

**Department of Physics and Astronomy**  
**University of Heidelberg**

Bachelor thesis in Physics  
submitted by

**Miriam Lena Gerharz**

born in Lahnstein (Germany)

**2019**



# Dynamical polarization control in X-ray quantum optics

This Bachelor thesis has been carried out by  
Miriam Lena Gerharz  
at the  
Max Planck Institute for Nuclear Physics in Heidelberg  
under the supervision of  
apl. Prof. Dr. Jörg Evers



## **Dynamische Polarisationskontrolle in der Röntgenquantenoptik**

Wann immer mit polarisiertem Licht gearbeitet wird, können Situationen auftreten, bei denen eine andere als die gegebene Polarisation erforderlich ist. Im optischen Bereich existieren Wellenplättchen, die eine Konvertierung von Polarisationen ermöglichen. Das Funktionsprinzip solcher Wellenplättchen basiert auf relativen Unterschieden von Brechungsindizes. Weil es für Röntgenstrahlung nicht einfach ist, ausreichend unterschiedliche Brechungsindizes zu erhalten, ist es schwierig dieses Prinzip auf den Röntgenbereich zu übertragen.

In dieser Arbeit schlagen wir einen neuen Ansatz vor, um die benötigten Phasenshifts zu erhalten, unter Verwendung von Kernstreuung. Das präsentierte Setup hat den Vorteil, dass die Polarizationen dynamisch kontrolliert werden können.

Wir diskutieren zunächst das Funktionsprinzip unseres neuen Ansatzes. Dabei gehen wir neben analytischen Erklärungen auch auf numerische Simulationen und Optimierungen mit dem python-Paket `pynuss` ein. Weitere Anwendungsmöglichkeiten unseres Ansatzes wie ein zirkulärer Polarisationsfilter,  $\frac{\lambda}{2}$ -Plättchen, schnelle Lichtschalter und präzise Messungen von Verschiebungen werden vorgestellt. Abschließend wird unser Setup mit existierenden optischen Elementen verglichen.

## **Dynamical polarisation control in X-ray quantum optics**

Whenever working with polarized light, there might be situations, in which another than the given polarization is required. In the optical regime, there exist wave plates, which are able to convert polarizations. The working principle of such wave plates is based on relative differences of refractive indices. Since it is hard to achieve sufficiently different refractive indices for X-rays, it is difficult to mimic this approach in the X-ray regime.

In this thesis, we propose a new approach to produce the necessary phase shift in the X-ray regime, by employing nuclear forward scattering. The suggested setup has the advantage, that the polarization can be dynamically controlled.

First, we discuss the working principle of our new approach. This includes analytical explanations as well as numerical simulations and optimization with the python-package `pynuss`. Subsequently, further applications such as circular polarization filters,  $\frac{\lambda}{2}$ -wave plates, fast light switches and precise displacement measurements will be presented. We conclude this thesis by a comparison with existing optical elements.



# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Theoretical basics</b>	<b>3</b>
2.1. General definitions . . . . .	3
2.1.1. Jones vectors and matrices . . . . .	3
2.1.2. Intensity and purity . . . . .	4
2.2. Standard $\frac{\lambda}{4}$ - and $\frac{\lambda}{2}$ -wave plate . . . . .	5
2.2.1. Functionality . . . . .	5
2.2.2. Effect of angle $\alpha$ and phase shift $\Delta\phi$ . . . . .	7
2.3. Combination of two wave plates . . . . .	8
2.3.1. Description in Jones formalism . . . . .	8
2.3.2. Effect of different phase shifts $\Delta\phi_1$ . . . . .	10
2.3.3. Effect of different angles $\alpha_i$ . . . . .	10
<b>3. Quantum optical description of the new approach</b>	<b>12</b>
3.1. Nuclear transitions of $^{57}\text{Fe}$ . . . . .	12
3.2. Phase modulation through moving sample . . . . .	13
3.2.1. Calculation of frequency spectra . . . . .	14
3.2.2. I(ntensity increase) and T(ransformation) . . . . .	16
3.2.3. Combination of two samples. . . . .	17
3.2.4. Time spectra . . . . .	18
3.3. New approach for polarization conversion . . . . .	19
3.3.1. Setup . . . . .	20
3.3.2. Functionality . . . . .	21
<b>4. Results</b>	<b>24</b>
4.1. One sample and general effects . . . . .	24
4.1.1. Frequency spectrum of one sample . . . . .	24
4.1.2. Real vs. ideal motion . . . . .	26
4.1.3. Shift of intensity of maximum . . . . .	27
4.1.4. Influence of thickness of targets $d$ . . . . .	29
4.1.5. Influence of magnetic angles $\alpha$ and displacement $\Delta z$ . . . . .	30

4.2.	Two samples . . . . .	31
4.2.1.	Frequency spectrum of two samples . . . . .	31
4.2.2.	Influence of different displacements $\Delta z_i$ . . . . .	33
4.2.3.	Influence of different angles $\alpha_i$ . . . . .	34
4.3.	Optimization . . . . .	35
4.4.	Behavior in time space . . . . .	37
4.4.1.	All frequencies . . . . .	37
4.4.2.	Around $-31.1\gamma$ . . . . .	39
4.5.	Other applications . . . . .	39
4.5.1.	Circular polarization filter . . . . .	41
4.5.2.	$\frac{\lambda}{2}$ -wave plate and light switch . . . . .	41
4.5.3.	Highly precise displacement measurement . . . . .	43
<b>5.</b>	<b>Comparisons with optical elements</b>	<b>45</b>
5.1.	Standard wave plate . . . . .	45
5.2.	Pockels cells . . . . .	46
<b>6.</b>	<b>Summary and Outlook</b>	<b>47</b>
<b>A.</b>	<b>Used python-scripts</b>	<b>49</b>
A.1.	Analysis of polarization transformation . . . . .	49
A.1.1.	Frequency space . . . . .	49
A.1.2.	Time space . . . . .	63
A.2.	Applications . . . . .	70
A.2.1.	Circular polarization filter . . . . .	70
A.2.2.	$\frac{\lambda}{2}$ -wave plate and light switch . . . . .	73
	<b>List of Figures</b>	<b>78</b>
	<b>List of Tables</b>	<b>79</b>

# 1. Introduction

Many experiments working with X-rays use synchrotron radiation (e.g. [1]). Unfortunately, these sources only deliver light with a fixed polarization. In some cases, it might be interesting to change the polarization in order to do other experiments. In the optical range, wave plates are a well established technique to convert polarizations into each other. A crystal with two different diffraction indices in different directions is used in order to give a relative phase shift between the two directions. As will be explained in chapter 2, a phase shift of  $\frac{\pi}{2}$  converts linear polarization into circular one and vice versa. Unfortunately, in the X-ray domain all refractive indices are close to 1 ([2]). Thus, it is much more difficult to achieve the necessary phase shift. Nevertheless, it was possible to build a  $\frac{\lambda}{4}$ -wave plate for X-rays (see [3]). In their approach, the same principle as in the optical regime is used. In a diamond single-crystal it was possible to achieve a sufficient difference in the two diffraction indices.

In this thesis, a completely new technique to transform polarizations and dynamically control them will be introduced. The idea is based on the work of P. Reiser, K.P. Heeg et. al. ([1], [4] and [5]) and uses nuclear forward scattering with synchrotron radiation. Shifting a sample by parts of the resonant wavelength right after its excitation makes it possible to dynamically modify the real part of the refractive index while the imaginary part remains unaffected. This way, relative phase shifts between the scattered and unscattered synchrotron pulse of order  $\pi$  can be achieved. While without motion, the scattered and unscattered light would interfere destructively, by applying a shift of  $\frac{\lambda}{2}$ , where  $\lambda$  is the resonant wave length, the scattered light obtains a phase shift of  $\pi$  resulting in constructive interference. With this technique, the intensity at the resonant wave length can be increased. This entire process will be explained in more detail in chapter 3.

The new idea is, to use this formalism to apply a phase shift of  $\frac{\pi}{4}$ . As we will explain in section 3.3, this enables us to transform linearly into circularly polarized light and vice versa. We will discuss this transformation, that has a similar result like a  $\frac{\lambda}{4}$ -wave plate, as a first example of adaptive optics realized by motion in detail in chapter 4. Since synchrotrons commonly deliver linearly polarized light (in our experiments), let us focus on the conversion of linear into circular polarization. Nevertheless, the entire calculations also apply to the transformation of circular into linear polarization. By controlling the motion of the sample, the phase shift and consequently the polarization can be dynamically controlled in frequency space. In principle, arbitrary phase shifts can

## 1. Introduction

be used. This makes it possible to construct elements which have similar functions as  $\frac{\lambda}{2}$ -wave plates and polarization filters. Also  $\frac{\lambda}{2}$ -wave plates can be used for a fast light switch. Besides, the setup could be used to do highly sensitive measurements of displacement. We will discuss these ideas in section 4.5.

In the future, we will try to better understand the time spectra in order to achieve more control about the time behavior of the polarizations. Hence, our idea could be further developed into an element, which has the same functionality as a Pockels cell.

The new technique was originally developed for the X-ray regime. Nevertheless, it could also function with electronic instead of nuclear transitions and a line splitting due to the Zeemann-effect. Hence, the proposed setup could also work in the optical range. Despite all discussed optical elements being available in the optical range, there could be some new effects, e.g. the sensing, that might also be interesting to have in this regime.

While presenting the theoretical descriptions and investigations, we always wish to keep the experimental possibilities in mind.

## 2. Theoretical basics

In this chapter, the theoretical basics to do calculations with polarizations will be introduced. Furthermore, the function of a ordinary  $\frac{\lambda}{4}$ - and  $\frac{\lambda}{2}$ -wave plate will be explained. Particular focus is given to investigate the influence of the effect of different phase shifts and angles at which the wave plates are positioned.

### 2.1. General definitions

We need a general description for polarizations of electromagnetic-waves and how to modify them. Besides, we want to be able to assign a scalar value to a given polarization. That way, we can characterize it by a single value, which makes it easier to compare the result for varying parameters.

#### 2.1.1. Jones vectors and matrices

To describe the optical setup, we will use the so-called *Jones vectors and matrices*. All equations in this section are adapted from [6, chap. 9].

Considering a light beam in z-direction, any electromagnetic wave can be described as

$$\vec{E}(z, t) = \vec{J} \cdot e^{i(kz - \omega t)}. \quad (2.1)$$

The Jones vector  $\vec{J}$  represents the polarization in the  $xy$ -plane, the wavenumber  $k$  and the frequency  $\omega$ . Since there is cannot be any polarization in the direction of propagation, we can focus on the  $xy$ -plane. For linear ( $\pi_0$ ) and perpendicular linearly ( $\pi_{0,\perp}$ ) as well as circularly right ( $\sigma_+$ ) and left ( $\sigma_-$ ) polarized light the Jones vectors are:

$$\pi_0 : \vec{J}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \pi_{0,\perp} : \vec{J}_{0,\perp} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \sigma_{\pm} : \vec{J}_{\pm} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i \end{pmatrix}. \quad (2.2)$$

Whenever the direction of the linear polarization does not matter, we will call it just  $\pi$ . Because of  $(-1)^2 = 1^2$  it does not matter whether we take the polarization vector in its original or opposite direction. We allow the opposite vector as an equivalent description

$$\vec{J} \equiv -\vec{J}. \quad (2.3)$$

## 2. Theoretical basics

Besides describing the electromagnetic wave, we also have to describe the optical elements. A phase retarder is represented by

$$\Phi(\phi_x, \phi_y) = \begin{pmatrix} \exp(i\phi_x) & 0 \\ 0 & \exp(i\phi_y) \end{pmatrix}. \quad (2.4)$$

It provides a phase shift  $\phi_x$  in x-direction and  $\phi_y$  in y-direction. A circular polarization filter can be formulated as

$$\Sigma_{\pm} = \frac{1}{2} \begin{pmatrix} 1 & \pm i \\ \mp i & 1 \end{pmatrix}. \quad (2.5)$$

It filters out  $\sigma_{\pm}$  polarized light. If an optical element  $\mathbf{M}_0$  is rotated by the angle  $\alpha$ , it can be written as

$$\mathbf{M}(\alpha) = \mathbf{R}(-\alpha)\mathbf{M}_0\mathbf{R}(\alpha), \quad (2.6)$$

with the rotation matrix

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}. \quad (2.7)$$

### 2.1.2. Intensity and purity

As we focus on the electric field, the intensity  $I$  is given by

$$I(z, t) = \left| \vec{E}(z, t) \right|^2 = \left| \vec{J} \right|^2 \cdot \underbrace{\left( e^{i(kz - \omega t)} \right)^2}_{I_0}, \quad (2.8)$$

where  $I_0$  is the intensity of the electric field before manipulating the polarizations.

To make comparisons between different configurations especially in the numerical analysis easier, we require a scalar measure for the kind of polarization, i.e.  $\sigma_+$ ,  $\pi$  or  $\sigma_-$ . Hence, we define the *purity*

$$P = \frac{I_+ - I_-}{I_+ + I_-}, \quad (2.9)$$

with  $I_{\pm}$  being the intensities that would be measured after the application of a  $\sigma_{\pm}$  polarization filter like eq. (2.5). With  $I_{\pm} = I_0$  and  $I_{\mp} = 0$  for  $\sigma_{\pm}$  as well as  $I_+ = I_- = \frac{I_0}{2}$  for  $\pi$ , the purity for circular and linear polarization can be calculated to be

$$\sigma_- : P = -1, \quad \pi/\pi_{0,\perp} : P = 0, \quad \sigma_+ : P = +1. \quad (2.10)$$

It can be easily seen, that  $P \in [-1, 1]$ . All other values indicate a mixture of these special cases. It can be directly seen, that with this measure we cannot distinguish between the two types of linear polarization. As we focus on linear polarization being converted into circular one, this is not a problem for our analysis. If it would be important to distinguish between the two linear polarizations, instead of the intensities after circular polarization

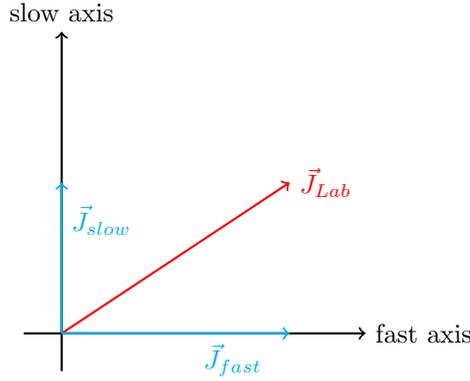


Figure 2.1.: In this figure the eigenbasis of the wave plate is shown (black). In addition, an arbitrary polarization vector of incoming light  $\vec{J}_{Lab}$  is displayed (red) as well as its projections onto the eigenvectors (cyan).

filters, the intensities after two perpendicular linear polarization filters could be employed to define purity. In this case, the distinguishability between the two circular polarization filters would be lost.

## 2.2. Standard $\frac{\lambda}{4}$ - and $\frac{\lambda}{2}$ -wave plate

Standard  $\frac{\lambda}{4}$ - and  $\frac{\lambda}{2}$ -wave plates use crystals that have different refractive indices in particular directions. Along the axis with the lower refractive index, the speed of light in the medium is higher resulting in a relative phase shift between the light beams along the two axis. If this phase shift is applied properly, linearly polarized light will be transformed into circularly polarized light.

### 2.2.1. Functionality

First, as it can be seen in eq. (2.8), for the intensity absolute phases do not matter. The relative phase shift due to different refractive indices is given by (see [7, chap. 8])

$$\Delta\phi = \frac{d}{\lambda} \cdot 2\pi \cdot (n_{slow} - n_{fast}), \quad (2.11)$$

with the thickness of the wave plate  $d$ , the wavelength of the light to be converted  $\lambda$  and the refractive indices  $n_{fast}$  and  $n_{slow}$ . It can be most easily described in the eigenbasis of the wave plate. A schematic can be seen in fig. 2.1. Here, we employ the axes with different refractive indices as a coordinate system. In this eigenbasis of the wave plate, the Jones-vector  $\vec{J}_{Lab}$  of an incoming polarization in the laboratory system can be uniquely decomposed

$$\vec{J}_{Lab} = \vec{J}_{fast} + \vec{J}_{slow}. \quad (2.12)$$

## 2. Theoretical basics

Applying this decomposition, only one direction receives the relative phase shift. Hence, eq. (2.4) becomes

$$\Phi_{Eigen}(\Delta\phi) = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\Delta\phi) \end{pmatrix}, \quad (2.13)$$

in the eigenbasis. To transform the matrices back into the laboratory system, we apply eq. (2.6) and thereby rotate the wave plate by an angle of  $\alpha$ . Here,  $\alpha$  is the angle between the x-axis of the laboratory system and the slow axis of the eigenbasis of the wave plate. We can then formulate the Jones matrix of a wave plate causing a phase shift  $\Delta\phi$  and being rotated by an angle of  $\alpha$  relative to the laboratory system

$$\Phi_{Lab}(\Delta\phi, \alpha) = \begin{pmatrix} \cos^2(\alpha) + \exp(i\Delta\phi) \sin^2(\alpha) & \cos(\alpha) \sin(\alpha) [1 - \exp(i\Delta\phi)] \\ \cos(\alpha) \sin(\alpha) [1 - \exp(i\Delta\phi)] & \exp(i\Delta\phi) \cos^2(\alpha) + \sin^2(\alpha) \end{pmatrix}. \quad (2.14)$$

in the laboratory system.

The polarization after such a wave plate is then given by

$$\vec{J}' = \Phi_{Lab}(\Delta\phi, \alpha) \cdot \vec{J}, \quad (2.15)$$

where  $\vec{J}$  is the incoming polarization. To implement eq. (2.14), we first consider two special cases:

$$\begin{aligned} \frac{\lambda}{4}\text{-wave plate: } \Delta\phi &= \frac{\pi}{2}, \alpha = \frac{\pi}{4}, \\ \frac{\lambda}{2}\text{-wave plate: } \Delta\phi &= \pi, \alpha = \frac{\pi}{4}. \end{aligned} \quad (2.16)$$

General values for  $\alpha$  and  $\Delta\phi$  will be discussed in the next subsection. In the two cases eq. (2.15) becomes for the separate polarizations

$$\begin{aligned} \frac{\lambda}{4}: \quad \vec{J}'_0 &= \frac{1}{2} \begin{pmatrix} 1+i \\ 1-i \end{pmatrix} = \mathbf{R}\left(-\frac{\pi}{4}\right) \vec{J}_-, \\ \vec{J}'_{0,\perp} &= \frac{1}{2} \begin{pmatrix} 1-i \\ 1+i \end{pmatrix} = \mathbf{R}\left(-\frac{\pi}{4}\right) \vec{J}_+, \\ \vec{J}'_+ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1+i \\ 0 \end{pmatrix} = \exp\left(\frac{\pi}{4}i\right) \vec{J}_0, \\ \vec{J}'_- &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1-i \end{pmatrix} = \exp\left(-\frac{\pi}{4}i\right) \vec{J}_{0,\perp}. \end{aligned} \quad (2.17)$$

Incoming polarization:	$\pi_0$	$\pi_{0,\perp}$	$\sigma_+$	$\sigma_-$
Polarization after $\frac{\lambda}{4}$ -wave plate:	$\sigma_-$	$\sigma_+$	$\pi_{0,\perp}$	$\pi_0$
Polarization after $\frac{\lambda}{2}$ -wave plate:	$\pi_{0,\perp}$	$\pi_0$	$\sigma_-$	$\sigma_+$

 Table 2.1.: The effect of the two wave plates for  $\alpha = \frac{\pi}{4}$  on different polarizations.

Similarly:

$$\begin{aligned}
 \frac{\lambda}{2} : \quad & \vec{J}_0 = \vec{J}_{0,\perp}, \quad \left( \text{arbitrary } \alpha : \vec{J}_0 = \mathbf{R}(-2\alpha)\vec{J}_0 \right) \\
 & \vec{J}_{0,\perp} = \vec{J}_0, \quad \left( \text{arbitrary } \alpha : \vec{J}_{0,\perp}' = \mathbf{R}(-2\alpha)\vec{J}_{0,\perp} \right) \\
 & \vec{J}_+ = \frac{1}{\sqrt{2}} \begin{pmatrix} i \\ 1 \end{pmatrix} = \vec{J}_-, \\
 & \vec{J}_- = \frac{1}{\sqrt{2}} \begin{pmatrix} -i \\ 1 \end{pmatrix} = \vec{J}_+.
 \end{aligned} \tag{2.18}$$

Later, we will be interested in the intensity and not in the electromagnetic field. In this case the global phase can be ignored. Hence, a  $\frac{\lambda}{4}$ -wave plate interchanges circularly and linearly polarized light while the  $\frac{\lambda}{2}$ -wave plate changes handedness for circularly polarized light and rotates linearly polarized light by  $2\alpha$ . A summary of the effects of both types of wave plates on the different polarizations can be found in table 2.1.

