf = f.GetNDF()
45     if ndf > 0:
46         chisq = f.GetChisquare()
47         lg.AddEntry(f, '#chi2/ndf = %.2f/%d = %.2f' % (
48             chisq, ndf, chisq/ndf), '')
49
50     for i in range(par_count):
51         pn, pv, pe = f.GetParName(i), f.GetParameter(i), f.GetParError(i)
52         lg.AddEntry(f, '%s = %.4g #pm %.4g' % (pn, pv, pe), '')
53
54     lg.AddEntry(f, '', '')
55
56     return lg
57
58 def print_fit_result(f, fcn_name = 'f(x)', show_fstr = True):
59     '''
60     Gibt ein huebscheres Fit-Ergebnis aus.
61     f:          Die Fitfunktion
62     fcn_name:   Bezeichnung der Fitfunktion
63     show_fstr:  Fitfunktion mit ausgeben
64     '''
65     par_count = f.GetNpar()
66
67     pnames = []
68     max_pname_len = 0
69     for i in range(par_count):
70         pn = f.GetParName(i)
71         for c in '#_{}':
72             pn = pn.replace(c, '')
73         pnames += [pn]
74         max_pname_len = max(max_pname_len, len(pn))
75
76     if show_fstr:
77         ft = f.GetTitle()
78         for i in range(par_count):
79             ft = ft.replace('[%d]' % i, pnames[i])
80         print '%s = %s' % (fcn_name, ft)
81
82     ndf = f.GetNDF()
83     if ndf > 0:
84         chisq = f.GetChisquare()
85         print 'chisq/ndf = %.2f/%d = %.4f' % (chisq, ndf, chisq/ndf)
86
87     ostr = '%' + '%d' % (max_pname_len+1) + 's = %10g +- %g'

```

```

88     for i in range(par_count):
89         pn, pv, pe = pnames[i], f.GetParameter(i), f.GetParError(i)
90         print ostr % (pn, pv, pe)
91
92 def get_fit_result(f):
93     '''
94     Liefert ein Dictionary mit den Fitergebnissen.
95     f: Die Fitfunktion
96     '''
97     par_count = f.GetNpar()
98
99     pnames = []
100    for i in range(par_count):
101        pn = f.GetParName(i)
102        for c in '#_{}':
103            pn = pn.replace(c, '')
104            pnames += [pn]
105
106    d = {}
107
108    d['ndf'] = f.GetNDF()
109    d['chisq'] = f.GetChisquare()
110    if d['ndf'] > 0:
111        d['rchisq'] = d['chisq'] / d['ndf']
112
113    for i in range(par_count):
114        d[pnames[i]] = (f.GetParameter(i), f.GetParError(i))
115
116    return d

```

B.15. Routinen zur Erzeugung von \LaTeX -Tabellen (texgen.py)

```

1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from konst import Q, mus, ms, GHz, A, mA, kHz, muT
5
6  def begin_table(*heading):
7      s = '\\begin{tabular}{%s}\n\\hline\n' % ('|' + len(heading)*'c|')
8      s += ' & '.join(heading)
9      s += ' \\\\n\\hline\n'
10     return s
11
12 def end_table(draw_hline=True):
13     s = '\\end{tabular}'
14     if draw_hline: return '\\hline\n' + s
15     else: return s
16
17 def table_neutral_filter(data):
18     s = begin_table(r'Bezeichnung', r'Filter $x_{\text{alt}}$'),

```



```

71     return s + end_table(draw_hline=False)
72
73 def table_dres_magnetfeld(B):
74     assert len(B) % 2 == 0
75     s = begin_table(r'Messung',
76                   r'${85}$Rb: $B_{\text{hor}}$ [$\unit{\mu T}$]',
77                   r'${87}$Rb: $B_{\text{hor}}$ [$\unit{\mu T}$]')
78     for i in xrange(0, len(B), 2):
79         B1, sB1 = B[i]
80         B2, sB2 = B[i+1]
81         l = '%d & ' % (i/2+1)
82         l += '$%.2f \pm %.2f$ & ' % (B1/muT, sB1/muT)
83         l += '$%.2f \pm %.2f$ \\\n' % (B2/muT, sB2/muT)
84         s += l.replace('.', ',\\!')
85     return s + end_table()
86
87 def table_dres_kernspin(I):
88     assert len(I) % 2 == 0
89     s = begin_table(r'Messung',
90                   r'${85}$Rb: Kernspin $I$',
91                   r'${87}$Rb: Kernspin $I$')
92     for i in xrange(0, len(I), 2):
93         I1, sI1 = I[i]
94         I2, sI2 = I[i+1]
95         l = '%d & ' % (i/2+1)
96         l += '$%.3f \pm %.3f$ & ' % (I1, sI1)
97         l += '$%.3f \pm %.3f$ \\\n' % (I2, sI2)
98         s += l.replace('.', ',\\!')
99     return s + end_table()
100
101 def table_spin_praez(m):
102     s = begin_table(r'Nr', r'Spulenstrom $I$ [$\unit{\mu A}$]',
103                   r'Zeit t [$\unit{\mu s}$]', r'Anzahl $n$',
104                   r'Frequenz $\nu$ [$\unit{kHz}$]')
105     for i in range(len(m.nr)):
106         l = '%d & ' % m.nr[i]
107         l += '$%.1f \pm %.1f$ & ' % (m.I[i]/mA, m.sI[i]/mA)
108         l += '$%.2f \pm %.2f$ & ' % (m.t[i]/mus, m.st[i]/mus)
109         l += '%d & ' % m.n[i]
110         l += '$%.2f \pm %.2f$ \\\n' % (m.nu[i]/kHz, m.snu[i]/kHz)
111         s += l.replace('.', ',\\!')
112     return s + end_table()
113
114 def table_dehmelt(bez, Ir, sIr, tau, stau, rtau, srtau):
115     s = begin_table(r'Neutralfilter', r'rel. Intensitat $I_{\text{rel}}$',
116                   r'Orientierungszeit $\tau$ [$\unit{ms}$]',
117                   r'$\tau^{-1}$ [$\unit{(ms)^{-1}}$]')
118     for i in range(len(bez)):
119         l = '\\verb|s| & ' % bez[i]
120         l += '$%.3f \pm %.3f$ & ' % (Ir[i], sIr[i])
121         l += '$%.3f \pm %.3f$ & ' % (tau[i]/ms, stau[i]/ms)
122         l += '$%.3f \pm %.3f$ \\\n' % (rtau[i]*ms, srtau[i]*ms)

```

```
123     s += l.replace('.', ',\\!')
124     return s + end_table()
```

B.16. Weitere Hilfsroutinen (tools.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  def gew_mittel(xsx):
5      '''gew_mittel(list(float,float)) -> (float, float)
6      xsx : Liste aus Tupeln der Messwerte mit jeweiligen Fehlern
7      ->  Tupel (gx, sgx) aus gewichtetem Mittel und dessen Fehler'''
8      suma = 0. * xsx[0][0] / xsx[0][1]**2
9      sumb = 0. * 1. / xsx[0][1]**2
10     for xi,sxi in xsx:
11         suma += xi / sxi**2
12         sumb += 1. / sxi**2
13     return (suma/sumb, sumb**(-0.5))
14
15 def arith_mittel(x):
16     '''arith_mittel(list(float)) -> float
17     x : Liste aus Messwerte
18     ->  arithmetisches Mittel der Messwerte'''
19     sumx = 0. * x[0]
20     for xi in x:
21         sumx += xi
22     return sumx / len(x)
```