### 2.2.2. Effect of angle $\alpha$ and phase shift $\Delta\phi$

As said before, we will now have a closer look on arbitrary angles  $\alpha$  and phase shifts  $\Delta\phi$ . The result of eq. (2.15) with arbitrary angles  $\alpha$  and phase shifts  $\Delta\phi$  in eq. (2.14) on  $\vec{J}_0 = (1, 0)^T$  is calculated. Then the purity (see eq. (2.9)) is calculated and shown in fig. 2.2 color-coded. Yellow indicates a polarization close to  $\sigma_+$ , blue close to  $\sigma_-$  and green close to  $\pi$ . We see that we have a periodicity of  $\Delta\alpha = \pi$ , which we expect because of eq. (2.3). We also observe a periodicity in the phase shift of  $\pi$ , which we would also expect intuitively due to  $\exp(i\phi) = \exp(i(\phi + 2\pi))$ , which results in a positive or negative sign in eq. (2.13). For  $\alpha = -\frac{n}{2}, n \in \mathbb{Z}$  we cannot achieve any conversion of polarization at all. For  $n \cdot \pi$ , this can be easily seen because eq. (2.14) reduces to eq. (2.13), that has no effect on  $\pi_0$ . In the  $(\frac{1}{2} + n)\pi$  case, eq. (2.13) reduces to  $\exp(i\Delta\phi) \cdot \Phi_{\text{eigen}}(-\Delta\phi)$ . With absolute phases being negligible, this also has no effect on  $\pi_0$ . For all the other angles, the purity depends on the phase shift. The already discussed configurations (see eq. (2.17))

$$\alpha = \pm\frac{\pi}{4} \text{ and } \Delta\phi = \pm\frac{\pi}{2}$$

## 2. Theoretical basics

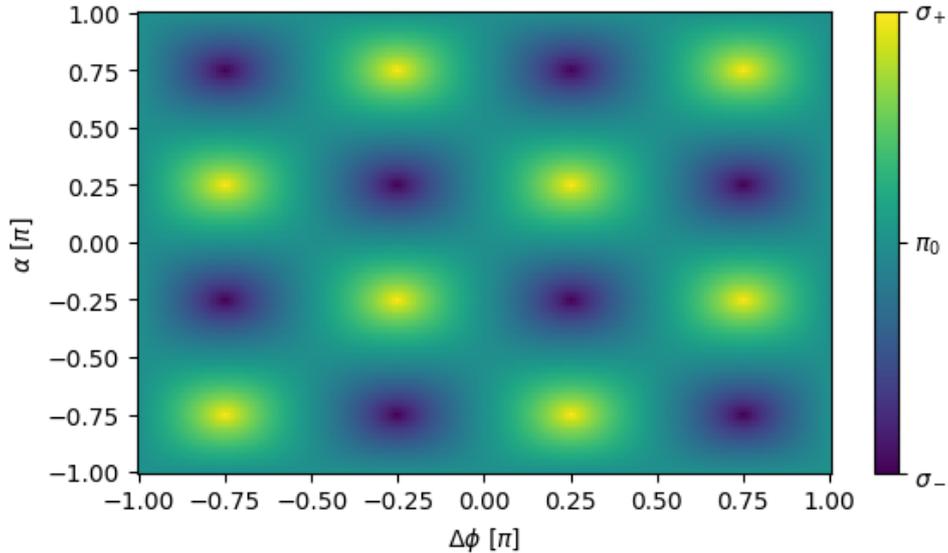


Figure 2.2.: In this figure the dependency of the purity on the angle  $\alpha$  and phase shift  $\Delta\phi$  is shown. The calculation are done in the Jones formalism using the matrices introduced in section 2.1.1.

lead to the a purity of  $P = \pm 1$ . But also the configurations with  $\pm\frac{\pi}{2}$  added onto  $\alpha$  as well as the ones with  $\pm\pi$  added onto  $\Delta\phi$  deliver  $P = \pm 1$ . This is because of  $\sin(\alpha) = -\sin(\alpha \pm \frac{\pi}{2})$  and  $\cos(\alpha) = -\cos(\alpha \pm \frac{\pi}{2})$  just giving signs in eq. (2.6) respectively because of  $\exp(i\phi) = -\exp(i(\phi \pm \pi))$  in eq. (2.13). This is also the explanation for the purity flipping sign when displacement or angle flips sign. For  $\Delta\phi = n \cdot \pi$  we get the  $\frac{\lambda}{2}$ -configuration (see eq. (2.18)), that only rotates the angle of linear polarization. Therefore, this results  $P = 0$  as we would expect for linearly polarized light.

## 2.3. Combination of two wave plates

### 2.3.1. Description in Jones formalism

Later, we wish to apply a phase shift of  $\frac{\pi}{2}$  to convert linear into circular polarizations and vice versa and a phase shift of  $\pi$  to increase the intensity. As we have learned in eq. (2.16), these are the cases of a  $\frac{\lambda}{4}$ - respectively a  $\frac{\lambda}{2}$ -wave plate. Note that later in our approach, the phase shift has to be dynamical in contrast to the static one of an ordinary wave plate. Nevertheless, let us calculate the combination of two wave plates in the static case. In section 4.2 we will compare this to our dynamical approach.

Successive application of wave plates, corresponds to multiplication of Jones matrices

$$\vec{J}' = \Phi(\Delta\phi_2, \alpha_2)\Phi(\Delta\phi_1, \alpha_1)\vec{J}. \quad (2.19)$$

As it will be explained in section 3.3, the transformation works with a phase shift of  $\frac{\pi}{2}$ , we

will associate the transformation (T) with the  $\frac{\lambda}{4}$ -wave plate. In the same way, we identify the intensity increase (I) with the  $\frac{\lambda}{2}$ -wave plate as both work with a phase shift of  $\pi$ . At first, we now have two different possibilities to arrange the two wave plates:

$$\begin{aligned} \text{Config IT: } \Delta\phi_1 = \pi \text{ and } \Delta\phi_2 = \frac{\pi}{2}, \alpha_1 = -\alpha_2 = \frac{\pi}{4}, \\ \text{Config TI: } \Delta\phi_1 = \frac{\pi}{2} \text{ and } \Delta\phi_2 = \pi, \alpha_1 = -\alpha_2 = \frac{\pi}{4}. \end{aligned} \quad (2.20)$$

In the first configuration, we could choose the first angle  $\alpha_1$  in an arbitrary way and ensure that the second one has an angle of  $\frac{\pi}{4}$  with the rotated polarization. In the second configuration, the first angle needs to be chosen to be  $\alpha_1 = \frac{\pi}{4} + n \cdot \frac{\pi}{2}, n \in \mathbb{Z}$  as we have seen in eq. (2.20). Then the second angle can be chosen in an arbitrary way. We choose  $\alpha_1 = -\alpha_2$ , so that for the  $\frac{\lambda}{4}$ -wave plate the angle is different in the two configurations. This will make it easier to detect differences of the two configurations as in one case the transformation takes place with an angle  $\alpha_1$  (TI) and in the other case with  $-\alpha_1$  (IT), which results in a relative sign. For the  $\frac{\lambda}{2}$ -wave plate there is no distinction among  $\alpha_1 = \alpha_2$  and  $\alpha_1 = -\alpha_2$ . This is because we are merely interested in whether we have linearly polarized light or not and not interested in getting  $\vec{J}$  or  $-\vec{J}$ .

In principle, also arbitrary angles  $\alpha_i$  and phase shifts  $\Delta\phi_i$  could be chosen. However, we will focus on the two configurations given above because the setup is easier to understand, if both wave plates have their own role. With other phase shifts or angles, that do not have the periodicity shown in fig. 2.2 (i.e. add  $\frac{\pi}{2}$  on  $\alpha_i$  or  $\Delta\phi_i$ ), the roles would be mixed up and it would be much harder to intuitively understand what happens.

Using eq. (2.19) for the parameters in eq. (2.20) we get the following Jones vectors

$$\begin{aligned} \sigma_+ : \vec{J}_{\text{IT}} &= \exp\left(-i\frac{3\pi}{4}\right) \vec{J}_{0,\perp} \equiv \vec{J}_{0,\perp} & \text{and} & \quad \vec{J}_{\text{TI}} = \exp\left(i\frac{3\pi}{4}\right) \vec{J}_0 \equiv \vec{J}_0, \\ \sigma_- : \vec{J}_{\text{IT}} &= \exp\left(i\frac{3\pi}{4}\right) \vec{J}_0 \equiv \vec{J}_0 & \text{and} & \quad \vec{J}_{\text{TI}} = \exp\left(i\frac{\pi}{4}\right) \vec{J}_{0,\perp} \equiv \vec{J}_{0,\perp}, \\ \pi_0 : \vec{J}_{\text{IT}} &= -\mathbf{R}\left(\frac{\pi}{4}\right) \vec{J}_+ \equiv \vec{J}_+ & \text{and} & \quad \vec{J}_{\text{TI}} = -\mathbf{R}\left(-\frac{\pi}{4}\right) \vec{J}_- \equiv \vec{J}_-, \\ \pi_{0,\perp} : \vec{J}_{\text{IT}} &= -\mathbf{R}\left(\frac{\pi}{4}\right) \vec{J}_- \equiv \vec{J}_- & \text{and} & \quad \vec{J}_{\text{TI}} = -\mathbf{R}\left(-\frac{\pi}{4}\right) \vec{J}_+ \equiv \vec{J}_+. \end{aligned} \quad (2.21)$$

Because of absolute phases and the rotation of the circular polarization vectors  $\vec{J}_{\pm}$  having no physical meaning, we can ignore them. The differences of the two configurations are due to the choice of the angles  $\alpha_1$  and  $\alpha_2$ . Comparing these results with table 2.1, we see that even with the additional  $\frac{\lambda}{2}$ -wave plate, the results look similar to those of a single  $\frac{\lambda}{4}$ -wave plate, despite some additional minus-signs in TI. They are caused by the  $\frac{\lambda}{2}$ -wave plate and cancel with the minus-sign of  $\alpha_2$  for IT.

In this section, we have seen, that an additional  $\frac{\lambda}{2}$ -wave plate with a phase shift of  $\Delta\phi = \pi$  does not harm the transformation of linear into circular polarization.

## 2. Theoretical basics

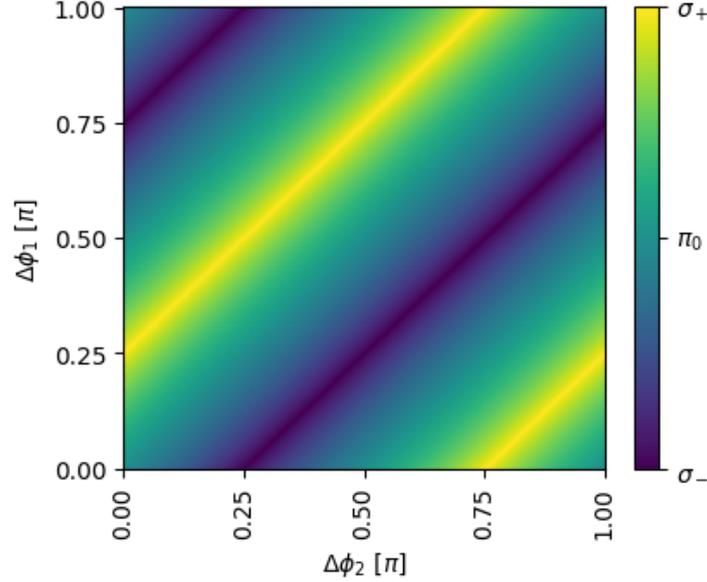


Figure 2.3.: In this figure the purity is shown for different phase shifts. The angles are chosen to be  $\alpha_1 = \frac{\pi}{4}$  and  $\alpha_2 = -\frac{\pi}{4}$ .

### 2.3.2. Effect of different phase shifts $\Delta\phi_1$

To understand the effect of different phase shifts  $\phi_1$  and  $\phi_2$ , we choose the configuration in eq. (2.20) and vary the phase shifts. As the two configurations only differ in phase shifts, it is sufficient to consider them as one configuration. Similar to the analysis of one sample, eq. (2.14) is used to calculate the polarization behind the setup with eq. (2.19) for varying  $\Delta\phi_i$ . Then the purity can be calculated and is displayed in fig. 2.3. Yellow areas indicate  $\sigma_{+-}$ , blue areas  $\sigma_{--}$  and green areas  $\pi$ -polarization. It can be seen, that there is a periodicity of  $\Delta\phi_i = 2\pi$  for both phase shifts in the purity values as discussed in case of one sample (see fig. 2.2). If we would exchange the two phase shifts  $\phi_1$  and  $\phi_2$ , i. e. switching the axis in fig. 2.3, the purity would flip its sign. This is due to our choice of different angles  $\alpha_1 = -\alpha_2 = -\frac{\pi}{4}$ , which gives an additional minus-sign in the second component of eq. (2.14). Furthermore, we see that for every  $\Delta\phi_i$ -step of  $\frac{\pi}{2}$  the purity changes sign. This is due to  $\exp(i\phi) = -\exp(i(\phi + \frac{\pi}{2}))$ , which leads to an additional minus sign in eq. (2.13).

### 2.3.3. Effect of different angles $\alpha_i$

We can now explore the effect of different angles  $\alpha_i$  for both of the two configurations in eq. (2.20). Again, eq. (2.14) and eq. (2.19) were used to calculate the polarization for varying angles  $\alpha_1$  and  $\alpha_2$ . The resulting purity is shown in fig. 2.4. As before, yellow indicates  $\sigma_{+-}$ , blue  $\sigma_{--}$  and green  $\pi$ -polarization. We can understand the lines of high purity:

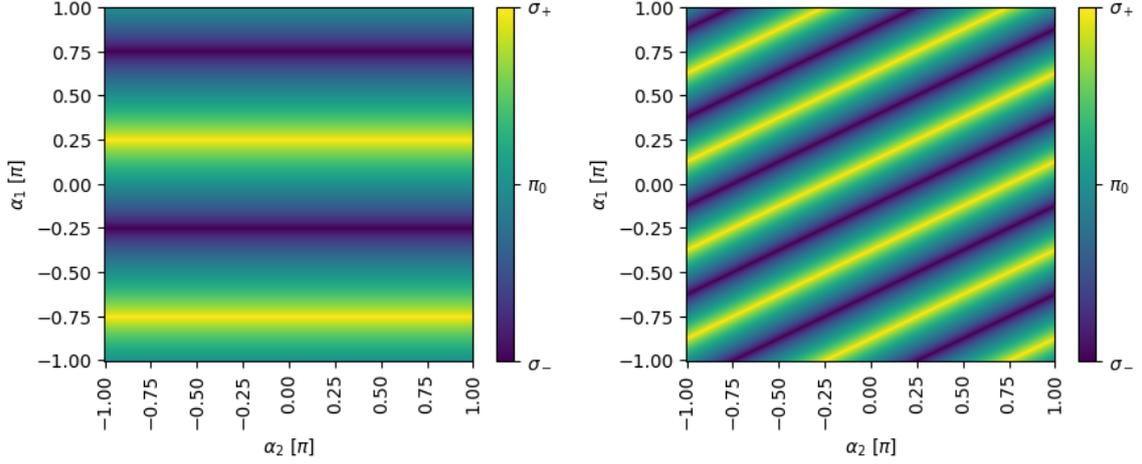


Figure 2.4.: The dependency of purity on the angles  $\alpha_1$  and  $\alpha_2$  of the two samples is shown. On the left side it is  $\Delta\phi_1 = \frac{\pi}{2}$  and  $\Delta\phi_2 = \pi$  (TI), on the right side  $\Delta\phi_1 = \pi$  and  $\Delta\phi_2 = \frac{\pi}{2}$  (IT).

- **TI:**

In this case, the  $\frac{\lambda}{4}$ -wave plate is in the first position. It transforms linear polarization into circular one. The angles of the subsequent  $\frac{\lambda}{2}$ -wave plate can be positioned at an arbitrary angle because a  $\frac{\lambda}{2}$ -wave plate changes the kind ( $\sigma_{\pm}$ ) of circular polarization (see table 2.1) .

- **IT:**

In this configuration, we have a  $\frac{\lambda}{2}$ -wave plate in first position that rotates the linear polarization by  $2\alpha$  (see eq. (2.18)). Therefore, the  $\frac{\lambda}{4}$ -wave plate in second position has to have an angle of  $\frac{\pi}{4}$  to the rotated linear polarization in order to fully transform the entering linear polarization into circular polarization. Hence, we expect high purity for  $\alpha_2 = 2\alpha_1 + \frac{(2n+1)\pi}{4}$ ,  $n \in \mathbb{Z}$ , which results in tilted lines. For all other angles, this setup of wave plates does merely perform incomplete transformation.

# 3. Quantum optical description of the new approach

After investigating the functionality and effects of angles and phase shifts of a standard  $\frac{\lambda}{4}$ - and  $\frac{\lambda}{2}$ -wave plate, in this chapter we will introduce our new approach. We start by looking at nuclear transitions of  $^{57}\text{Fe}$ . Subsequently, we will briefly explain how the intensity increase works. We conclude this exposition by elaborating our new approach for polarization transformation.

## 3.1. Nuclear transitions of $^{57}\text{Fe}$

As we want to use  $^{57}\text{Fe}$  in our setup, we will discuss nuclear resonances on the example of this atom. This iron-isotope is a Mössbauer-active element. These kinds of element have a nearly recoil-free, resonant absorption and emission of  $\gamma$ -rays making energy resolution on a neV-scale possible (see [8]).

In fig. 3.1 the level scheme of  $^{57}\text{Fe}$  can be seen. In absence of a magnetic field, this atom can be regarded as a simple two-level-system with transition frequency  $\omega_0 = 14.4\text{ keV}$  and line width  $\gamma = 4.7\text{ neV}$  (see [5]). Besides, the nuclear spins are  $I_g = \frac{1}{2}$  and  $I_e = \frac{3}{2}$  for the ground and excited state, respectively. Consequently, due to the effect that already a weak magnetic field is able to align the internal magnetic field of about 33 T, magnetic hyper fine splitting can occur. In the case of an internal magnetic field of 33 T, we have a splitting of  $\delta_g = 39.7\gamma$  and  $\delta_e = 22.5\gamma$  for the ground and the excited state respectively. Because of the selection rules for magnetic dipoles, only the transitions with  $m_e - m_g = 0, \pm 1$  can be driven (see [8, chap. 4]). The properties of the six resulting lines are given in table 3.1. These six transitions have different energy differences  $\Delta E$  resulting in different detunings  $\Delta\omega = \omega_0 - \omega_{transition}$ . Furthermore, the six transitions have three different types of polarization. We will focus on the linear transitions ( $n = 2$  and  $n = 5$ ) because it is easier to understand what happens in the system than for the circular ones. They have transition energy difference of

$$\Delta E = \pm 31.05\gamma \approx \pm 146\text{ neV}. \quad (3.1)$$

Hence, it can be seen, that the energy difference in the magnetic hyper fine splitting

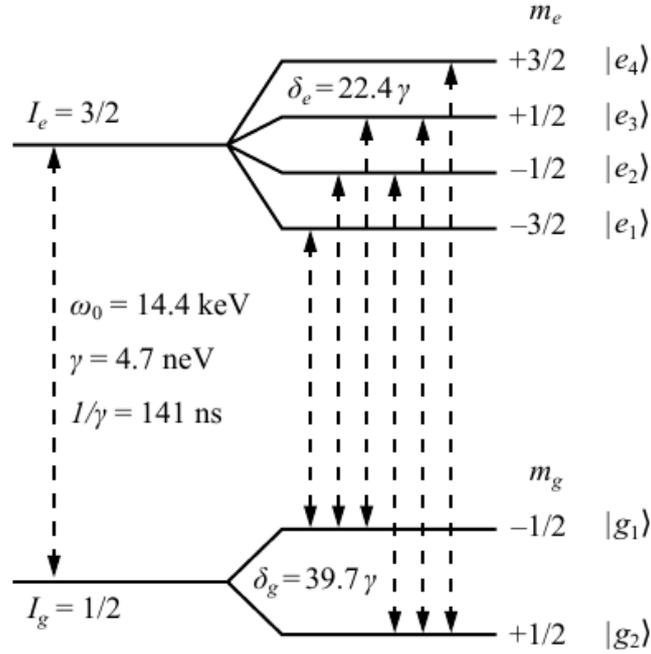


Figure 3.1.: The level scheme of  $^{57}\text{Fe}$  taken from [5, p. 22]. On the left side, the transitions without a magnetic hyper fine splitting can be seen. On the right side, those with the hyper fine splitting are displayed.

(146 neV) is many orders of magnitude smaller than the energy of the transition (14.4 keV). Therefore, we will only be interested in the detuning  $\Delta E$  and not in the absolute value  $E$  from now on.

## 3.2. Phase modulation through moving sample

We will now have a closer look at the phase modulation which was performed in [1, 4]. In these works, a displacement of the sample right after the excitation by a synchrotron pulse gives the possibility to redistribute the energy in frequency space. How this is done, will be explained in this section as well as the effects an arbitrary phase shift causes. Thereby, we only calculate the scalar case without consideration of polarizations in detail. By

$n$	Transition	$\Delta E$	Polarization
1	$ g_1\rangle \leftrightarrow  e_1\rangle$	$-\delta_g/2 - 3/2\delta_e$	$\sigma_-$
2	$ g_1\rangle \leftrightarrow  e_2\rangle$	$-\delta_g/2 - 1/2\delta_e$	$\pi^0$
3	$ g_1\rangle \leftrightarrow  e_3\rangle$	$-\delta_g/2 + 1/2\delta_e$	$\sigma_+$
4	$ g_2\rangle \leftrightarrow  e_2\rangle$	$\delta_g/2 - 1/2\delta_e$	$\sigma_-$
5	$ g_2\rangle \leftrightarrow  e_3\rangle$	$\delta_g/2 + 1/2\delta_e$	$\pi^0$
6	$ g_2\rangle \leftrightarrow  e_4\rangle$	$\delta_g/2 + 3/2\delta_e$	$\sigma_+$

Table 3.1.: Properties of the six transition lines of  $^{57}\text{Fe}$  adapted from [5].

### 3. Quantum optical description of the new approach

using an angle of the magnetic dipole of  $\alpha = \frac{\pi}{4}$ , we can select only the linear lines (see eq. (3.20)). Hence, we focus on these lines and ignore the other lines in this section. The considerations of polarizations will be included in section 3.3.

#### 3.2.1. Calculation of frequency spectra

In the calculations of this subsection, we will follow [4, sec. 2.2] closely.

**Static case.** First, we can describe a synchrotron pulse as a  $\delta(t)$ -pulse in time space, because on the characteristic time scale of the system  $\gamma^{-1} = 141$  ns, these pulses are very short (bunch length of 44 psec at PETRA III, [9]). In frequency space, the spectrum of the scattered light can be calculated by multiplying the initial spectrum  $E_0(\omega)$  by the response function of the scattering sample  $R(\omega)$  (see [10])

$$E'(\omega) = R(\omega) \cdot E_0(\omega). \quad (3.2)$$

The response function for forward scattering of a layer of resonant nuclei is given by (see [10])

$$R(\tau, d) = \delta(\tau) - \sqrt{\frac{b}{\tau}} J_1(2\sqrt{b\tau}) e^{-i\omega_0\tau} e^{-\gamma\tau} \theta(\tau), \quad (3.3)$$

with the Bessel function of first kind  $J_1$ , the frequency  $\omega_0$  and line width  $\gamma$  of the transition, the retarded time  $\tau = t - \frac{z}{c}$  and a material constant  $b = T \frac{\gamma}{2}$ , with the optical thickness  $T = \rho f_R \sigma_0 d$ . It is  $\rho$  the sample density,  $z$  the sample thickness,  $\sigma_0$  the absorption cross section at resonance and  $f_R$  the Lamb-Mössbauer factor.

For  $t < \frac{4}{b}$ , i. e. short times or thin samples, we can use a Taylor expansion

$$\frac{J_1(2\sqrt{bt})}{\sqrt{bt}} \approx 1 - \frac{bt}{2} + \frac{b^2 t^2}{12} + \mathcal{O}(b^3 t^3). \quad (3.4)$$

With these approximations eq. (3.3) becomes

$$R(t) \approx \delta(t) - b e^{(-i\omega_0 t)} e^{-(\gamma + \frac{b}{2})t} \theta(t). \quad (3.5)$$

Transforming this into frequency space and applying eq. (3.2) then gives

$$E'_{\omega_0}(\omega) = E_0 \left( 1 - \frac{ib}{\omega - \omega_0 + i(\gamma + \frac{b}{2})} \right). \quad (3.6)$$

For two transition frequencies  $\omega_0$  and  $\omega'_0$  we can simply add the respective electric fields

$$E(\omega) = E_{\omega_0}(\omega) + E_{\omega'_0}(\omega). \quad (3.7)$$

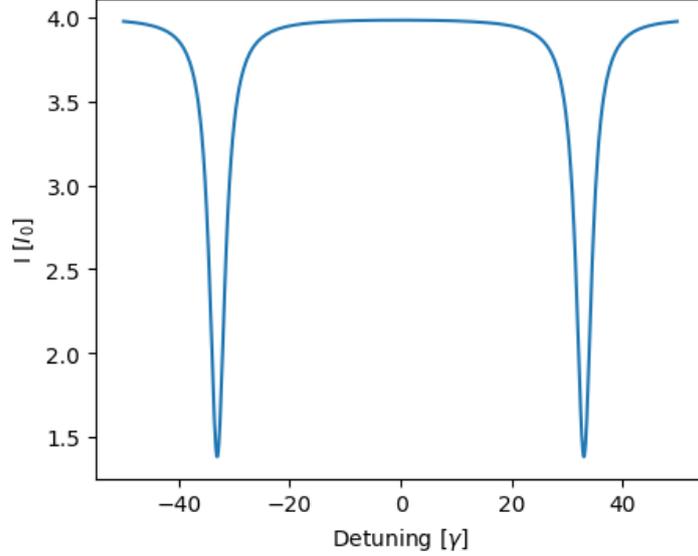


Figure 3.2.: The intensity is plotted against the detuning for the linear transitions according to eq. (3.6). The parameters in this equation are chosen to be  $b = 1.4$ ,  $\omega_0 = -31.1\gamma$  and  $\omega'_0 = 31.1\gamma$ . For better comparison, a horizontal line representing  $I_0$  as well as vertical lines at  $\omega_0$  and  $\omega'_0$  are displayed.

The intensity  $I(\omega) = |E(\omega)|^2$  corresponding to eq. (3.6) is shown in fig. 3.2. Here  $\omega_0$  and  $\omega'_0$  are chosen to be  $\pm 31.1\gamma$ , because this is where the interesting lines occur (see eq. (3.1)). For a  $^{57}\text{Fe}$ -foil  $b = 1.4$  is a realistic value for a sample thickness in the range of  $d = 1\ \mu\text{m}$ . We can clearly see dips in the intensity at  $\omega = \pm 31.1\gamma$ . This makes sense because these are the transition frequencies, where light is absorbed to excite the nucleus.

**Dynamical case.** Now, let us include the sample shift which is electronically controllable. We consider the easiest case (ideal motion), in which the sample does an instantaneous jump

$$\Delta z(t) = \Delta z_0 \cdot \theta(t - t_{shift}), \quad (3.8)$$

where  $t_{shift}$  is the time between the prompt synchrotron pulse and the jump and  $\Delta z_0$  the amplitude of the displacement. Because of  $\Delta\phi = \omega_0 \cdot \Delta t = \omega_0 \cdot \frac{\Delta z}{c}$ , a displacement  $\Delta z$  results in a phase shift of

$$\Delta\phi = \frac{\Delta z}{\lambda} \cdot 2\pi, \quad (3.9)$$

with the wavelength of the transition without hyper fine splitting

$$\lambda = 8.6 \times 10^{-11} \text{ m} = 0.86 \text{ \AA}. \quad (3.10)$$

In principle, one has to employ the wavelength of the actual transition in the splitted system. However, as the energy splitting is many orders smaller than the transition energy in the two-level system,  $\lambda_{splitted} \approx \lambda_{unsplitted}$  for all six transitions. It is important

### 3. Quantum optical description of the new approach

to note that while the phase shift  $\Delta\phi$  is of order  $\pi$ , the time shift  $\Delta t = \frac{\Delta z}{c} \approx 10^{-10}$  ns is negligible, with  $c$  being the speed of light. Hence, we can assume, that the scattered and the unscattered light do not have a time difference. Our motion in eq. (3.8) results in a phase shift of

$$\Delta\phi(t) = \Delta\phi_0 \cdot \theta(t - t_{\text{shift}}), \quad (3.11)$$

with  $\Delta\phi_0 = \frac{\Delta z_0}{\lambda} \cdot 2\pi$  as stated in eq. (3.9). It is important, that the time shift is negligible while the phase shift is of order 1. To consider an arbitrary, phase shift of  $\Delta\phi(t)$  we can just multiply it into the scattering part of the response function in time domain.

$$R_{\Delta\phi_0}(t, z) = \delta(t) - e^{i\Delta\phi_0 \cdot \theta(t - t_{\text{shift}})} \sqrt{\frac{b}{t}} J_1(2\sqrt{bt}) e^{-i\omega_0 t} e^{-\gamma t} \theta(t). \quad (3.12)$$

This is convenient, because of the scattering taking place wherever the sample is in the beam. The motion just gives a relative phase shift in the scattered light as this is moved right after the excitation by the synchrotron pulse. With a constant phase shift of eq. (3.11) the response function eq. (3.12) becomes

$$R_0(t) \approx \delta(t) - e^{i\Delta\phi_0 \cdot \theta(t - t_{\text{shift}})} b e^{-i\omega_0 t} \exp\left[-\left(\gamma + \frac{b}{2}\right)t\right] \theta(t). \quad (3.13)$$

By applying eq. (3.2) the approximation in eq. (3.4) and another Taylor expansion for  $t < \frac{4}{b}$

$$\exp\left(-\frac{b}{2}t\right) \approx 1 - \frac{bt}{2} + \frac{b^2 t^2}{8} + \mathcal{O}(b^3 t^3), \quad (3.14)$$

we get the final result for the scattered electric field

$$\begin{aligned} E_\phi(\omega) &= E_0 \cdot R(\omega, \phi(t)) \\ &= E_0 \left( 1 - \frac{ib}{\omega - \omega_0 + i\left(\gamma + \frac{b}{2}\right)} \left[ 1 + (e^{i\Delta\phi_0} - 1) e^{[\omega - \omega_0 + i\left(\gamma + \frac{b}{2}\right)]it_{\text{shift}}} \right] \right). \end{aligned} \quad (3.15)$$

Also with the included phase shift eq. (3.7) still holds, because  $\omega$  is independent of the phase shift that occurs in time space. Consequently, the electric field is additive and we can investigate the single fields, i. e. the single transitions.

#### 3.2.2. I(ntensity increase) and T(ransformation)

Now, that we know how to include an instantaneous phase shift  $\Delta\phi_0$  into the electric field (see eq. (3.15)), we can see what happens for the phase shift  $\Delta\phi_0 = \pi$  (I) and  $\Delta\phi_0 = \frac{\pi}{2}$  (T). The corresponding intensities  $I' = |E'|^2$  are shown in fig. 3.3. The considerations are still scalar, i. e. without considering the polarization.

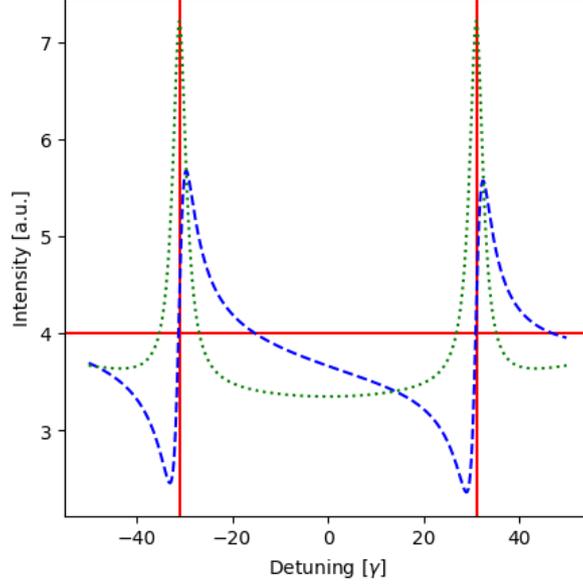


Figure 3.3.: The intensities  $I' = |E'|^2$  corresponding to the electric field in eq. (3.15) for different  $\phi_0$  are shown. The blue (dashed) curve represents  $\phi_0 = \frac{\pi}{2}$  (Fano shape, T) and for the green (dotted) curve  $\phi_0 = \pi$  (Lorentz shape, I) was taken. The parameters are chosen to be  $b = 1.4$ ,  $\omega_0 = -31.1\gamma$ ,  $\omega'_0 = 31.1\gamma$  and  $t_{shift} = \frac{0.1}{2\gamma}$ . For better comparison, a horizontal line representing  $I_0$  as well as vertical lines at  $\omega_0$  and  $\omega'_0$  are displayed.

**Intensity increase.** For  $\Delta z = \frac{\lambda}{2} \Leftrightarrow \Delta\phi = \pi$ , we get a symmetric Lorentz-shaped peak at the former absorption line at  $\omega = \pm 31.1\gamma$ . We see that the intensity is redistributed. In some regions, the intensity is lower than before. However, around  $\omega = \pm 31.1\gamma$  it is much higher.

**Transformation.** For  $\Delta z = \frac{\lambda}{4} \Leftrightarrow \Delta\phi = \frac{\pi}{2}$ , we obtain an asymmetric so-called *Fano line*. It can be clearly seen, that intensity increase is not as high as it is in the Lorentz-shape case. Besides, at  $\omega = \pm 31.1\gamma$ , there is no intensity gain at all, but  $I' = I_0$ . This results in a shift of the maximum relative to the detuning of the linear lines. We will come back to this in section 4.1.3. Furthermore, we see that the slopes of the peaks on the left and right side of  $\Delta = \pm 31.1\gamma$  are different. The decrease is faster in the direction towards  $\Delta = 0$ . Additionally we see, that on the right side of  $\Delta = \pm 31.1\gamma$  there is a maximum, while on the left side in the same distance there is a minimum.

### 3.2.3. Combination of two samples.

As explained before (see chapter 1), we will combine two samples in our new setup. Therefore, we want to formulate the combination of two samples. In frequency space, we

### 3. Quantum optical description of the new approach

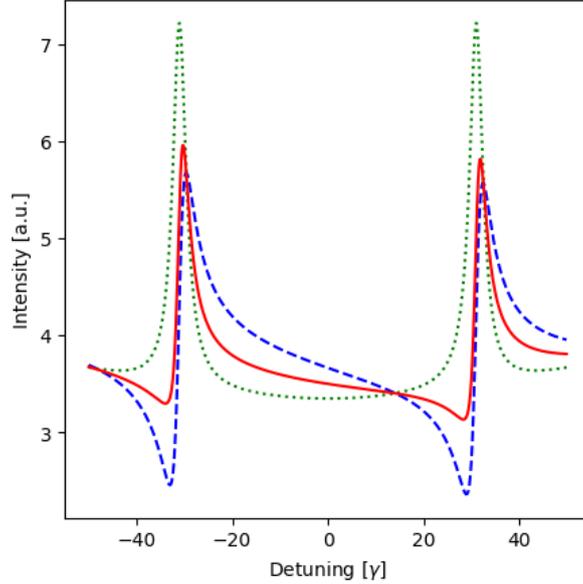


Figure 3.4.: The resulting intensities  $I' = |E'|^2$  of eq. (3.15) are shown for  $\Delta\phi = \pi$  (green, dotted line) and  $\Delta\phi' = \frac{\pi}{2}$  (blue, dashed line) as well as the intensity of their combination according to eq. (3.16) (red, solid line). The parameters are chosen to be  $b = 1.4$ ,  $\omega_0 = -31.1\gamma$ ,  $\omega'_0 = 31.1\gamma$ ,  $\gamma = 1$  and  $t_{shift} = \frac{0.1}{2\gamma}$

can just multiply the electric fields of both samples.

$$E(\omega) = E_\phi(\omega) \cdot E_{\phi'}(\omega). \quad (3.16)$$

If two different transitions shall be considered as in eq. (3.7), this must be done on the stage of the single samples, i. e. before combining two samples by multiplication as in eq. (3.16). Because we did not consider any angles  $\alpha_i$  at the moment and eq. (3.16) being commutative, the two configurations IT and TI are equivalent. The intensity of the resulting electric field for  $\Delta\phi = \pi$  and  $\Delta\phi' = \frac{\pi}{2}$  or vice versa can be seen in fig. 3.4. Due to the shift of the maximum for the Fano line, also the product of Fano and Lorentz-line has a shifted maximum relative to  $\omega = 31.1\gamma$ . In addition, the intensity increase is just a little bit higher, than for a single Fano line. Furthermore, the minima are weaker, the peaks are thinner and the whole spectrum is less asymmetric than it was before.

#### 3.2.4. Time spectra

After looking at the system in frequency space, we can now have a short look at the spectrum in the time domain. We apply a Fourier transform on eq. (3.15). In fig. 3.5, the time spectrum is shown for two different sample widths. An overall decrease of intensity can be clearly observed. This decrease is expected because of the decay of the excited states and implemented in eq. (3.12) as the term  $\exp(-\gamma t)$ . Besides, we see a periodic structure with a repeating time of about 12 ns, due to the term  $\exp(-i\omega_0 t)$  in eq. (3.12).

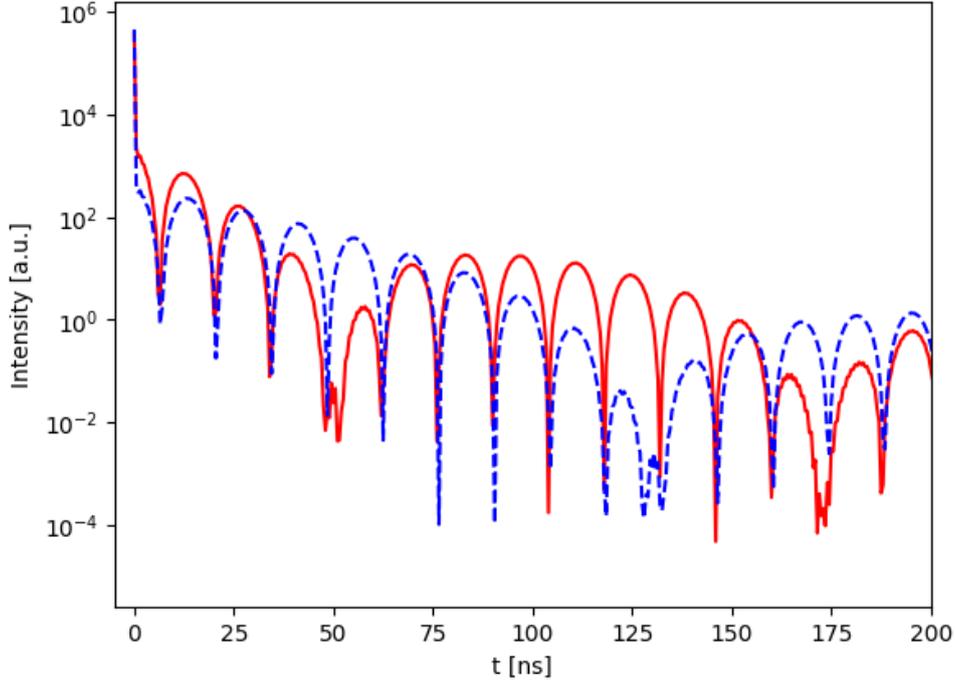


Figure 3.5.: A typical time spectrum can be seen. Besides the decreasing intensity, the dynamical beats can be seen. The blue (dashed) line is for a sample thickness of  $d = 2 \mu\text{m}$ , the red (solid) one for  $d = 5 \mu\text{m}$ .

These are the so-called *quantum beats*. The dips at  $t \approx 50 \text{ ns}$  and  $t \approx 175 \text{ ns}$  in the red curve are the so called *dynamical beats*. They are caused by the roots of the Bessel function  $J_1$  in eq. (3.12) and depend on the sample thickness  $d$  since it is  $b \sim d$ . The thinner the thickness, the later the first quantum beat.

### 3.3. New approach for polarization conversion

In this section we will now introduce our new quantum optical approach in detail. We will combine two samples that have a similar function as the two wave plates in section 2.3. One of these samples is supposed to transform the polarization, the other one is supposed to increase the intensity. Because of  $^{57}\text{Fe}$  having a magnetic transition M1, we will from now on use the angle of the magnetic dipole instead of the electric one. The two descriptions are equivalent if we keep in mind that there is an angle of  $\frac{\pi}{2}$  between the angle of the magnetic dipole  $\alpha_M$  and the electric one  $\alpha_E$

$$\alpha_M = \alpha_E + \frac{\pi}{2}. \quad (3.17)$$

As said in chapter 1, we focus on the conversion of linearly polarized light into circularly polarized one. In principle, the setup that will be discussed in this section can also be used to convert polarizations the other way around. For simplicity, we will calculate only

### 3. Quantum optical description of the new approach

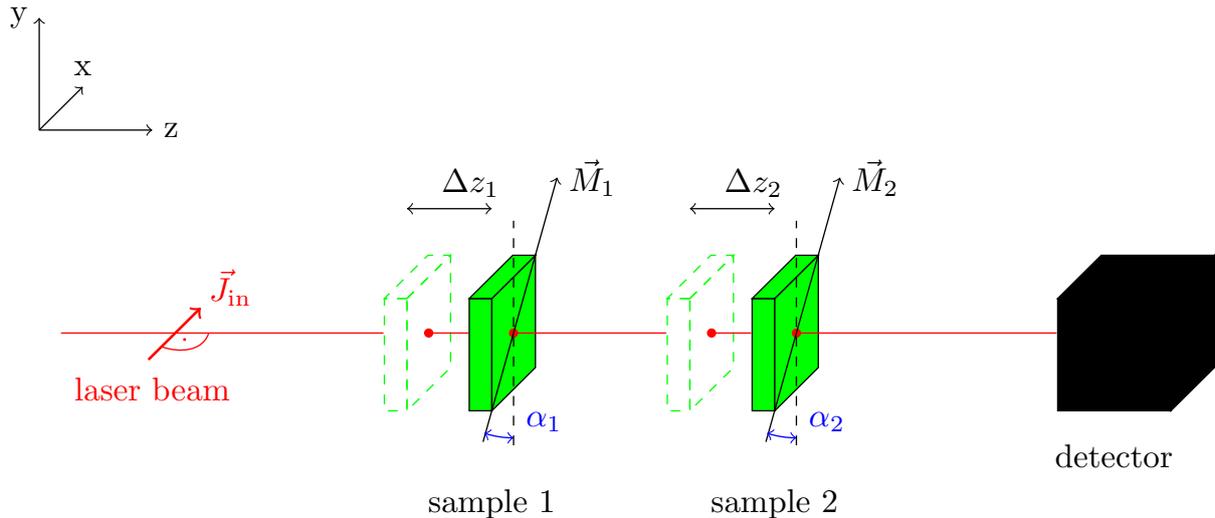


Figure 3.6.: This is the setup of our quantum optical approach for polarization conversion. A beam in  $z$ -direction with linear polarization in  $x$ -direction enters the setup (red). It passes through two very thin ( $d = 1 \mu\text{m}$ ) samples (green). Mounting them on piezos allows a displacement of  $\Delta z_i$  of order  $\lambda$ . In the end, we detect the intensity for a given polarization at a detector (black). At the samples there is a magnetic field applied to arrange the internal electric field. The angle of magnetization  $\vec{M}_i$  in the  $xy$ -plane with the  $y$ -axis is the magnetic angle  $\alpha_i$  (blue). For simplicity, the magnets, that are used to arrange the internal magnetic field, are not displayed.

the special case of linear into circular polarization.

#### 3.3.1. Setup

Our proposed setup is shown in fig. 3.6. It consists of two  $^{57}\text{Fe}$  foils, that are mounted onto a piezo element. That way, the samples can be displaced by fractions of the resonant wavelength  $\lambda$  of the magnetic transition discussed in section 3.1. As discussed before (see eq. (3.9)), this results in a phase shift of order  $\pi$ . Both samples are placed in a magnet to align the internal magnetic field. In principle, it would also be possible to expose the samples to a magnetic field before mounting them in order to orientate the internal field, but then the magnetization could be easily changed by weak fields. Therefore, it is better to install magnets permanently in order to make the whole system more stable. The angles of the magnetic dipole moment with the  $x$ -axis are  $\alpha_{M,i}$  and will be just called  $\alpha_i$  if it is clear that the magnetic angle is meant. We assume linearly polarized light ( $\pi_0$ ) entering the setup in  $z$ -direction. This then scatters with the first sample and afterwards with the second one. For the analysis in this thesis, a detector is placed after the second sample. It is able to detect the intensities  $I_{\pm}$  for the circular polarizations  $\sigma_{\pm}$ .

### 3.3.2. Functionality

Now, we take all lines in table 3.1 into account again. It is clear that for discussing the transformation of polarizations, we cannot ignore them anymore. Therefore, we need to expand our scalar model developed in section 3.2.

Because we have to consider a magnetic transition, we cannot just consider the polarizations in the  $xy$ -plane perpendicular to the beam propagation, but have to consider all three dimensions. If the magnetization is perpendicular to the beam polarization along the  $y$ -axis, i. e.  $\alpha_M = \frac{\pi}{2}$ , we can write the polarizations in the following way (see [8, chap.4])

$$\vec{J}_0^{\text{M},0} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{J}_{0,\perp}^{\text{M},0} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \vec{J}_{\pm}^{\text{M},0} = \begin{pmatrix} 0 \\ 1 \\ \pm i \end{pmatrix}. \quad (3.18)$$

In our setup, it is  $\alpha_M = \frac{\pi}{4}$ . Thus we need to rotate the polarizations by an angle of  $\frac{\pi}{4}$  around the  $z$ -axis and find

$$\vec{J}_0^{\text{M}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad \vec{J}_{0,\perp}^{\text{M}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \vec{J}_{\pm}^{\text{M}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ \pm i \end{pmatrix}. \quad (3.19)$$

The coupling is then given by (see [5, chap. 3])

$$g \sim \vec{J}_{\text{transition}}^{\text{M}} \cdot \vec{J}_{\text{in}}^{\text{M}}, \quad (3.20)$$

where  $\vec{J}_{\text{transition}}^{\text{M}}$  is the polarization of the transition and  $\vec{J}_{\text{in}}^{\text{M}}$  the one of the incoming light. In our setup it is  $\vec{J}_{\text{in}}^{\text{M}} = (1, 0, 0)^T$ .

We can implement the different coupling by multiplying the respective electric field by the coupling constant

$$E'(\omega) = \sum_{i=1}^6 g_i \cdot E_{\Delta\phi, \omega'_i}(\omega). \quad (3.21)$$

Because of the electric field being additive (see eq. (3.7)), we can look at each transitions separately.

**One sample.** Let us focus on the linear transitions again. Similarly to section 2.2, we can decompose the incoming polarization

$$\vec{J}_{\text{in}} = \vec{J}_{\parallel} + \vec{J}_{\perp}, \quad (3.22)$$

into one vector  $\vec{J}_{\parallel}$  parallel to the magnetic polarization vector of the linear transition giving  $g_{\parallel} \sim 1$  and  $\vec{J}_{\perp}$  perpendicular to that giving  $g_{\perp} = 0$  (see eq. (3.20)). As the coupling for  $\vec{J}_{\perp}$  is  $g_{\perp} = 0$ , this component is not affected by the sample as no interaction takes place. Therefore, the phase shift caused by the motion of the piezo is only applied to  $\vec{J}_{\parallel}$ . This vector is along the magnetic dipole moment.

Using the basis of  $\vec{J}_{\parallel}$  and  $\vec{J}_{\perp}$  in the  $xy$ -plane instead of the eigenbasis of the wave plate

### 3. Quantum optical description of the new approach

Polarization of transition:	$\pi_0$	$\pi_{0,\perp}$	$\sigma_+$	$\sigma_-$
Polarization after $\frac{\lambda}{4}$ -config:	$\sigma_-$	$\sigma_+$	$\sigma_+$	$\sigma_+$
Polarization after $\frac{\lambda}{2}$ -config:	$\pi_{0,\perp}$	$\pi_0$	$\pi_0$	$\pi_0$

Table 3.2.: The effect of our setup on  $\pi_0$ -polarized light interacting with the different transitions is shown for  $\alpha = \frac{\pi}{4}$ .

and converting the angles of electric respectively magnetic dipole  $\alpha_E$  and  $\alpha_M$  according to eq. (3.17), we can apply eq. (2.13)-(2.18) to our setup. That way, we can convert polarizations into each other in a similar way as a  $\frac{\lambda}{4}$ -wave plate does. If we choose  $\alpha_M = \frac{\pi}{4}$ , we are able to transform linearly polarized light into circularly polarized one as discussed in section 2.2. Hence, we would expect that our setup has a similar angular and displacement dependency as a  $\frac{\lambda}{4}$ -wave plate. In particular, we find the following transformations:

$$\begin{aligned} \alpha = \frac{\pi}{4}, \Delta_z = \frac{\lambda}{4} : \pi_0 \rightarrow \sigma_+ \text{ and } \pi_{0,\perp} \rightarrow \sigma_-, \\ \alpha = \frac{\pi}{4}, \Delta_z = \frac{\lambda}{2} : \pi_0 \rightarrow \pi_{0,\perp} \text{ and } \pi_{0,\perp} \rightarrow \pi_0. \end{aligned} \quad (3.23)$$

As it can be seen from eq. (3.18), the circular polarizations can be written as a superposition

$$\vec{J}_{\pm}^M = \vec{J}_{0,\perp}^M \pm i \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad (3.24)$$

of linear polarization perpendicular to the one of our incoming light and another vector, that will not couple to linearly polarized incoming light for any angle  $\alpha_M$  as it is

$$(a \ b \ 0) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0, \quad \forall a, b \in \mathbb{R}. \quad (3.25)$$

For this reason, the circular lines behave as if they would have a  $\pi_{0,\perp}$ -polarization. Hence, the argumentation for  $\pi_0$  polarization from above can be applied. That way we get

$$\begin{aligned} \alpha = \frac{\pi}{4}, \Delta_z = \frac{\lambda}{4} : \sigma_{\pm} \rightarrow \sigma_-, \\ \alpha = \frac{\pi}{4}, \Delta_z = \frac{\lambda}{2} : \sigma_{\pm} \rightarrow \pi_0. \end{aligned} \quad (3.26)$$

A summary of the effect of our setup on linearly polarized light ( $\pi_0$ ) interacting with the different transitions is shown in table 3.2. The circular polarizations behave exactly like  $\pi_{0,\perp}$ -polarizations.

**Two samples.** As we have seen in eq. (3.16) to combine two sample, we just need to multiply their electric fields. Choosing the decomposition in eq. (3.22), the phase shift is only in one component of the polarization vector  $\vec{J}$ . Therefore, the single components can

### 3.3. *New approach for polarization conversion*

still be described by the scalar case. Hence, we can combine two samples by multiplying the phase shift matrices for the respective angles  $\alpha_i$  (eq. (2.14)). That way, our setup can be described by the equations of section 2.3. Hence, our setup will in principle behave similarly to a system of two standard wave plates.

## 4. Results

After introducing our new approach in section 3.3, we now want to further discuss the angular and displacement dependency as well as what happens for different sample thicknesses. Besides, we will look at other applications of our setup. To do so, we will calculate the spectra in frequency and time space. The equations in chapter 3 are just for illustrating the principle function of our setup. To discuss our approach we will use the python library `pynuss`. It is based on `conuss` ([11]) and works with the layer-formalism (see [8]). That way, multiple other effect will be included into the numerical calculations that where not in section 3.3.

Until stated otherwise, the parameters in table 4.1 are used: Basically, if just circularly polarized light is wanted, any of the six transition frequencies could be taken. Because of linear lines being a bit more intuitive as they do not behave as another polarization, we stay with these lines for the analysis of our setup.

As we have seen in chapter 3, in frequency space the calculations are mostly multiplications. According to the convolution theorem, a multiplication in frequency space corresponds to a convolution in time space, which is the respective Fourier space. Hence, calculations in frequency space will be much more intuitively understandable than in time space. Therefore, most of the analysis of our setup is done in frequency space.

### 4.1. One sample and general effects

In this section, the spectra for only one sample are calculated in order to investigate only the transformation from linear into circular polarization without the intensity increase. This makes the system easier for the beginning. Besides, some effects that occur for one and more samples are discussed.

#### 4.1.1. Frequency spectrum of one sample

Let us start by investigating the spectrum of a single sample. As we have seen before (see eq. (3.23)), we need a phase shift of  $\frac{\pi}{2}$  to convert linearly polarized into circularly polarized light. To better see, how the spectrum for a moving sample, in fig. 4.1 the spectrum of only one sample in the static case is shown for  $\alpha = \frac{\pi}{4}$  (code: appendix A.1.1). The absorption lines at the transition frequencies can be clearly seen. As we have seen in

Parameter	Value
Beam direction	$(0, 0, 1)^T$
Beam polarization	$\vec{J} = (1, 0, 0)^T$
Motion	ideal as in eq. (3.8)
Thickness of the samples	$d = 1 \mu\text{m}$
Internal magnetic field	$B = 33 \text{ T}$
Resonant wavelength	$\lambda = 8.6 \times 10^{-10} \text{ m}$
Detuning of the linear lines	$\Delta = \pm 31.1 \gamma$
Magnetic angle (sample 1)	$\alpha_1 = \frac{\pi}{4}$
Magnetic angle (sample 2)	$\alpha_2 = -\frac{\pi}{4}$
Displacement (sample 1)	$\Delta z_1 = 0.25\lambda$
Displacement (sample 2)	$\Delta z_2 = 0.5\lambda$

Table 4.1.: Standard parameters for the calculation of spectra.

section 3.3, the circular transitions behave as if they would have a polarization of  $\pi_{0,\pm}$ . Therefore, the purity is  $P = 0$  for all lines in the static case. Because of the intensity after a  $\sigma_{\pm}$ -polarization filter  $I_{\pm}$  satisfies  $I_{\pm} = \frac{I_0}{2}$  for linear polarized light, the spectra for  $I_{\pm}$  would look the same as the total spectrum  $I$  scaled by a factor of  $\frac{1}{2}$ .

In fig. 4.2 the spectrum of the dynamical case

$$\frac{\lambda}{4}\text{-configuration: } \alpha = \frac{\pi}{4} \text{ and } \Delta z = \frac{\lambda}{4} \quad (4.1)$$

can be seen. This configuration converts  $\pi_0$ -polarization into  $\sigma_+$  and  $\sigma_{\pm}$  into  $\sigma_-$  (see eq. (3.23)). Instead of absorption dips as in fig. 4.1, the spectrum has even local maxima at detuning  $= \pm 31.1\gamma$  for the  $\sigma_+$ -detector. For linear polarized light, a  $\sigma_{\pm}$ -detector would measure  $I = 0.5I_0$ . In the figure, the intensity is way higher than  $I = 0.5I_0$ . That shows, that the linear polarized light is in fact transformed into circularly polarized light. For the purities, we observe that the original purity (static case)  $P = 0$ , becomes  $P \approx 1$  after our

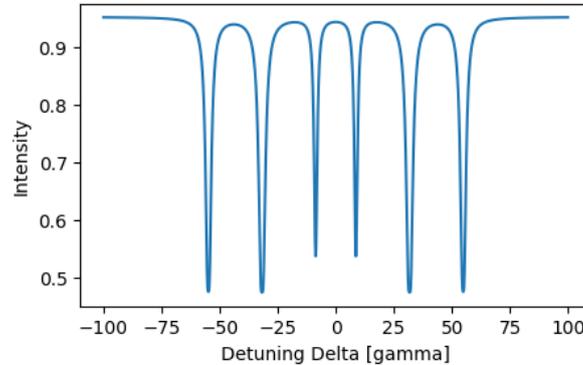


Figure 4.1.: The spectrum for the parameters in table 4.1, but with one unmoved sample is shown. The magnetic field is rotated by  $\frac{\pi}{4}$  against the polarization of the beam.

## 4. Results

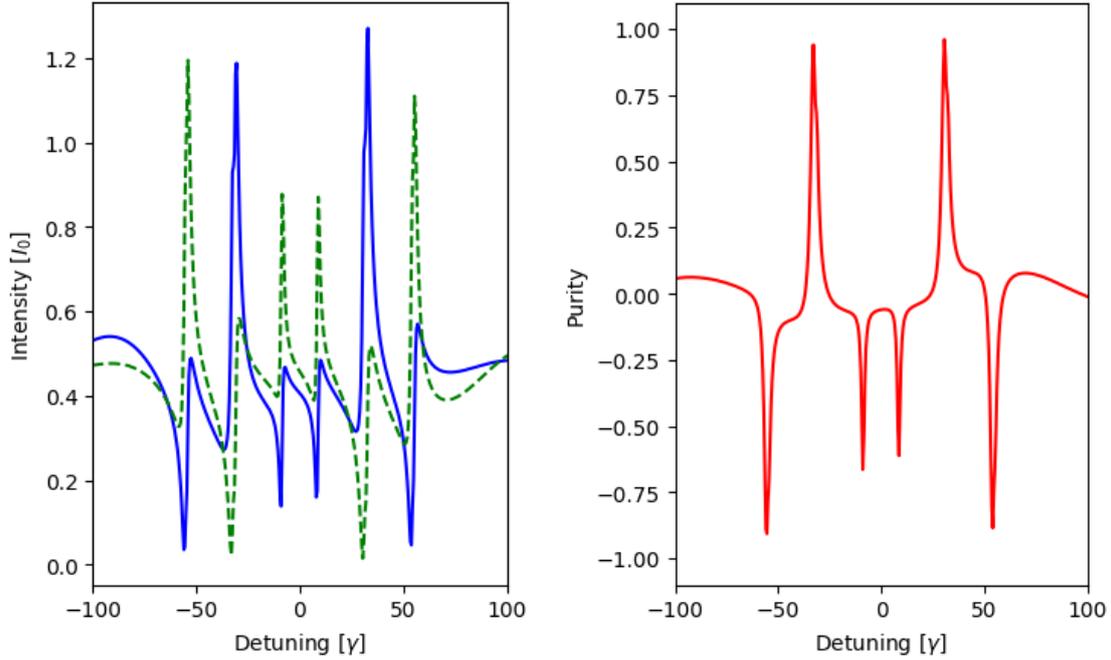


Figure 4.2.: The spectrum for parameters in table 4.1, but with only one sample is displayed. The intensities (left) and purity (right) for the  $\frac{\lambda}{4}$ -configuration are shown. The blue (solid) line shows the spectrum after a  $\sigma_+$ -polarization filter, the green (dotted) line after a  $\sigma_-$  one.

setup for a detuning of  $\pm 31.1\gamma$ . For all the circular transitions, we get  $\sigma_-$  polarization. Despite a little loss in the purity this is the result we expected. Later on (see section 4.3) we will try to reduce that loss. If we ignore all  $\sigma_{\pm}$  polarized lines for a moment, the spectrum looks pretty much similar to fig. 3.4. It can be clearly seen, that the proposed setup indeed changes linearly to circularly polarized light at a detuning of  $31.1\gamma$ . We also already have an intensity increase at this special frequency. Nevertheless, we will increase this even more with a second sample later on (see section 4.2).

### 4.1.2. Real vs. ideal motion

Until now we have calculated everything using an ideal motion (see eq. (3.8)). A realistic piezo would never move as prescribed by a step function because it works as a capacity. A more realistic motion using the error function erf would be

$$\phi(t) = \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{t - t_{\text{shift}}}{t_{\text{rise}}} \right) \right] \quad (4.2)$$

where  $t_{\text{rise}}$  is the rise time and  $t_{\text{shift}}$  is the time between the motion and the excitation by a synchrotron pulse. An ideal motion as well as a more realistic one are shown in fig. 4.3 for two different rise times  $t_{\text{rise}}$ . The height of the phase shift can be scaled to get arbitrary shifts.

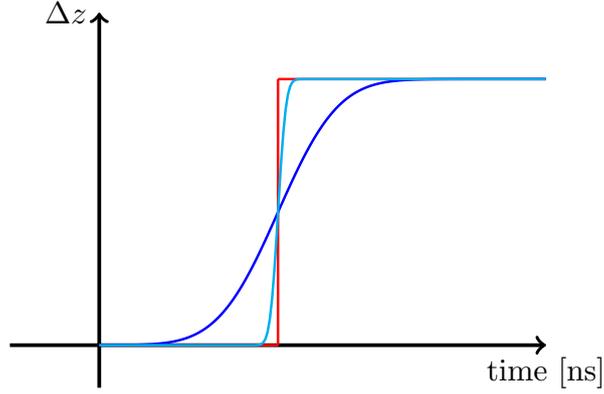


Figure 4.3.: Three different motions are plotted. The red one is an ideal motion, the blue ( $t_{\text{rise}} = 15$  ns) and cyan ( $t_{\text{rise}} = 2$  ns) are more realistic motions as shown in eq. (4.2). It is  $t_{\text{shift}} = 10$  ns.

A rise time of  $t_{\text{rise}} = 15$  ns is possible in experiments (see [1]), but it should be possible to decrease the rise time with faster reacting piezos. Then the step-function would become an even better approximation.

In fig. 4.4 the spectra are plotted for the motions in fig. 4.3 (code: appendix A.1.1). As one can see for  $t_{\text{rise}} = 2$  ns (dashed line) there is not really a difference between ideal and more realistic motion at all. For  $t_{\text{rise}} = 15$  ns the approximation with an ideal motion is quite good around the transition frequencies. As we are particularly interested in these region, we will stick to the ideal motion to make thing easier.

### 4.1.3. Shift of intensity of maximum

We detect a shift of up to  $\pm 0.5\gamma$  between the detuning with maximal intensity and the detuning of the linear transitions for the Fano lines (code: appendix A.1.1). We observed an even bigger shift for two samples. The shift is caused by the asymmetric Fano lines, which do not have a maximum at  $\Delta = 31.1\gamma$  (see section 3.2.2). Hence, we will also detect this shift for two samples because multiplying a Fano line by a Lorentz line leads to a shifted maximum. Despite the shift being small, it has a big impact on the calculations due to the sharp lines. When we want to compare different sets of parameters to each other, we need a scalar value for intensity and purity, that represent the spectrum in the interesting detuning range. The first idea would be to just take the intensity and purity at  $\Delta = 31.1\gamma$ . But due to the shift, this will not be the maximum of intensity and purity for every set of parameters. Instead, there are a number reasonable possibilities to define the detuning, at which we evaluate intensity and purity. Among others, there are the following approaches:

- **Theoretical value**

Here, we take the theoretical value of the linear line  $\Delta = 31.1\gamma$  and evaluate intensity

## 4. Results

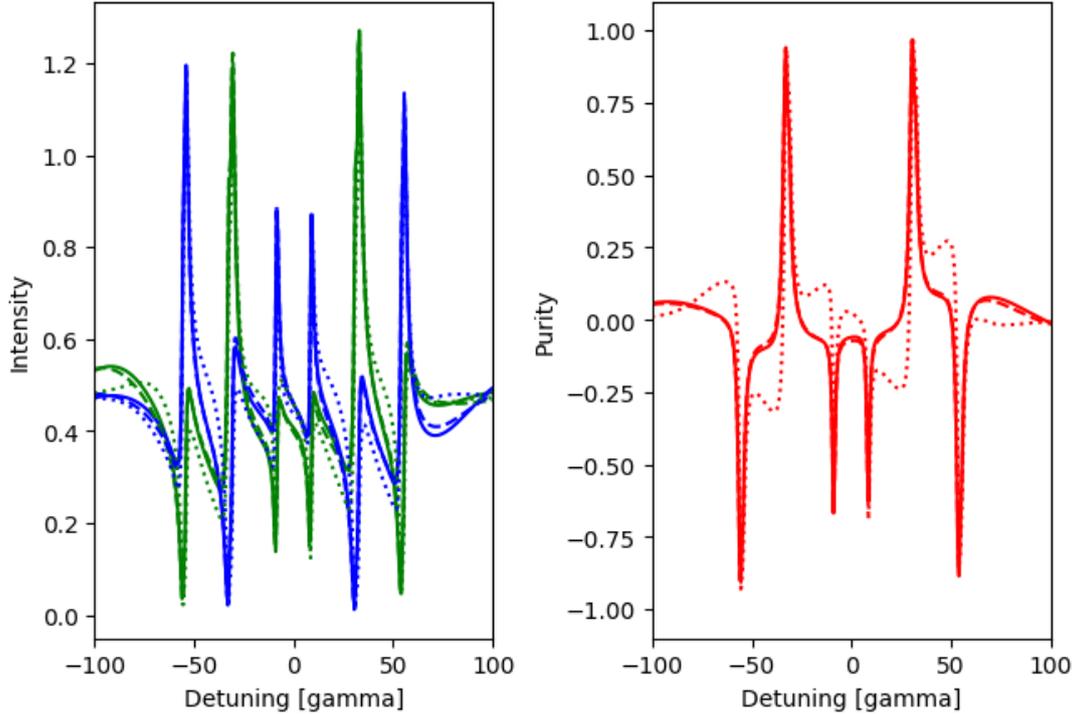


Figure 4.4.: The spectra for the parameters in table 4.1, but with only one sample and different types of motion are shown. The solid line is of the ideal motion, whereas the dotted ( $t_{rise} = 15$  ns) and dashed ( $t_{rise} = 2$  ns) lines are of the more realistic motion in eq. (4.2).

and purity there. Because of the shift, this does not give the best intensity and purity we could have in the neighborhood of the theoretical value. In an experiment the frequency would also be set at a given frequency.

- **Maximum of intensity**

In this approach, we take the detuning, where the intensity has a maximum in the neighborhood of the theoretical value, for the detuning, on which intensity and purity are evaluated. This causes problems for the Fano lines. They occur at the  $\frac{\lambda}{4}$ -configuration and do not have a maximum at the interesting linear frequency, but a root. Despite that, this measure would be better than the *Theoretical value* to compare the numerical results in this chapter with the analytical ones in chapter 2, because here the shift of the maximum does not play a role.

- **Fitting**

In this case, we would fit the square of the electric field in eq. (3.15) for two samples to the spectrum. This would give back the theoretical value of  $\Delta = \pm 31.1 \gamma$ . But this would need much more effort, whereas the precision is limited to the step width (here step width  $0.5 \gamma$ ) nevertheless.

- **Maximum of |purity|**

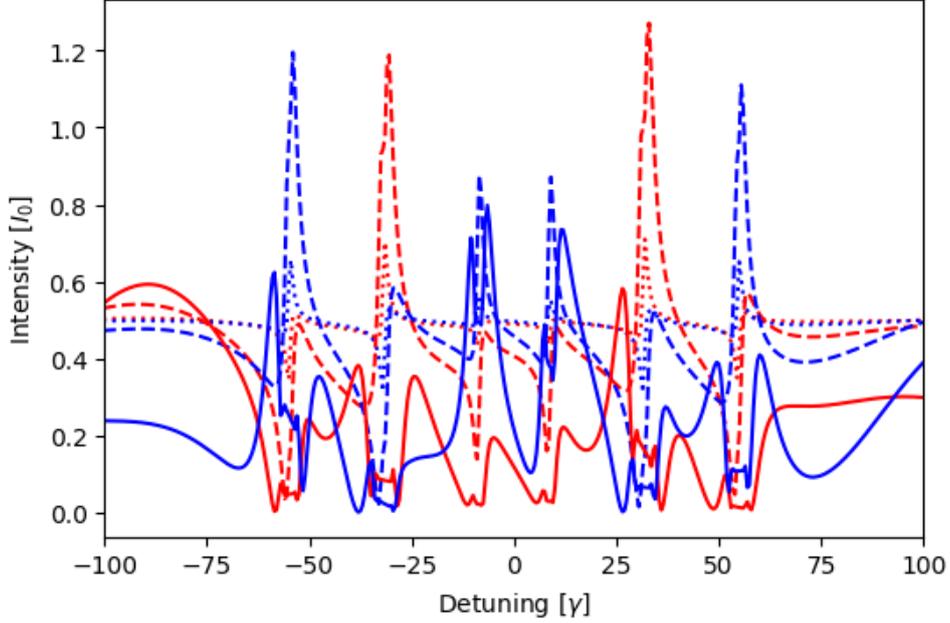


Figure 4.5.: The spectra for the parameters in table 4.1, but with only one sample and different thicknesses are shown. The blue lines indicate  $I_+$ , the red ones  $I_-$ . The dotted lines represent a sample thickness of  $d = 0.1 \mu\text{m}$ , the dashed lines of  $d = 1 \mu\text{m}$  and the solid lines of  $d = 10 \mu\text{m}$ .

This is similar to *Maximum of intensity*, despite searching for the maximum of the absolute value of purity instead of intensity. When scanning over different angles  $\alpha$  and phase shifts  $\Delta\phi$ , we will have situation, in which the lines are converted into circular lines ( $P \approx \pm 1$ ), and situations, in which they are converted into linear lines ( $P = 0$ ). In the first case, we are indeed searching for a maximum, but in the second we would expect a root. Hence, this method works for the interesting transition, but gives wrong results for some other parameter sets.

In the end, we decided to use the *Theoretical value* because in most applications the wavelength, and with that the detuning, is defined. For our analysis, we keep in mind, that for this choice we will not always get the highest intensity or purity. The actual results can be even better.

#### 4.1.4. Influence of thickness of targets $d$

We will now investigate the effect of the thickness of the foils (code: appendix A.1.1). The dynamical spectra for a sample thickness of  $d = 0.1 \mu\text{m}$ ,  $d = 1 \mu\text{m}$  and  $d = 10 \mu\text{m}$  are shown in fig. 4.5. It can be clearly seen, that a thickness of  $d = 1 \mu\text{m}$  gives the highest intensities. As this also has clear peaks and is accessible in experiments, our choice of taking  $d = 1 \mu\text{m}$  is appropriate. For a thinner sample the intensity is much smaller because less photons scatter. For a thicker sample the spectrum looks quite different. If

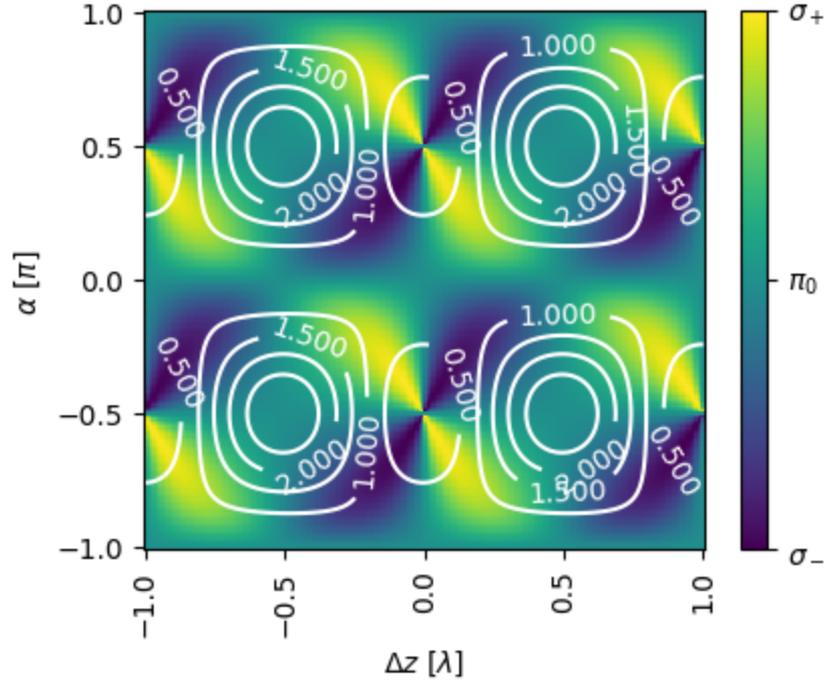


Figure 4.6.: Purity (colors) and intensity (contour lines) are shown for one sample with different magnetic angles and displacements. All other parameters are like in table 4.1.

the sample is this thick, the dynamical beat comes to early and the intensity increase does not work properly, resulting in dips in the middle of the peaks.

#### 4.1.5. Influence of magnetic angles $\alpha$ and displacement $\Delta z$

Before starting the investigation of two samples, we want to investigate the influence of the magnetization angle  $\alpha$  and the displacement  $\Delta z$  for one sample. From a theoretical point of view (see chapter 2) we would expect an optimal configuration for the  $\frac{\lambda}{4}$ -configuration (eq. (4.1)). To investigate that behavior, we first calculate the spectrum for different angles and displacements. We then take the intensity and purity at  $\Delta = -31.1 \gamma$  (code: appendix A.1.1). The result is shown in fig. 4.6. As in the pictures in chapter 2, the purity is color coded. Yellow indicates pure  $\sigma_{+-}$ , blue pure  $\sigma_{--}$  and green  $\pi$ -polarization. The intensity is shown as white contour lines. Comparing the analytical expectation (see fig. 2.2) to the numerical results, we see that in fact we have a high purity at the positions with  $\alpha = \frac{\pi}{4} + n\frac{\pi}{2}$  and  $\Delta z = \frac{\lambda}{4} + m\frac{\lambda}{2}$  with  $m, n \in \mathbf{Z}$ . But there is a high tendency of the purity smearing out towards  $\alpha = \pm\frac{\pi}{2}$  and  $\Delta z = -1, 0, 1$ . Thus far, we do not understand the origin of this effect. However, it allows to employ this setup as a high precision sensor. This it will be discussed in section 4.5.3. Furthermore, as we would expect with a look back at fig. 3.3 we find the regions with high intensity for  $\Delta z = \pm 0.5$  So already here we see that the regions with high purity are not the same as the one with high intensity.

Experimental applications thus have to pick between a high intensity or high purity.

## 4.2. Two samples

After investigating the simple case with just one sample, we will now go one step further and include a second sample to increase the intensity even more as discussed in section 3.2. We will then investigate the effects of different parameters.

### 4.2.1. Frequency spectrum of two samples

To combine the two samples there are two possibilities as discussed before (see eq. (2.20)): First  $\frac{\lambda}{2}$  (I), second  $\frac{\lambda}{4}$  (T) or the other way around:

$$\begin{aligned} \text{Config IT: } \Delta z_1 &= \frac{\lambda}{2}, \Delta z_2 = \frac{\lambda}{4}, \alpha_1 = \frac{\pi}{4} \text{ and } \alpha_2 = -\frac{\pi}{4}, \\ \text{Config TI: } \Delta z_1 &= \frac{\lambda}{4}, \Delta z_2 = \frac{\lambda}{2}, \alpha_1 = \frac{\pi}{4} \text{ and } \alpha_2 = -\frac{\pi}{4}. \end{aligned} \tag{4.3}$$

These angles have been chosen because they are admissible for both of the above configurations as it can be seen in chapter 2)

A spectrum is shown in fig. 4.7 for both configurations (code: appendix A.1.1). As for one sample, we can see, that the conversion into circular polarization has taken place at all transitions in both cases. This can be seen from the intensities  $I_+$  and  $I_-$ , that have minima or maxima at the transition frequencies depending on the output polarization. Besides, it can be deduced from the purity that is  $P \approx \pm 0.75 \neq 0$  at the respective frequencies. But we also see, that we do not get pure circularly polarized light because  $P \neq \pm 1$ . For one sample the purity was higher, but the intensity was lower (see fig. 4.2). We will optimize purity and intensity in section 4.3. Besides, if we compare the spectra to the ones of one sample, we see that for two samples, the spectra are more symmetric. This is expected because in this case a Fano line is multiplied with a Lorentz line resulting in a more symmetric spectrum (see fig. 3.4).

When comparing the two configurations IT and TI to each other, it can be clearly seen, that they have opposite purity. This is caused by our choice of  $\alpha_1 = \frac{\pi}{4} = -\alpha_2$ . For the intensity increase (I) this makes no difference because of  $\vec{J} \equiv -\vec{J}$  (see eq. (2.3)). The intensity increase leads to a rotation of  $\vec{J}$  by  $\frac{\pi}{2}$  respectively  $-\frac{\pi}{2}$  leading to  $\vec{J}$  respectively  $-\vec{J}$ . For the transformation (T) our choice of angles has a big effect. As it is  $\cos\left(\frac{\pi}{4}\right) = \cos\left(-\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) = -\sin\left(-\frac{\pi}{4}\right)$ , plugging that into eq. (2.14) results in an additional minus-sign in the second component of the polarization vector  $\vec{J}$ . Hence, the purity of circularly polarized light is changed. Because of the transformation taking place with one of the two angles, the handedness is different for the two configurations. Despite the different polarization, we also find that the intensities are higher for the (IT) configuration.

#### 4. Results

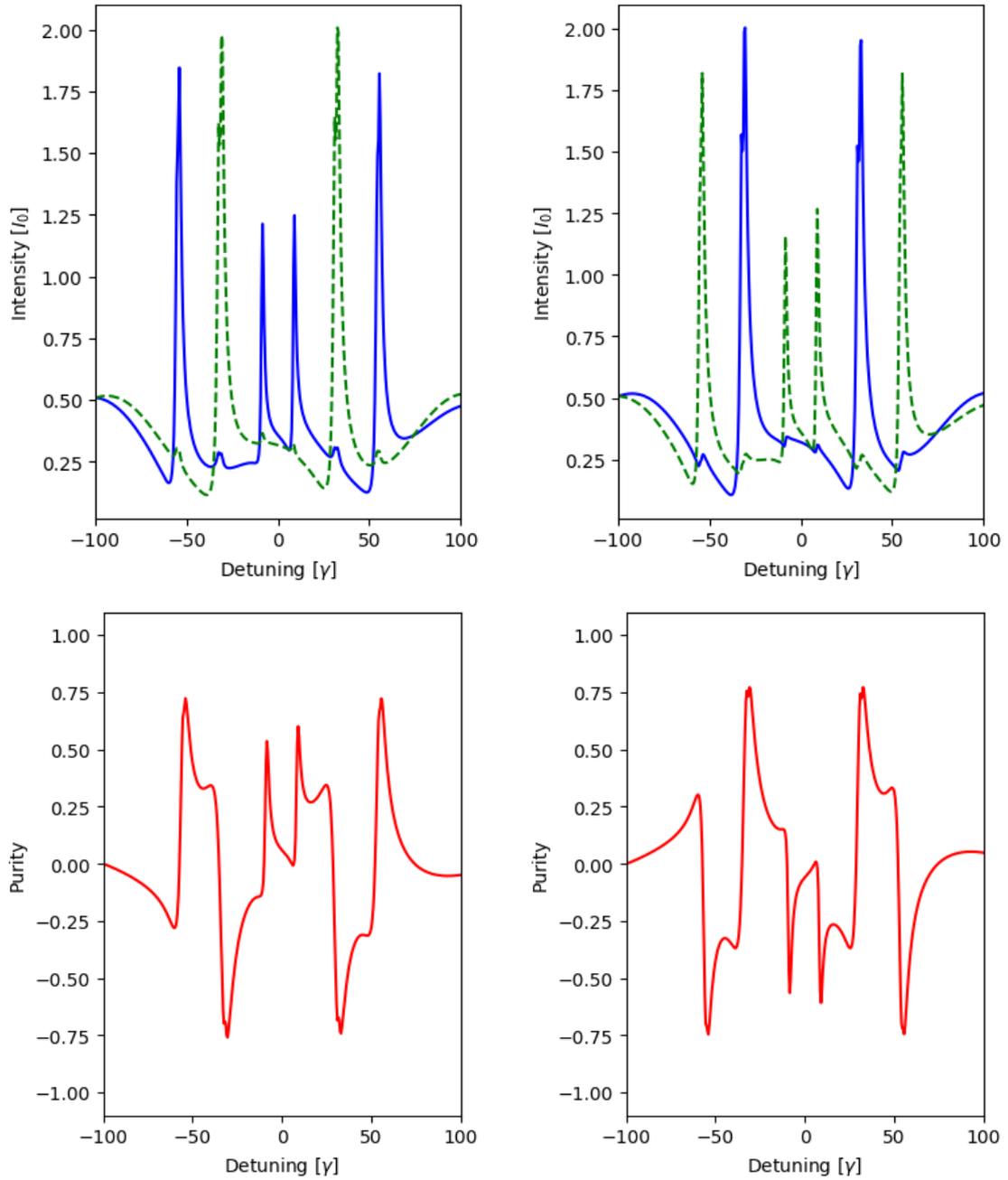


Figure 4.7.: On the left side the spectra for  $\Delta z_1 = \frac{\lambda}{2}$  and  $\Delta z_2 = \frac{\lambda}{4}$  (IT), on the right side the ones for  $\Delta z_1 = \frac{\lambda}{4}$  and  $\Delta z_2 = \frac{\lambda}{2}$  (TI) is shown. All other parameters are like in table 4.1. The blue, dotted and green, dashed lines represent the intensities  $I_+$  and  $I_-$ , whilst the red, solid lines indicate the purity.

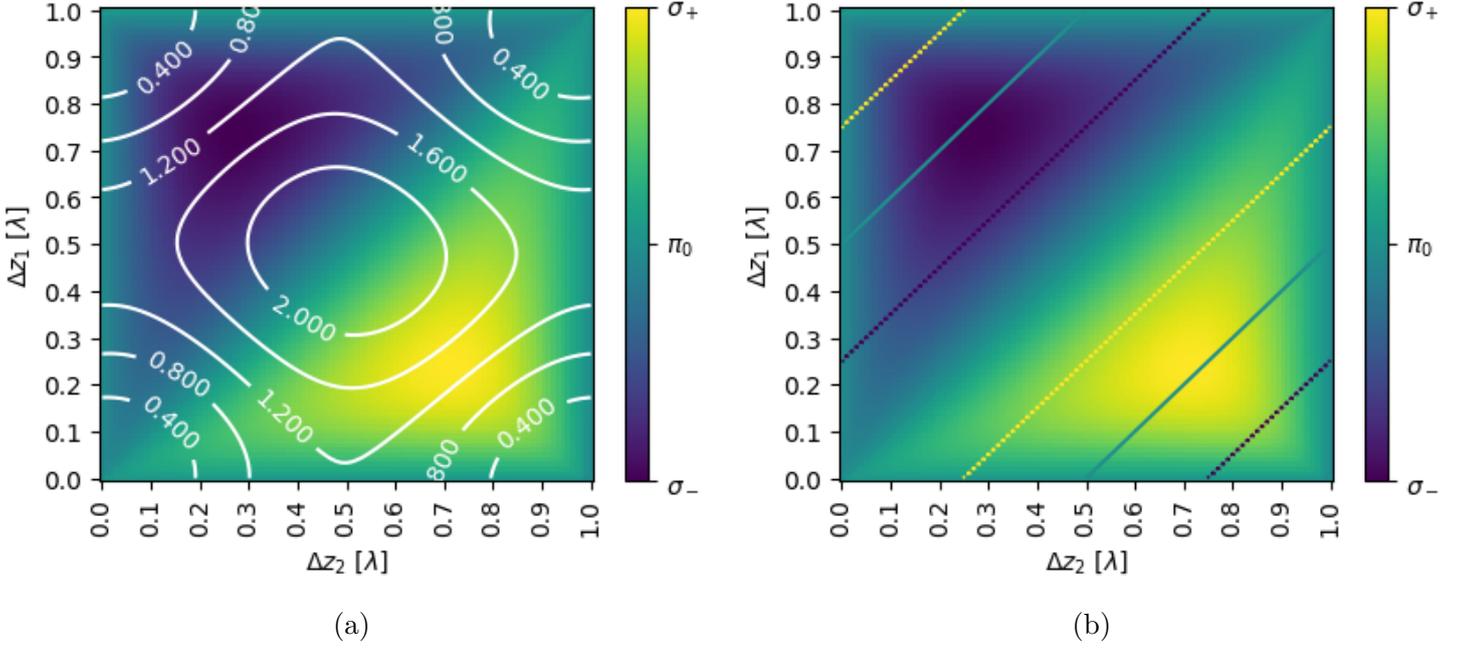


Figure 4.8.: The spectra for the parameters in table 4.1 with varying displacements are investigated. On the left side the influence of on purity (colors) and intensity (contour lines) is shown. On the right, it the numerical purity (colors) is compared to the analytical prediction in fig. 2.3 (colored contour lines). The colors of the lines represent the purity in the same way as the color bar on the right does. It is  $\alpha_1 = \frac{\pi}{4}$  and  $\alpha_2 = -\frac{\pi}{4}$ .

In our illustrative calculations in chapter 3, the two samples were commutative. As `pynuss` includes more effects, we see that they will not be commutative in reality.

#### 4.2.2. Influence of different displacements $\Delta z_i$

Because of the two different configuration TI and IT (see eq. (4.3)) having the same set of angles ( $\alpha_1 = \frac{\pi}{4}, \alpha_2 = -\frac{\pi}{4}$ ), we do not need to distinguish between them to investigate the influence of different displacements  $\Delta \phi_i$ . As before, the spectra are calculated for different displacements (code: appendix A.1.1). Then purity and intensity are taken at  $\Delta = 31.1 \gamma$  and plotted as color plot respectively white contour lines in fig. 4.8a. We see, that the regions of high purity (yellow and blue) are not at the same place as the region of high intensity (middle). This is due to the fact, that intensity increase and polarization transformation need different phase shifts. The region with the highest intensity is at  $\Delta z_1 = \Delta z_2 = \frac{\lambda}{2}$ , which corresponds to intensity increase at both samples.

Furthermore, we see that the configurations in eq. (4.3) do not deliver the highest absolute purity. Hence, there is some space for optimization. Besides, we see that the two configurations with their respective neighborhoods have opposite purity as it was discussed in the context of fig. 4.7.

To make it easier to compare numerical result to the analytical one in fig. 2.3, in

## 4. Results

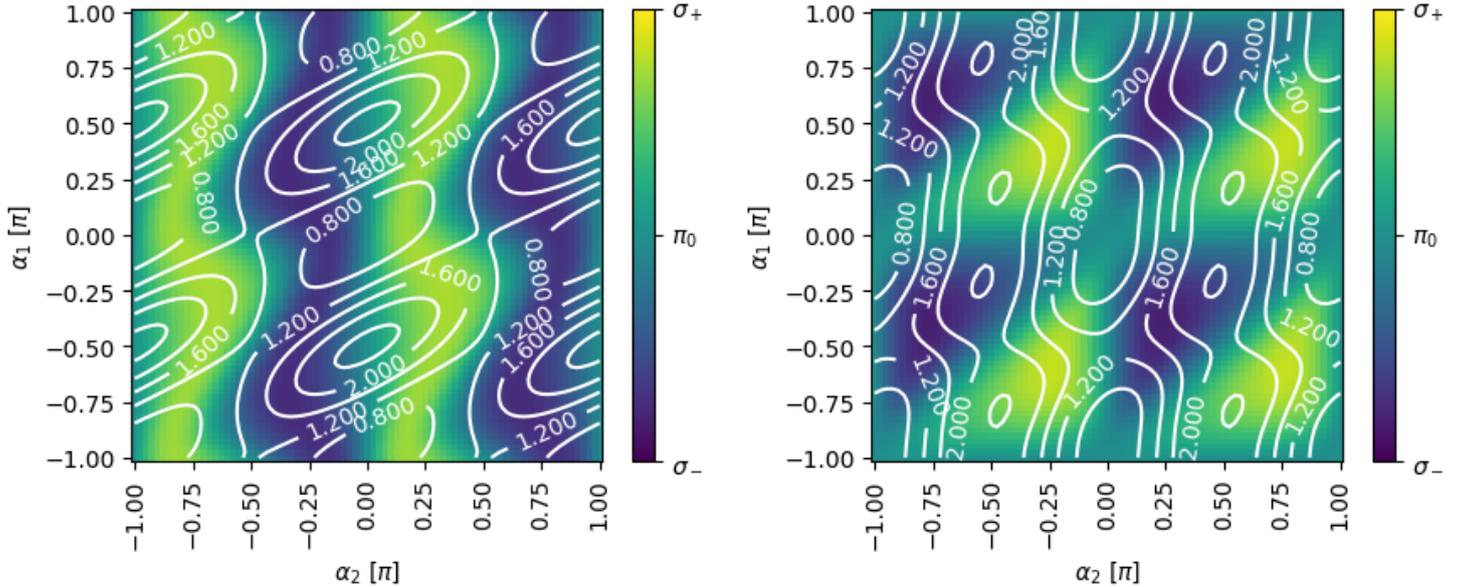


Figure 4.9.: The spectra for the parameters in table 4.1 are investigated for different angles. On the left side the angular dependency for  $\Delta z_1 = \frac{\lambda}{2}$  and  $\Delta z_2 = \frac{\lambda}{4}$  (IT), on the right side the one for  $\Delta z_1 = \frac{\lambda}{4}$  and  $\Delta z_2 = \frac{\lambda}{2}$  (TI) is shown. The white contour lines represent the intensity while the colors indicate the purity.

fig. 4.8b, the numerical purity is plotted as a color plot with the analytical purity as colored contour lines inside. The colors of the contour lines have the same interpretation as the ones in the color map, i.e. yellow indicating pure  $\sigma_+$ -polarization et cetera. We see that the behavior is roughly the way it would be expected from a  $\frac{\lambda}{4}$ -wave plate: We have  $P = 0$  at the diagonal  $\Delta z_1 = \Delta z_2$  and  $P = 1$  ( $P = -1$ ) below (above) that line. But these are the only similarities.

We also see, that the regions of high purity are shifted towards the corners of the figure in the numerical case. Furthermore, at the edge, i.e.  $\Delta_{1/2} \approx 0$  or  $\lambda$ , the purity is also  $P = 0$ . This means, that now conversion into circular polarization has taken place. Besides, we see that the numerical regions with high purity are more widespread ranges instead of thin lines. Thus, also small variations in the displacement would still lead to high purity. These effects might probably be connected, but have not yet been understood.

### 4.2.3. Influence of different angles $\alpha_i$

Because of the two configurations TI and IT having different displacements (see eq. (4.3)), we need to investigate them separately. The numerical results (code: appendix A.1.1) are shown in fig. 4.9. The white contour lines represent the intensity, the colors the purity.

Comparing the two configurations shows big differences. While for IT (left side) the second angle  $\alpha_2$  seems to have a much bigger effect on the purity than  $\alpha_1$  has, for TI (right side) they have more or less the same influence. Also the forms of regions with high absolute purity are quite different. Furthermore, the shapes of the regions of high

n	—P—	$\alpha_1$ [ $\pi$ ]	$\alpha_2$ [ $\pi$ ]	$\Delta z_1$ [ $\lambda$ ]	$\Delta z_2$ [ $\lambda$ ]	$I$ [ $I_0$ ]
1	0.5	0.35	-0.4	0.65	0.45	3.066
2	0.6	0.3	-0.4	0.65	0.4	2.909
3	0.7	-0.35	0.3	0.60	0.35	2.716
4	0.75	0.3	-0.35	0.65	0.35	2.598
5	0.8	-0.35	0.3	0.60	0.30	2.5
6	0.85	-0.35	0.3	0.65	0.30	2.316
7	0.9	-0.35	0.3	0.70	0.30	2.093
8	0.95	0.35	-0.3	0.25	0.65	1.914
9	0.99	0.35	-0.3	0.25	0.70	1.678

Table 4.2.: The parameters, that produce the highest intensity for two samples, are displayed for nine degrees of absolute purity.

intensity are quite different. Nevertheless, the regions with highest purity and the ones with highest intensity are roughly at the same position. Besides, the regions of high purity are large. Hence, small variations in the angles have no big influence on the purity. If we would interchange the angles  $\alpha_1$  and  $\alpha_2$  for IT, so that the phase shift is always combined with the angle  $\alpha_1$ , we would transform the configuration IT into TI. In fact, we would then have roughly, but not exactly, the same behavior on the angles as in the original plot of TI (fig. 4.9), right side). This indicates, that interchanging the samples does not lead to equivalent setups, in contrast to our assumption in chapter 3. Again this is due to `pynuss` considering more effects than our calculation did.

Comparing the numerical results to the analytical ones in fig. 2.4 shows big differences. As mentioned before, in configuration IT the purity depends much more on  $\alpha_2$ , while in the analytical calculations it depends on both angles in a similar way. For TI we can more or less identify the analytical horizontal lines. Nevertheless, while there is nearly no dependency on  $\alpha_2$  in the analytical figure, the influence of  $\alpha_1$  is not negligible in the numerical result. This is due to additional effects included in `pynuss`.

### 4.3. Optimization

In fig. 4.11 the spectrum is shown for this configuration.

During our investigations in the previous sections, we have seen that there might be some space for optimizing the angles  $\alpha_i$  and displacements  $\Delta z_i$  in order to get higher purity and higher intensity. Because of people not always wanting to have  $P \approx \pm 1$ , if this can only be realized with less intensity, the optimization is done for different degrees of purity. To optimize our setup, we loop over  $\alpha_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  and  $\Delta z_i \in [0, \lambda]$  in steps of  $0.05\pi$  and  $0.05\lambda$  respectively (code: appendix A.1.1). The result is shown in fig. 4.10.

On the  $x$ -axis the demanded degree of absolute purity is displayed, on the  $y$ -axis the highest intensity that is possible with the varied parameters for this degree of purity. We

#### 4. Results

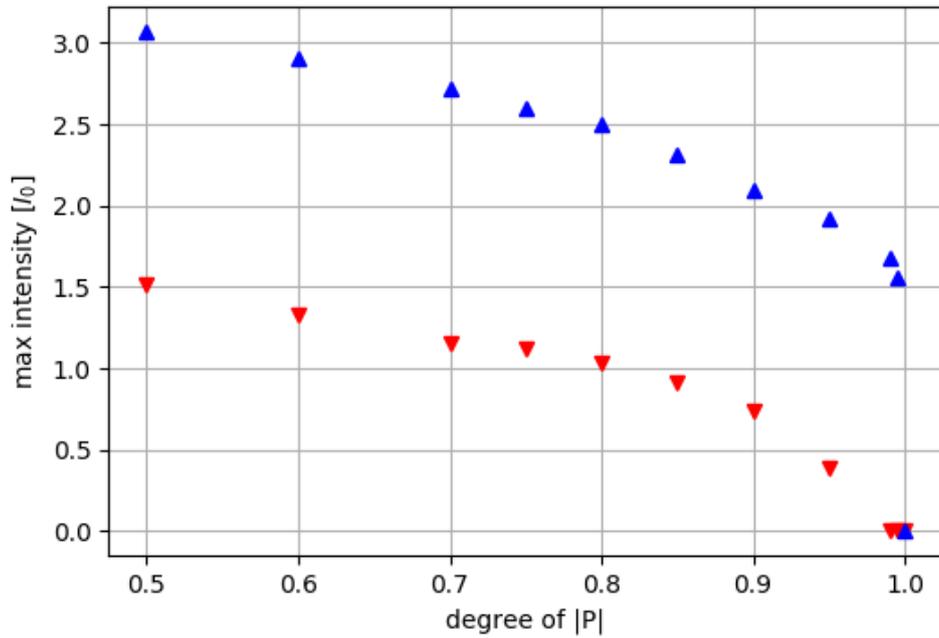


Figure 4.10.: Out of the parameters in table 4.1 the angles  $\alpha_i$  and displacements  $\Delta z_i$  are optimized. For a given degree of absolute purity the maximal intensity is shown. The blue  $\Delta$  are the values for two samples, the red  $\nabla$  for one sample.

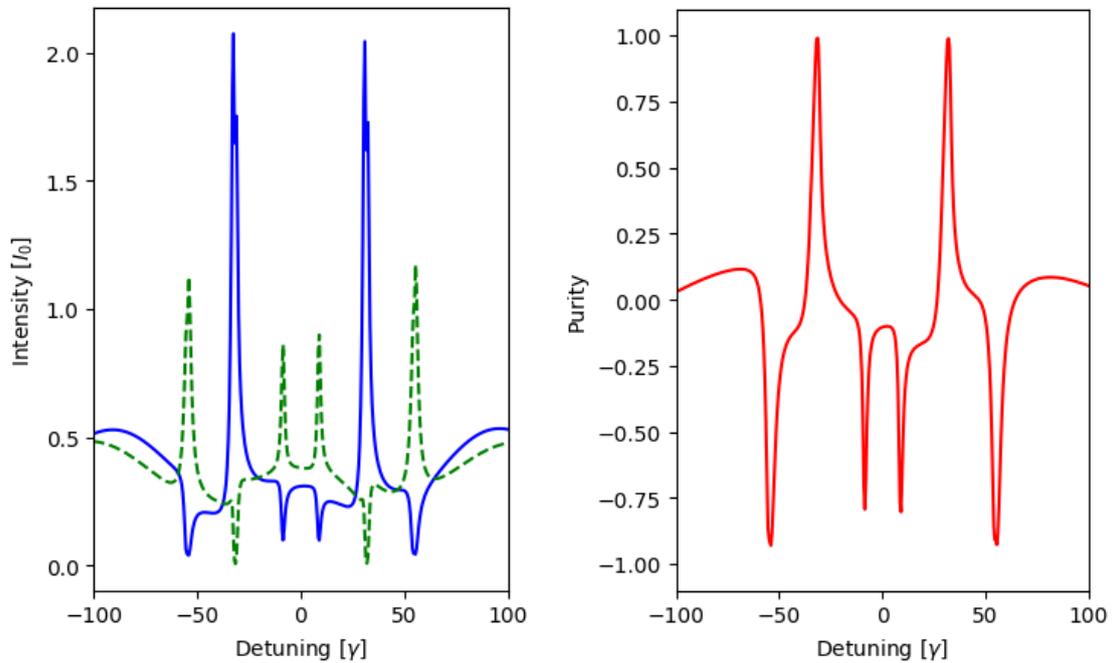


Figure 4.11.: The intensities (left) and purity for the optimal configuration (eq. (4.4)) are shown. The other parameters are taken as in table 4.1. The blue, solid line shows the spectrum after a  $\sigma_+$  polarization filter, the green, dotted line after a  $\sigma_-$  one. The purity is indicated by the red, solid curve.

see that including a second sample has a really big effect on the intensity. As we would expect because of the regions of high purity being different from that of high intensity (see section 4.2), the intensity decreases for a higher degree of purity. Still, we get an intensity of  $1.7I_0$  for  $P = 0.99$  when using two samples. In the optimization we cannot achieve  $P = 1$  due to numerical rounding errors. The best configurations for two sample are shown in table 4.2. We see that the highest intensity is neither for the IT-configuration nor for the TI-configuration, but for a configuration, in which the none of the two samples does only intensity increase or transformation. Furthermore, while for a low degree of freedom the parameters tend towards the  $\frac{\lambda}{2}$ -configuration (see eq. (2.16)), for high degrees of absolute purity they tend towards the  $\frac{\lambda}{4}$ -configuration for the single samples.

The highest intensity at highest degree of absolute purity is realized by setup 9

$$\text{optimum config.: } \alpha_1 = 0.35\pi \quad \alpha_2 = -0.3\pi \quad \Delta z_1 = 0.25\lambda \quad \Delta z_2 = 0.7\lambda. \quad (4.4)$$

It can be seen, that the spectrum in fig. 4.11 is much more symmetric than the spectra in fig. 4.7. Additionally, we see, that we get even an increasing factor of 2 close to the resonant line. Furthermore, at the circular lines we see absorption dips again in  $I_+$ . So not only are the linear lines quite intense and pure, but also the circular lines are suppressed in  $I_+$ . This results in a very clean spectrum with roughly only two peaks at  $\Delta = \pm 31.1\gamma$ . In consequence, this leads to quite a clear time spectrum as we will see in the next section.

## 4.4. Behavior in time space

Until now, we have only investigated the spectra in frequency space because we want to modify the behavior of the light at a specific detuning. By modifying the spectrum accordingly in the frequency space, we can also control polarizations in time space. Therefore, in this section we will have a short look at the spectra in time space (code: appendix A.1.2). We can ask for the time behavior of the total intensity, regarding all frequencies, as well as for the behavior at a certain detuning, here at  $\Delta = -31.1\gamma$ .

### 4.4.1. All frequencies

In fig. 4.12 the time behavior of intensities and purity are shown for the optimal configuration. When looking at the total intensity  $I$ , we can see the quantum beats. Because of our sample being quite thin, there is no dynamical beat visible in the considered time range. We see, that  $I_+$  has a periodicity as proposed with the frequency spectrum of the optimal configuration. The minima of  $I_+$  have a time difference of

$$\Delta t = 14.0 \pm 0.6 \text{ ns}. \quad (4.5)$$

#### 4. Results

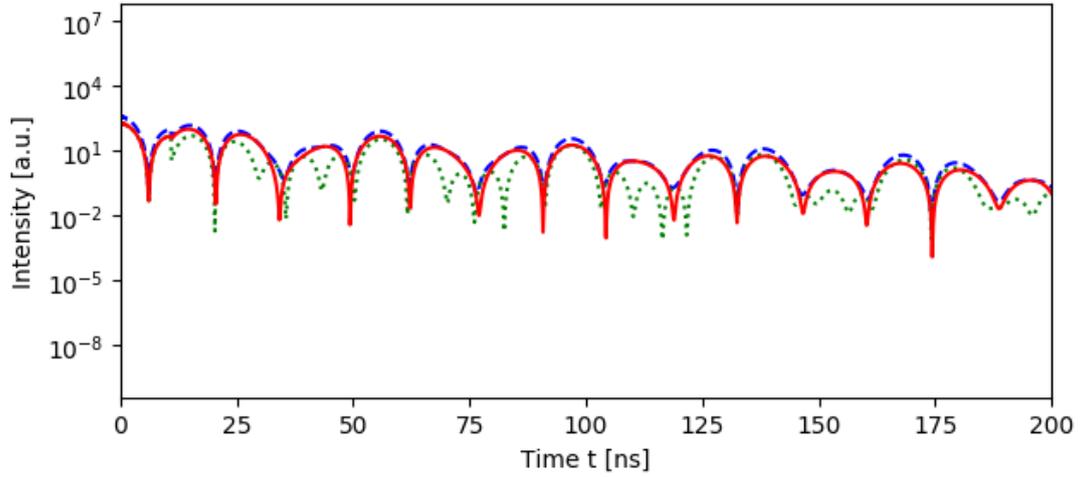


Figure 4.12.: The time spectrum of the optimum configuration (eq. (4.4)) is shown. All other parameters are as in table 4.1. Here the blue line indicates  $I$ , the green one  $I_-$  and the red one  $I_+$ .

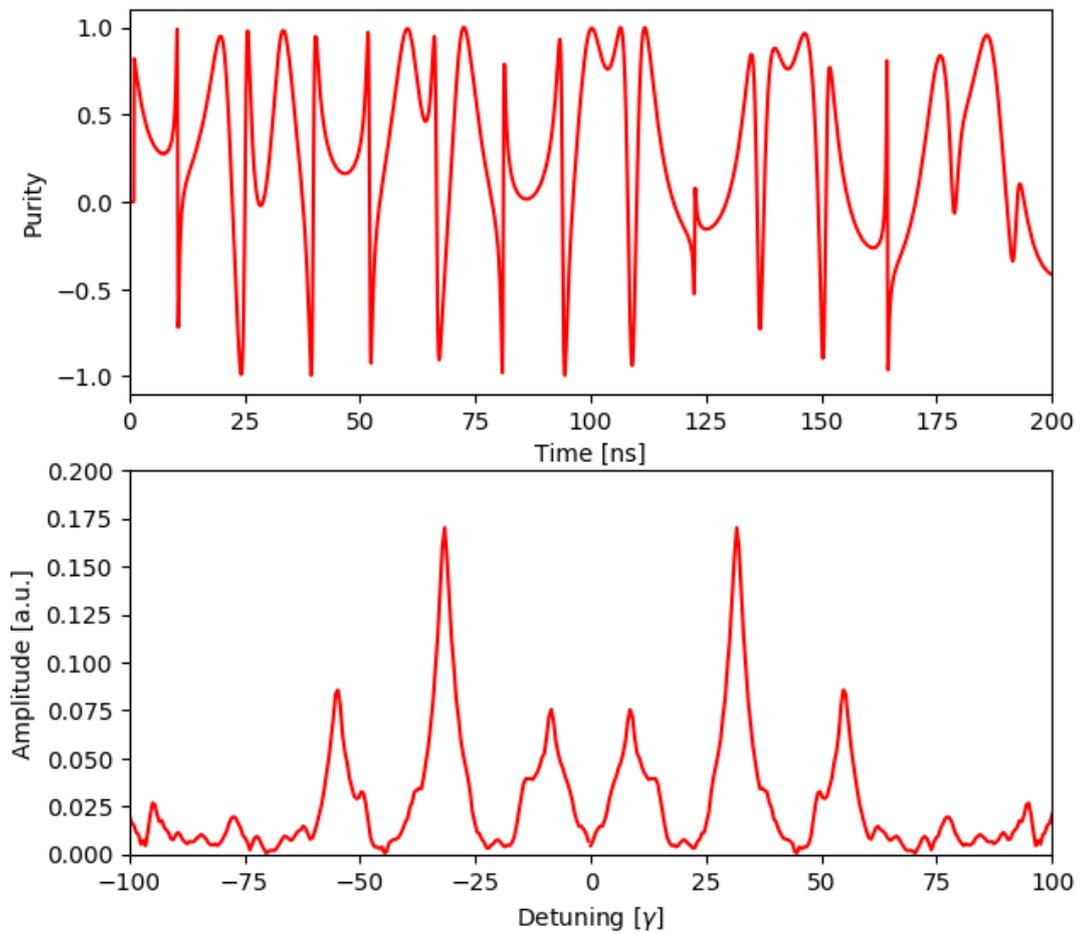


Figure 4.13.: The time dependency of the purity (top) and its Fourier transform (bottom) are shown for the total spectrum. The parameters are as in table 4.1, but with the angles and displacements of the optimal configuration in eq. (4.4).

This is also the time scale on which the purity is changing. Unfortunately,  $I_-$ , that is not that periodic, is on a similar intensity level resulting in a quite heavy oscillating purity in the region around  $P \approx +1$ , while the periodicity of  $I_+$  is dominant at  $P \approx -1$ . By comparing the minima of purity and intensity, we observe that nearly all points of minimal purity, i. e. these where light is  $\sigma_-$  polarized instead of  $\sigma_+$ , also the intensity is very low. That means, that in the time domain, we get more or less pulses of length  $\Delta t$  with purity  $P \approx 1$ .

To further analyze the time dependence, we Fourier transform the purity. This transformation is not the inverse of the Fourier transform that produced the time spectrum. This is because in the original transformation the electric field was transformed instead of its square, which defines purity. The Fourier transform is shown in fig. 4.13. We can identify the peaks with the frequencies of the nuclear transitions (see table 3.1). Besides, the two main peaks correspond to the linear lines. So we see, that the frequencies of the linear lines dominate the time spectra as it would be expected.

In experiment, it should be possible to block out certain frequencies in order to obtain a cleaner time spectrum.

#### 4.4.2. Around $-31.1\gamma$

By calculating two-dimensional energy-time-spectra, it is possible to look at the time behavior of one frequency. Here, we choose to look at the linear line by integrating up the intensities of the two-dimensional spectrum in the interval  $[-32.1\gamma, -30.1\gamma]$ . We choose this integration range, because we do not wanted to observe artefacts of the other transitions nor do we want to see effects of small oscillations which might occur along a very small range of detuning. The intensities  $I_+$  and  $I_-$  are displayed in fig. 4.14. We cannot see any periodicity in the intensities  $I_+$  and  $I_-$  in the figure. Therefore, we apply a Fourier transform to find any hidden structures. The time dependency of the purity and its Fourier transform are displayed in fig. 4.15. To make the identification of the frequencies easier, also vertical lines are plotted at the transition frequencies. We can identify the prominent peaks with the six transitions. If we compare the result of the Fourier analysis to the result for the whole detuning range (see fig. 4.13), we see that the peaks of the linear transitions are smaller in contrast to the other ones. For the small detuning range, they are nearly as high as the outer circular lines. Therefore, the time dependency looks so complicated.

## 4.5. Other applications

After investigating the  $\frac{\lambda}{4}$ -wave plate in detail, we will now look at some other applications of our setup. We only provide a proof of principle in this thesis. A deeper investigation

#### 4. Results

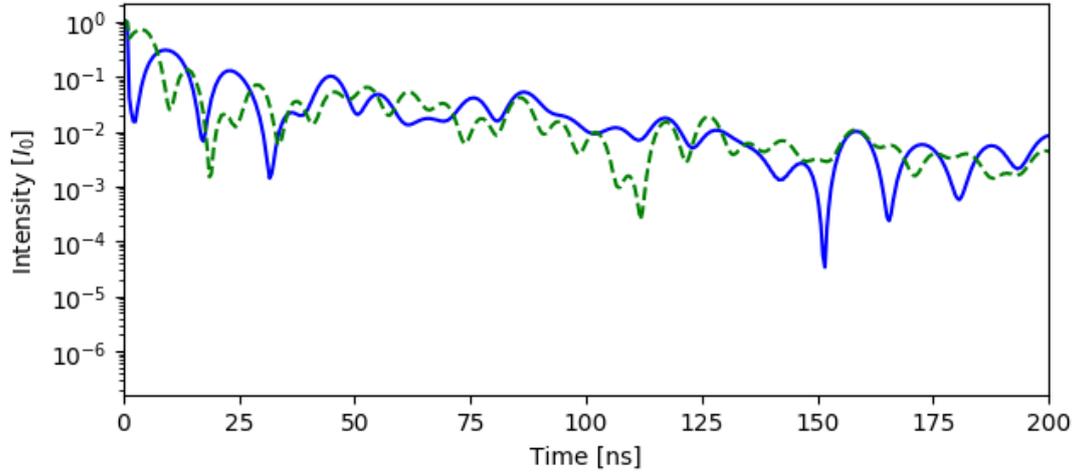


Figure 4.14.: The time dependency of the integrated intensities  $I_+$  (blue, solid line) and  $I_-$  (green, dashed line) at the integration range of  $[-32.1 \gamma, -30.1 \gamma]$  are shown.

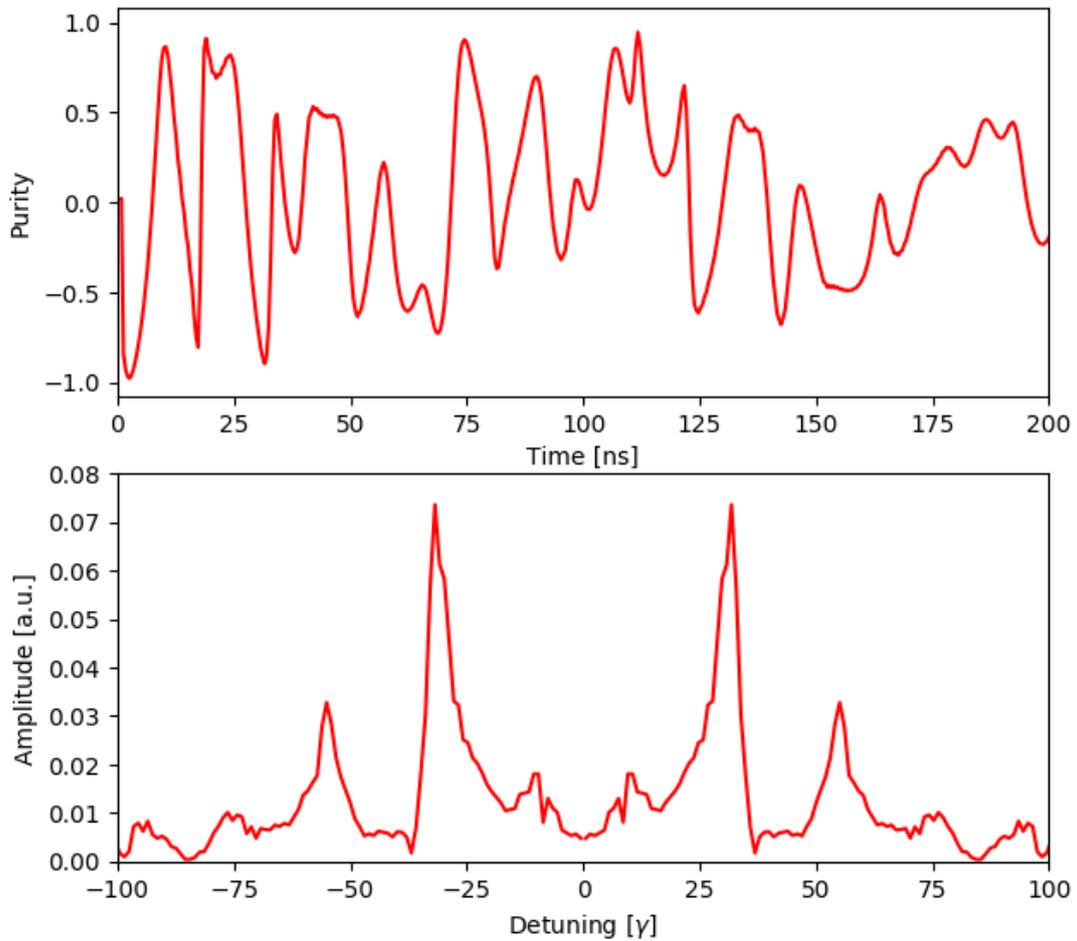


Figure 4.15.: The time dependency of the purity (top) and its Fourier transform (bottom) integrated over  $[-32.1 \gamma, -30.1 \gamma]$  are shown. The parameters are as in table 4.1, but with the angles and displacements of the optimal configuration eq. (4.4).

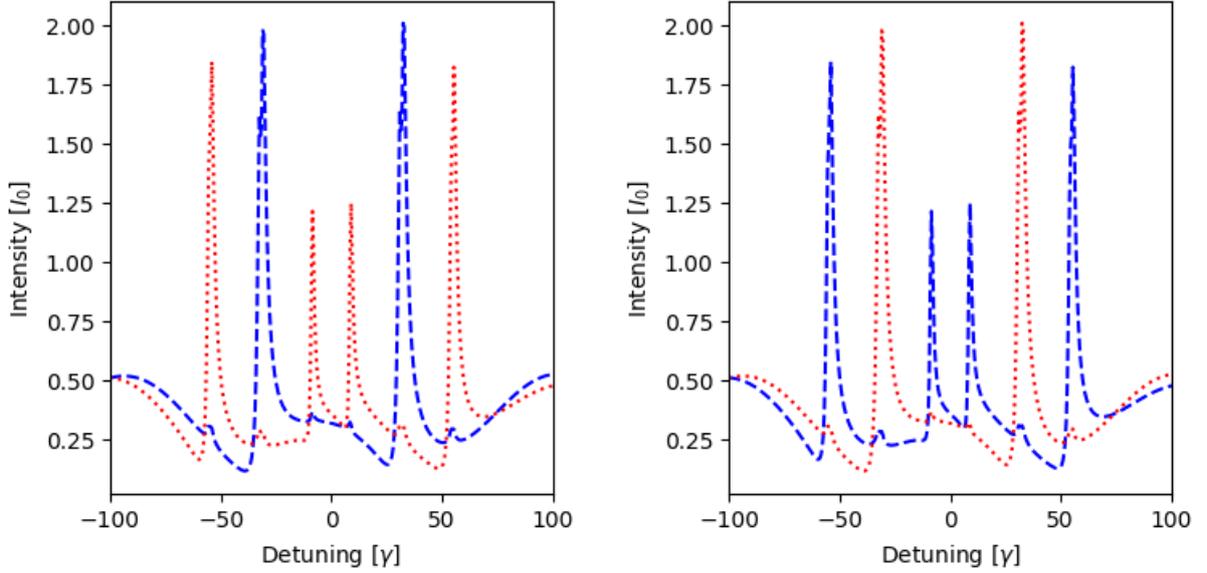


Figure 4.16.: We choose the IT-configuration (see eq. (4.3)) to realize a polarization filter. The other parameters are as in table 4.1. On the left side, the spectrum for a beam with polarization  $\sigma_+$  is shown. On the right side we display the spectrum for a beam with  $\sigma_-$ . The blue (dashed) line is the spectrum of a detector with polarization  $\pi_0$ , the red (dotted) line of one with  $\pi_{0,\perp}$ .

is reserved for future work.

#### 4.5.1. Circular polarization filter

In the previous sections, we discussed the conversion of linearly polarized light into circularly polarized light. However, our setup can also be used the other way around, i. e. to analyze circularly polarized light by transforming it into linearly polarized light. The so-obtained linearly polarized light can then be analyzed by a linear polarization filter, that exists (see [12]).

In fig. 4.16 the principle function of a polarization filter as realized by our setup is shown (code: appendix A.2.1). We chose the IT setup because then the intensity increase takes in circularly polarized light and the angle  $\alpha_1$  can be arbitrary. It can be clearly seen, that the incoming circularly polarized light is transformed into  $\pi_0$  or  $\pi_{0,\perp}$  polarized light, depending on the handedness. Applying a linear polarization filter in the right angle behind our setup, results in filtering out one handedness of circularly polarized light.

#### 4.5.2. $\frac{\lambda}{2}$ -wave plate and light switch

As it can be seen in eq. (2.18), one can also use our setup to build a  $\frac{\lambda}{2}$ -wave plate, that rotates the linearly polarized light by  $2\alpha$ , where  $\alpha$  is the angle of the magnetic dipole moment. Again, one sample would be enough to rotate the polarization as it was for

## 4. Results

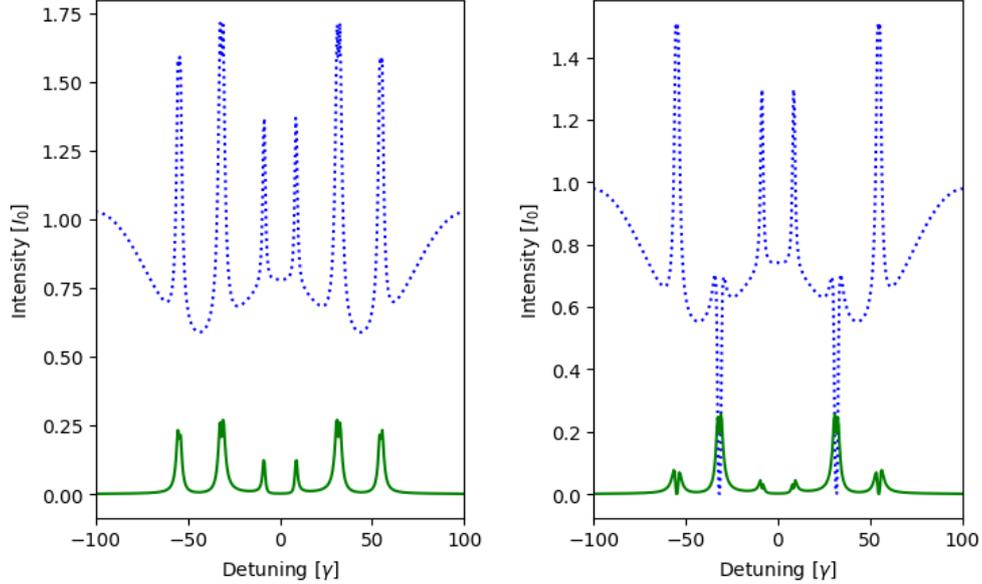


Figure 4.17.: The spectra for a  $\frac{\lambda}{2}$ -wave plate are shown. On the left side for using one sample ( $\alpha = \frac{\pi}{4}$  and  $\Delta z = \frac{\lambda}{2}$ ), on the right side for two samples ( $\alpha_1 = \frac{\pi}{4}$ ,  $\alpha_2 = \frac{\pi}{2}$  and  $\Delta z_1 = \Delta z_2 = \frac{\lambda}{2}$ ). The blue (dashed) line is the intensity of the detector in the beam polarization direction ( $\pi_0$ ), the green (solid) for a perpendicular one ( $\pi_{0,\perp}$ ).

the transformation of polarization for the  $\frac{\lambda}{4}$ -wave plate. The rotation of polarization and the intensity increase both need a phase shift of  $\pi$ . Thus, the intensity increase is already included in the rotation. If another sample should be applied for a higher intensity increase, the angles should be chosen to be  $\alpha_2 = 2\alpha_1$ . Then the angle of the rotated polarization is not affected by the second intensity increase as it is  $\vec{J} \equiv -\vec{J}$  (see eq. (2.3)).

Exemplary, the spectrum for a rotation of  $\frac{\pi}{2}$  ( $\alpha = \frac{\pi}{4}$ ) is shown for one sample and two samples in fig. 4.17 (code: appendix A.2.2). To make the rotation visible, the intensities  $I_{\parallel}$  and  $I_{\perp}$  corresponding to the detectors with linear polarization filters parallel respectively perpendicular to the incoming polarization are taken for the calculation instead of the intensities  $I_{\pm}$ . For no rotation, it should be  $I_{\perp} = 0$ . For the dynamical case of one and two samples as shown in the figure, it clearly is  $I_{\perp} \neq 0$ . Hence, the rotation has worked. For two samples it is even  $I_{\parallel} \approx 0$ , which means, that the polarization of nearly all light has been rotated.

Oftentimes, the experimental construction allows to be rotated and no  $\frac{\lambda}{2}$ -wave plate is needed. However, it might be useful to have fast time control on the angle of linear polarization.

Nevertheless, a  $\frac{\lambda}{2}$ -wave plate can be used to construct a fast light switch. Consider a polarimeter (see [12]) to filter out the unscattered synchrotron pulse that is not wanted in forward scattering synchrotron experiments. The basic setup of the polarimeter respec-

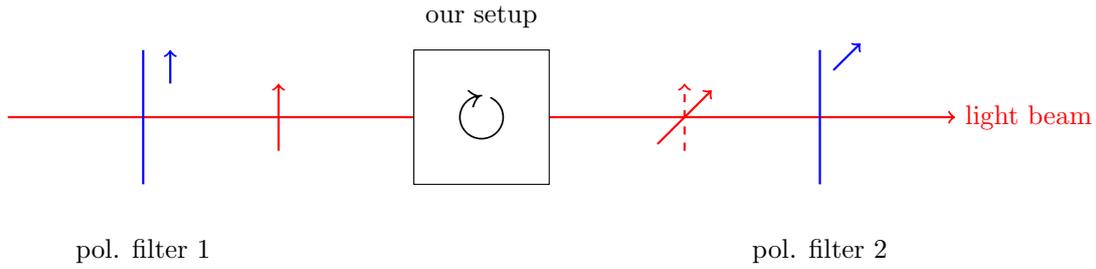


Figure 4.18.: The setup of a fast light switch is shown. It consists of two linear polarization filters perpendicular to each other (blue) and our setup (black), respectively some nuclei in the original polarimeter setup, that rotates the polarization (solid red arrow) or not (dashed red arrow). The polarizations are also shown (small arrows).

tively the light switch is shown in fig. 4.18. It basically consists of a first polarization filter in the direction of the beam polarization, next some nuclei which rotate the polarization by  $\frac{\pi}{2}$  and finally a second polarization filter perpendicular to the first one. Then only the scattered, i.e. the rotated, light can get through the second polarization filter while the prompt pulse is blocked out. Currently this setup has big losses at the rotation. With our  $\frac{\lambda}{2}$ -wave plate, we could rotate the polarization between the two polarization filters with even an increase of intensity. This combination of a polarimeter and our  $\frac{\lambda}{2}$ -wave plate could then also be used to switch on and off a light beam on a fast time scale by changing the polarization so that the second polarization filter lets the beam through or blocks it out.

### 4.5.3. Highly precise displacement measurement

As we have seen in fig. 4.6, our setup is quite sensible to variations in angle or displacement at around  $\Delta\alpha = \frac{\pi}{2}$  and  $\Delta z = 0$ . This could be used to measure variations of angle and displacement. Because of it being more common to measure highly precise displacements, we will look at this case in the following. The variation of angles could be detected in a similar way.

We again take two samples in order to increase the intensity. This is because the intensities are quite small in this region otherwise. We choose the first angle to be  $\alpha_1 = \frac{\pi}{4}$  and the second angle close to  $\alpha = \frac{\pi}{2}$  (here  $\alpha_2 = 0.52\pi$ ). The displacement of first sample is taken as  $\Delta z_1 = \frac{\lambda}{2}$  and we slightly vary the displacement  $\Delta z_2$ . The result is shown in fig. 4.19 (code: appendix A.1.1).

It can be seen, that the two points of high purity have a distance of  $\Delta z \approx 0.2\lambda < \frac{\lambda}{2}$ , which is normally the border of resolution for a measurement with wavelength  $\lambda$ . By varying  $\Delta z_1$ , the range where the variation it easy to measure should be able to be shifted by parts of  $\lambda$ , as it can be seen in the right part of fig. 4.19.

To use this application in experiments, the knowledge of two out of the three intensities

## 4. Results

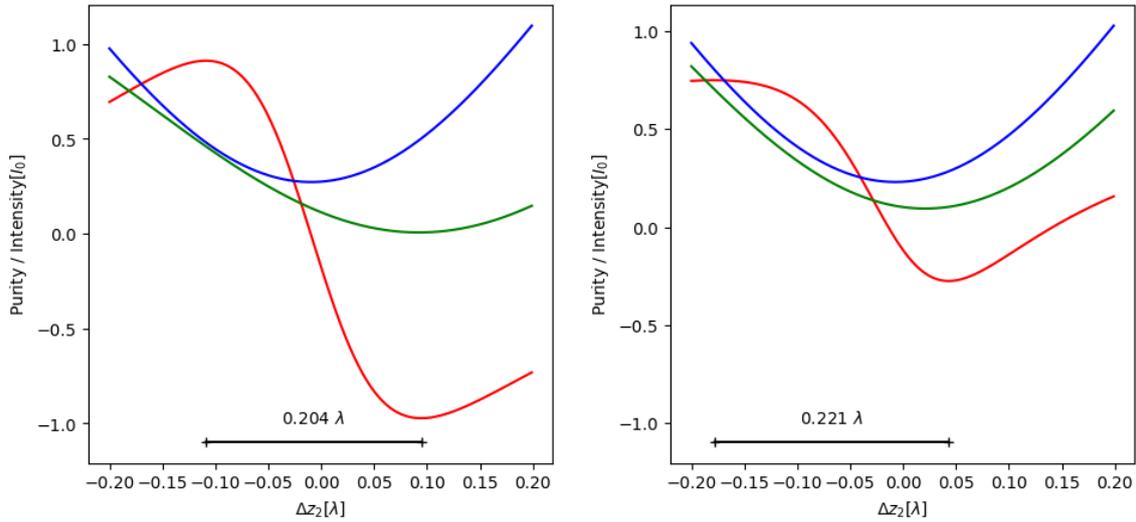


Figure 4.19.: On the left side, purity (red) and the intensities  $I$  (blue) and  $I_+$  (green) are shown for  $\Delta z_1 = 0.5\lambda$ . On the right side we have the same plot for  $\Delta z_1 = 0.3\lambda$ . The angles are chosen to be  $\alpha_1 = \frac{\pi}{4}$ ,  $\alpha_2 = 0.52\pi$ . All other parameters are as in table 4.1.

$I_0$ ,  $I_+$  and  $I_-$  is needed. Otherwise, the purity cannot be calculated.

For a first proof of principle experiment instead of purity the intensity  $I_+$  can be measured. It also differs for varying displacements, but the magnitude is much lower than that of the purity (see fig. 4.19).

## 5. Comparisons with optical elements

In this chapter, we compare our new approach to already existing technologies in the optical regime.

### 5.1. Standard wave plate

As we wanted to create an element, that converts linear into circular polarization, it is interesting to look at advantages and disadvantages in contrast to a standard  $\frac{\lambda}{4}$ -wave plate.

**Intensity.** While for an ideal common  $\frac{\lambda}{4}$ -wave plate we get the same intensity in front of and behind the wave plate. In experiments, the intensity is slightly decreased after passage through the plate. In our approach we even obtain an increased intensity after the wave plate for the wavelength of interest. The increase depends on the required purity. For  $P = 0.995$  it is increased by a factor of 1.7.

**Wavelength.** As we have seen in eq. (2.11), standard  $\frac{\lambda}{4}$ -wave plates have a wavelength dependency. Nevertheless, it is possible to cover a big range, e.g. from 310 nm to 1100 nm (see [13]), by using achromatic crystals. In our approach, the transformation can only be done at the linear lines that are really thin. This is fairly limiting. This limitation can be reduced by using different sample materials because they have different resonant wavelengths

**Flexibility.** If a common  $\frac{\lambda}{4}$ -wave plate is adjusted, it is fixed and it requires a lot of effort to change the handedness of circularly polarized light or the angle of linearly polarized one. Our setup in contrast is very flexible in this aspect. This is because the purity can be controlled electronically. It makes it possible to change the purity on a time scale of the reaction time of the piezo. Concrete, it is possible to achieve these variations on a time scale of  $\approx 10$  ms.

With the  $\frac{\lambda}{2}$  wave plate setup we can control the angle of polarization instead of the purity. This can also be achieved on the same time scale.

## 5. Comparisons with optical elements

**Time.** Despite the intensity difference, the time dependence might be the biggest difference. The effect of a standard  $\frac{\lambda}{4}$ -wave plate is constant in time. In contrast our setup allows to control the polarization in time space.

### 5.2. Pockels cells

In optics *Pockels Cells* are applied in order to do fast and arbitrary polarization changes. In these elements, the diffraction index is changed by an electric field due to the *Pockels effect*. Unfortunately, until now there are no Pockels cells in the X-ray regime, but only in optical domain. Standard Pockels cells work on the time scale of milliseconds (see [14]).

Our setup has the potential of achieving a similar functionality. Depending on the reaction time of the piezo, we could achieve even shorter shorter response times than standard Pockels cells. However, we do not already have enough time control on our setup to use it as a similar element as a Pockels cell, even though we already have dynamical polarization control and can achieve arbitrary polarizations by arbitrary phase shifts. Hence, our proposed setup would fill the gap of an element with the functionality of a Pockels cell in the X-ray regime.

## 6. Summary and Outlook

In this thesis, an approach for dynamical polarization control in X-ray quantum optics by using nuclear forward scattering was proposed. As the main application, the transformation of linear into circular polarization was discussed. Nuclear forward scattering is used to produce a phase shift of the scattered light relative to the unscattered one. Selecting a magnetization, that is not perpendicular to the beam polarization, is used in order to decompose the incoming polarization into one part, that obtains a phase shift, whilst the other part remains unaffected. By modelling the phase shift by a suitable instantaneous displacement and by choosing the angle of the magnetization appropriately, we can achieve arbitrary polarization conversion.

At first, the function of a standard wave plate as well as its dependency on angle and phase shift were discussed for one and two combined plates (chapter 2). After this, our setup was introduced and explained (chapter 3). Subsequently, this setup was investigated numerically (chapter 4) by use of the python library `pynuss`. Most effects were especially explored at the nuclear transitions with linear polarization. It could be seen, that the setup works in principle and that an intensity increase could be realized by use of a second sample. For the influence of angle and phase shift we obtained similar results for the numerical (chapter 4) and analytical calculations (chapter 3) for one sample. For two samples we found some similarities in the angular and displacement behavior respectively. However, there appeared big differences, that originate from the effects that are included in `pynuss`. The regions of high purity are shifted with respect to the analytical calculations. The purity structure for the different angles looked much more complicated than analytically expected. In the future, we would like to understand the origin of these effects. An optimization of the parameters of angle  $\alpha_1$  and  $\alpha_2$  and displacement  $\Delta z_1$  and  $\Delta z_2$  showed, that for a purity of  $P = +0.99$  an intensity of  $I_+ = 1.7I_0$  could be achieved. This means, that very high purities with a significant intensity increase can be obtained. We have also seen, that we have a varying purity in time space. By blocking out certain frequencies, it should be possible to have time control of the purity.

As further applications of our setup a polarization filter, a  $\frac{\lambda}{2}$ -wave plate, a light switch and a high precision sensor were discussed in section 4.5. As of now, we have merely provided a proof of principle. If these proposed setups were to be applied, they should be further investigated.

A comparison with a standard wave plate and a Pockels cell showed, that our setup

## 6. Summary and Outlook

has nearly all features of a wave plate combined with the ones of a Pockels cell – at least in frequency space. Our distinct disadvantage compared to standard Pockels cells and wave plates is that our setup has a much smaller range of operational wavelength because we have to focus on resonance lines. It can be faced by using different materials. Nevertheless, it is probably not possible to get as big wavelength ranges as for ordinary wave plates.

In the future, we want to further develop the control in time space. Besides, it would be interesting to further investigate the features of the other applications, especially the displacement measurement. Furthermore, it might be worthwhile to think about transferring the approach into the optical regime. Despite good optical with similar functions exist, the sensitivity of our setup might be interesting.

Besides, the setup should be tested in experiments. A first proof of principle experiment was conducted during a beamtime at DESY. The spectrum of one sample in the  $\frac{\lambda}{4}$ -configuration could be analyzed by another  $^{57}\text{Fe}$ -foil that was mounted on a Mössbauer drive. For linear polarization (static case) the spectrum should depend on the angle of the magnetization of the second foil, while for circular polarization in the  $xy$ -plane (dynamical case) there should be no dependency. Unfortunately, the measured data show very small effects. We believe, that this was due to technical problems in the experimental setup. A detailed data evaluation of the obtained measurements, a repetition of this proof of principle experiment as well as first experimental tests of the mentioned applications are reserved for future work.

**Acknowledgements.** I would like to thank the Max Planck Institute for Nuclear Physics, especially the division of Christoph Keitel, for hospitality. I enjoyed the pleasant working atmosphere at the institute and the opportunity to take part in a beam time at DESY. The interesting conversations with Thomas Pfeiffer and Christian Ott were very enlightening. My special thanks go to the research group of Jörg Evers. Especially, I want to thank Kilian Heeg, Dominik Lentrodt and Benedikt Herkommer for stimulating discussions. Let me also thank Celine Karle, Dominic Lentrodt, Jan Arne and Martin Bies for proofreading this thesis.

I am very grateful to my supervisor Jörg Evers for giving me the opportunity to work on such an interesting topic. Your encouragements and valuable suggestions made this Bachelor thesis possible.

Last but not least, I want to express my gratitude to my friends, with whom I really enjoyed my studies, and especially to my family, that supported me during my Bachelor studies. Without you, this would not have been possible!

# A. Used python-scripts

In this chapter, all the used scripts are displayed.

## A.1. Analysis of polarization transformation

With these two scripts the calculations of the analysis of the polarization conversion was done.

### A.1.1. Frequency space

For better understanding, the script, that was used for the analysis in frequency space, is divided into different parts.

#### Preamble

```
1 import functools
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm
import scipy.special
import pynuss
from matplotlib.cm import get_cmap
from matplotlib.colors import Normalize
from scipy.optimize import curve_fit
from scipy.signal import find_peaks
11 import datetime

# get current filename
import sys
import os
file_name = os.path.basename(sys.argv[0])

# measure running time
import time
start = time.time()
```

#### Parameters

```
### general parameters

# consider only one sample
one_sample = False

# real or ideal motion
motion_real = False
t_rise = 2. #15. experimental
t_rise = 15.0

10 # samples
layer_thickness1 = 1e-6 # m
layer_thickness2 = 1e-6 # m
layer_thickness3 = 1e-12 #m (no sample)
```

## A. Used python-scripts

```
internal_magnetic_field = 33 # Tesla

# method to find extremum
20 pos_of_purity = -(39.7+22.4)/2 # theoretical prediction

#method = 'extremum-2' # extremum of intensity
method = 'theoretical_value'
#method = 'fit'
#method = 'purity'
#method = 'purity2'

### parameters for the different parts (if not set, the paramters of "no loop" are taken)
30 # loops
no_loop = True
loop_displacement = False
loop_thickness = False
loop_angle = False
loop_angle_dis = False
loop_everything = False
loop_dis_1D = False
40 #loop_angle_1D = False

# no loop
alpha1 = np.pi*0.35
alpha2 = -np.pi*0.3

displacement1 = 0.25 # times resonant wavelength
displacement2 = 0.7

save_pic = "pictures/opt"
50 ideal_vs_real = False

# ... over displacements
dis1_start = -0.0 #res. wavelength
dis1_end = 1.01
dis1_step = 0.01
dis2_start = -0.0
dis2_end = 1.01
dis2_step = 0.01
60 file_name_dis = 'files/dis_20'

# ... over thicknesses
thicknesses_loop = [1e-7, 1e-6, 1e-5]
save_pic_thicknesses = 'pictures/thicknesses_one'

# ... over different relative phase shifts
70 shift_start = -0.5
shift_end = 0.5
shift_step = 0.1
dis1_shift = 0.3

# ... over different absolute phase shifts
dis_start = 0.0
dis_end = 1.1
dis_step = 0.1
80 phase_shift = 0.25

# ... over different angles
angle1_start = -1 #pi
angle1_end = 1.00001
angle1_step = 0.025
angle2_start = -1.0
angle2_end = 1.00001
```

```

90 angle2_step = 0.025
    disp1 = 0.5
    disp2 = 0.25

    file_name_angles = 'files/angles_21'

    #... over one angle and displacement
    first_fixed = False
100 angle_start = -1 #pi
    angle_end = 1.0001 #pi
    angle_step = 0.01

    displacement_start = -1
    displacement_end = 1.0000001
    displacement_step = 0.01

    angle_fixed = np.pi/4
    displacement_fixed = 0.25
110 file_name_angle_dis = 'files/angle_dis_6'

    # ... loop over everything
    thicknesses = np.array([1e-6])
    angles1 = np.arange(-0.5* np.pi, 0.501 *np.pi, 0.025*np.pi)
    angles2 = np.arange(-0.5* np.pi, 0.501 *np.pi, 0.025*np.pi)
    displacements1 = np.arange(0, 1.000001, 0.0025)
    displacements2 = np.arange(0, 1.000001, 0.0025)
120 if one_sample == True:
        displacements2 = np.arange(0, 1, 1)
        angles2 = np.arange(0, 1, 1)

    min_purity = np.array([0.5, 0.6, 0.7,0.75, 0.8, 0.85, 0.9, 0.95, 0.99, 0.995, 0.999, 1])

    file_name_everything = "files/loop_one_sample_025_3_merged_results.txt"
130 # ... loop dis 1D (one sample)
    dis_1D_start = -0.2
    dis_1D_end = 0.2
    dis_1D_step = 0.001

    angle_dis_1D = 0.52 * np.pi

    alpha1_dis_1D = np.pi/4
    dis1_dis_1D = 0.3
140 file_name_dis_1D = 'pictures/dis_1D_5'

```

## Definition of functions

```

1 ##### FUNCTIONS FOR CALCULATIONS #####

##### check whether folder 'pictures' and 'files' exist and create them
if not os.path.exists('pictures'):
    os.makedirs('pictures')
if not os.path.exists('files'):
    os.makedirs('files')

##### combine paramtere accordingly for calculation
11 ##### set up beam and detector
Beam = pynuss.Beam([0,0,1])
Detector_plus = pynuss.Detector(Beam)
Detector_minus = pynuss.Detector(Beam)

```

## A. Used python-scripts

```

# set polarizations
Beam.SetLinearPolarization([1,0,0])
Detector_plus.SetCircularFilter(+1)
Detector_minus.SetCircularFilter(-1)

21 beam_direction = [0,0,1]

magnetisation_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
magnetisation_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

# arrays
# ... over different displacements
dis1 = np.arange(dis1_start, dis1_end, dis1_step)
31 dis2 = np.arange(dis2_start, dis2_end, dis2_step)

# ... over different relative phase shifts
shifts = np.arange(shift_start, shift_end, shift_step)

# ... over different absolute phase shifts
displacements = np.arange(dis_start, dis_end, dis_step)

# ... over different angles
41 angle1 = np.arange(angle1_start, angle1_end, angle1_step) * np.pi
angle2 = np.arange(angle2_start, angle2_end, angle2_step) * np.pi

# ... over angle and displacement
angle = np.arange(angle_start, angle_end, angle_step) * np.pi
displacement = np.arange(displacement_start, displacement_end, displacement_step)

# ... over dis 1D
dis_1D = np.arange(dis_1D_start, dis_1D_end, dis_1D_step)

51 beam_pol = str([1,0,0])

# function to create samples
def define_sample(thickness, magnetization, int_field):
    eFe = pynuss.ResonantElement.fromTemplate('Fe57')
    eFe.MagneticHyperfineField = int_field
    eFe.SetMagnetizationDirection(magnetization)
    mFe = pynuss.Material.fromElement(eFe)
    lFe = pynuss.Layer(mFe, thickness)
    return lFe

61 def Fit_gaussian(x, A, mu, sigma, offset):
    return A*np.exp(-0.5*((x-mu)/sigma)**2) + offset

# algorithm to define position where purity and intensity are taken
def define_position(method_loc, pos0, I, purity, Detuning):
    # theoretical value
    position_Det = pos0 # theoretical prediction

71 # position index
step_Det = Detuning[1]-Detuning[0]
position = int(len(Detuning)/2 + position_Det / step_Det)

maxima_det = 0

if 'extremum_2' == method_loc:
    pos2 = np.argmax(Intensity_F[position-10:position+10]) - 10
    position = position + pos2
    maxima_det = (position - len(Detuning)/2) *step_Det

81 if 'purity' == method_loc:
    position = np.argmax(abs(purity))
    position_value = True
    method = "max_|purity|"

    maxima_det = (position - len(Detuning)/2) *step_Det

if 'purity2' == method_loc:

```

## A.1. Analysis of polarization transformation

```

    if (abs(purity[position]) > 0.2): # no Fano line, no linear pol
        position_relative = np.argmax(abs(purity[int(position-3/step_Det):int(position+3/
91         ↪ step_Det)])) -3
        position = position_relative + position
        maxima_det = (position - len(Detuning)/2) *step_Det

    return position, maxima_det

# calculate the spectrum
def calculate_spectra(displacement_1, displacement_2):

    def MotionSmoothStep1(t):
101     # Times in ns
        tRise = t_rise
        tShift = 10.
        lambda0 = 8.60254801e-11 # m
        x0 = displacement_1 * lambda0
        if motion_real == True:
            return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
        else:
            return x0 * np.heaviside((t - tShift -1), 0.5)

    def MotionSmoothStep2(t):
111     # Times in ns
        tRise = t_rise
        tShift = 10.
        lambda0 = 8.60254801e-11 # m
        x0 = displacement_2 * lambda0
        if motion_real == True:
            return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
        else:
            return x0 * np.heaviside((t - tShift -1), 0.5)

121     # set up frequency domain calculation
    fw1_plus = pynuss.ForwardScattering(Beam, Detector_plus, lFe1) # normal spectrum
    fw2_plus = pynuss.ForwardScattering(Beam, Detector_plus, lFe2)
    fw1_minus = pynuss.ForwardScattering(Beam, Detector_minus, lFe1) # spectrum with other
        ↪ handedness
    fw2_minus = pynuss.ForwardScattering(Beam, Detector_minus, lFe2)

    if one_sample == True:
        # combine samples, detector plus (
131     cs_plus = pynuss.tools.CombineSamples(
        (fw1_plus.TransmissionMatrix, MotionSmoothStep1), # sample 1
        )
        # combine samples, detector minus (
        cs_minus = pynuss.tools.CombineSamples(
        (fw1_minus.TransmissionMatrix, MotionSmoothStep1), # sample 1
        )
    else:
        # combine samples, detector plus (
        cs_plus = pynuss.tools.CombineSamples(
141     (fw1_plus.TransmissionMatrix, MotionSmoothStep1), # sample 1
        (fw2_plus.TransmissionMatrix, MotionSmoothStep2) # sample 2
        )
        # combine samples, detector minus (
        cs_minus = pynuss.tools.CombineSamples(
        (fw1_minus.TransmissionMatrix, MotionSmoothStep1), # sample 1
        (fw2_minus.TransmissionMatrix, MotionSmoothStep2) # sample 2
        )

151     # compute ResponseMatrix
    Detuning = cs_plus.DetuningGrid(100, 0.5, 800, 0.2)
    DetuningStep = Detuning[1] - Detuning[0]
    RM_F_stat_plus = cs_plus.ResponseStatic(Detuning) # static response
    RM_F_stat_minus = cs_minus.ResponseStatic(Detuning) # static response
    RM_F_dyn_plus = cs_plus.ResponseWithMotion(Detuning) # dynamic response
    RM_F_dyn_minus = cs_minus.ResponseWithMotion(Detuning) # dynamic response

    # compute intensity

```

## A. Used python-scripts

```
161 Intensity_F_stat_plus = RM_F_stat_plus.Intensity()
Intensity_F_stat_minus = RM_F_stat_minus.Intensity()
Intensity_F_dyn_plus = RM_F_dyn_plus.Intensity()
Intensity_F_dyn_minus = RM_F_dyn_minus.Intensity()

# Transform into time domain
RM_T_stat_plus, TStep = pynuss.tools.FreqToTime(RM_F_stat_plus, DetuningStep)
RM_T_stat_minus, TStep = pynuss.tools.FreqToTime(RM_F_stat_minus, DetuningStep)
RM_T_dyn_plus, TStep = pynuss.tools.FreqToTime(RM_F_dyn_plus, DetuningStep)
RM_T_dyn_minus, TStep = pynuss.tools.FreqToTime(RM_F_dyn_minus, DetuningStep)
Times = np.arange(len(Detuning)) * (TStep)

171 # compute intensity in time domain
Intensity_T_stat_plus = RM_T_stat_plus.Intensity()
Intensity_T_stat_minus = RM_T_stat_minus.Intensity()
Intensity_T_dyn_plus = RM_T_dyn_plus.Intensity()
Intensity_T_dyn_minus = RM_T_dyn_minus.Intensity()

# compute total intensity
181 Intensity_F_stat = Intensity_F_stat_plus + Intensity_F_stat_minus
Intensity_T_stat = Intensity_T_stat_plus + Intensity_T_stat_minus
Intensity_F_dyn = Intensity_F_dyn_plus + Intensity_F_dyn_minus
Intensity_T_dyn = Intensity_T_dyn_plus + Intensity_T_dyn_minus

# compute purity
purity_F = (Intensity_F_dyn_plus - Intensity_F_dyn_minus)/Intensity_F_dyn
purity_T = (Intensity_T_dyn_plus - Intensity_T_dyn_minus)/Intensity_T_dyn
purity_F_stat = (Intensity_F_stat_plus - Intensity_F_stat_minus)/Intensity_F_stat
purity_T_stat = (Intensity_T_stat_plus - Intensity_T_stat_minus)/Intensity_T_stat

191 values_F_stat = [Intensity_F_stat_plus, Intensity_F_stat_minus, Intensity_F_stat,
                  ↪ purity_F_stat]
values_T_stat = [Intensity_T_stat_plus, Intensity_T_stat_minus, Intensity_T_stat,
                 ↪ purity_T_stat]

# intensity array to return
Intensity = [Intensity_F_dyn, Intensity_F_dyn_plus, Intensity_F_dyn_minus, values_F_stat,
            ↪ Intensity_T_dyn, Intensity_T_dyn_plus, Intensity_T_dyn_minus, values_T_stat]

return Intensity, purity_F, purity_T, Detuning, Times

# functions for theoretical calculation
201 def rotation(alpha):
    return np.array([[np.cos(alpha), -np.sin(alpha)], [np.sin(alpha), np.cos(alpha)]])

def waveplate(alpha, phi):
    M = np.array([[1, 0], [0, np.exp(1j*phi)]])
    return rotation(-alpha).dot(M.dot(rotation(alpha)))

def pol_filter(sign):
    return 1/2*np.array([[1, -1j*sign], [1j*sign, 1]])
```

## Spectrum

```
if no_loop == True:
    # create samples
    lFe1 = define_sample(layer_thickness1, magnetisation_pol1, internal_magnetic_field)
    lFe2 = define_sample(layer_thickness2, magnetisation_pol2, internal_magnetic_field)

7 # Calculate spectra
Intensity, purity_F_array, purity_T, Detuning, Times = calculate_spectra(displacement1,
                                  ↪ displacement2)

Intensity_F = Intensity[0]
Intensity_F_dyn_plus = Intensity[1]
Intensity_F_dyn_minus = Intensity[2]
Intensity_F_stat = Intensity[3]
Intensity_T_dyn = Intensity[4]
Intensity_T_dyn_plus = Intensity[5]
```

```

Intensity_T_dyn_minus = Intensity[6]
Intensity_T_stat = Intensity[7]
17
if ideal_vs_real == True:
    # real vs ideal motion
    motion_real = False
    Intensity_ideal, purity_F_ideal, purity_T_ideal, Detuning_ideal, Times_ideal =
        ↪ calculate_spectra(displacement1, displacement2)

    motion_real = True
    t_rise = 2.0
    Intensity_real, purity_F_real, purity_T_real, Detuning_real, Times_real =
        ↪ calculate_spectra(displacement1, displacement2)
27
    t_rise = 15.0
    Intensity_real2, purity_F_real2, purity_T_real2, Detuning_real2, Times_real2 =
        ↪ calculate_spectra(displacement1, displacement2)

# shift of maximum
position_num = define_position('extremum.2', pos_of_purity, Intensity_F_dyn_plus,
    ↪ purity_F_array, Detuning)[0]
position_real = (-len(Detuning)//2 + position_num)*(Detuning[1]-Detuning[0])
print('shift:', round(position_real-pos_of_purity,3))

### plot
37 # spectra for thesis
# intensities
fig, ax1 = plt.subplots(figsize = (3.5,5))

ax1.set_xlabel('Detuning_[$\gamma$]')
ax1.set_ylabel('Intensity_[$I_0$]')

ax1.set_xlim([-100, 100])

47 ax1.plot(Detuning, Intensity_F_dyn_plus, label='Moving_+$I_+$', linestyle = "-", color = "blue"
    ↪ ")
ax1.plot(Detuning, Intensity_F_dyn_minus, label='Moving_-$I_-$', linestyle = "-", color = "
    ↪ green")

fig.savefig(save_pic+'_int', bbox_inches='tight')

# purity
fig, ax1 = plt.subplots(figsize = (3.5,5))

ax1.set_xlabel('Detuning_[$\gamma$]')
ax1.set_ylabel('Purity')
57
ax1.set_xlim([-100, 100])
ax1.set_ylim([-1.1, 1.1])

ax1.plot(Detuning, (Intensity_F_dyn_plus - Intensity_F_dyn_minus)/(Intensity_F_dyn_plus +
    ↪ Intensity_F_dyn_minus), label='$\frac{I_+-I_-}{I_++I_-}$', linestyle = "-", color =
    ↪ "red")

fig.savefig(save_pic+'_pur', bbox_inches='tight')

# real vs ideal motion
67
if ideal_vs_real == True:
    fig, axes = plt.subplots(nrows=1,ncols=2, figsize=(7.4,5))

    ax1 = axes[0]
    ax2 = axes[1]

    ax1.set_xlabel('Detuning_[$\gamma$]')
    ax1.set_ylabel('Intensity')
    ax1.set_xlim([-100, 100])

77
    ax1.plot(Detuning, Intensity_ideal[1], linestyle = "-", color = "green")
    ax1.plot(Detuning, Intensity_ideal[2], linestyle = "-", color = "blue")

    ax1.plot(Detuning, Intensity_real[1], linestyle = "-", color = "green")

```

## A. Used python-scripts

```
ax1.plot(Detuning, Intensity_real[2], linestyle = "—", color = "blue")

ax1.plot(Detuning, Intensity_real2[1], linestyle = ":", color = "green")
ax1.plot(Detuning, Intensity_real2[2], linestyle = ":", color = "blue")

ax2.set_xlabel('Detuning_['gamma]')
ax2.set_ylabel('Purity')
87 ax2.set_xlim([-100, 100])
ax2.set_ylim([-1.1, 1.1])

ax2.plot(Detuning, purity_F_ideal, label='ideal_,'purity', linestyle = "—", color = "red")
ax2.plot(Detuning, purity_F_real, label='real_,'purity', linestyle = "—", color = "red")
ax2.plot(Detuning, purity_F_real2, label='real_,'purity', linestyle = ":", color = "red")

fig.subplots_adjust(wspace=0.4)
plt.savefig(save_pic+'IvR', bbox_inches='tight')
```

## Different displacements

```
if loop_displacement == True:
2   print("loop_over_displacements")
   print(alpha1, alpha2)
   print("written_into", file_name_dis)

   # create samples
   lFe1 = define_sample(layer_thickness1, magnetisation_pol1, internal_magnetic_field)
   lFe2 = define_sample(layer_thickness2, magnetisation_pol2, internal_magnetic_field)

   phi_theo1 = dis1 *2*np.pi
   phi_theo2 = (dis2) *2*np.pi
12  offset_alpha1 = np.pi/2
   offset_alpha2 = np.pi/2
   lin = np.array([1, 0])

   maxima_det = [[-100 for x in range(len(dis2))] for y in range(len(dis1))]
   purity = [[-2 for x in range(len(dis2))] for y in range(len(dis1))]
   purity_theo = [[-2 for x in range(len(dis2))] for y in range(len(dis1))]

22  for i in range(len(dis1)):
     for j in range(len(dis2)):
         # nice control in terminal
         end = time.time()
         current_time = round(end-start)
         current_time_form = str(datetime.timedelta(seconds=current_time))
         percentage = (i*len(dis1)+j)/(len(dis1)*len(dis2))*100
         if percentage != 0:
             remaining_time = round((100/percentage-1)*current_time)
             remaining_time_form = str(datetime.timedelta(seconds=remaining_time))
32  else:
         remaining_time_form = ''
         print('completed:_{0:.3f}%,_running:_{1:6}s,_remaining:_{2:6}s_\'r'.format(percentage,
             ↪ current_time_form, remaining_time_form))

         # Calculate spectra
         Intensity, purity_F_array, purity_T, Detuning, Times = calculate_spectra(dis1[i],
             ↪ dis2[j])

         Intensity_F = Intensity[0]
         Intensity_F_dyn_plus = Intensity[1]
42  Intensity_F_dyn_minus = Intensity[2]
         Intensity_F_stat = Intensity[3]
         Intensity_T_dyn = Intensity[4]
         Intensity_T_dyn_plus = Intensity[5]
         Intensity_T_dyn_minus = Intensity[6]
         Intensity_T_stat = Intensity[7]
```

```

position , maxima_det[i][j] = define_position(method, pos_of_purity , Intensity_F ,
    ↪ purity_F_array , Detuning)

52  purity[i][j] = purity_F_array[position]

    # calculate theoretical purity
    lin2 = waveplate(alpha1+offset_alpha1 , phi_theo1[i]).dot(lin)
    lin4 = waveplate(alpha2+offset_alpha2 , phi_theo2[j]).dot(lin2)

    pol_plus = pol_filter(+1).dot(lin4)
    pol_minus = pol_filter(-1).dot(lin4)
    I_plus = np.sqrt(abs(pol_plus[0])**2 + abs(pol_plus[1])**2)
    I_minus = np.sqrt(abs(pol_minus[0])**2 + abs(pol_minus[1])**2)

62  purity_theo[i][j] = (I_plus-I_minus) / (I_plus + I_minus)

# write into file
    f_pur = open(file_name_dis+'_purity.txt','a')
    f_int = open(file_name_dis+'_intensity.txt','a')
    f_theo = open(file_name_dis+'_purity_theo.txt','a')

72  # settings
    settings = ["#_angle_1:_ " + str(alpha1), "#_angle_2:_ " + str(alpha2), "#_thickness_1:_"+str(
        ↪ layer_thickness1), "#_thickness_2:_ " + str(layer_thickness2), "#_real_motion:_"+str(
        ↪ motion_real), "#_method_to_find_position:_"+str(method), "#_used_code:_"+str(file_name)
        ↪ ]

    f_pur.write("%s\n" % len(settings))
    f_int.write("%s\n" % len(settings))
    f_theo.write("%s\n" % len(settings))

    for item in settings:
        f_pur.write("%s\n" % item)
        f_int.write("%s\n" % item)
        f_theo.write("%s\n" % item)

82  # x-axis
    np.savetxt(f_pur , np.array(dis2)[None] , delimiter=',')
    np.savetxt(f_int , np.array(dis2)[None] , delimiter=',')
    np.savetxt(f_theo , np.array(dis2)[None] , delimiter=',')
    # y-axis
    np.savetxt(f_pur , np.array(dis1)[None] , delimiter=',')
    np.savetxt(f_int , np.array(dis1)[None] , delimiter=',')
    np.savetxt(f_theo , np.array(dis1)[None] , delimiter=',')

92  pur = np.array(purity)
    inten = np.array(maxima)
    pur_theo = np.array(purity_theo)
    np.savetxt(f_pur , pur , delimiter = ',')
    np.savetxt(f_int , inten , delimiter = ',')
    np.savetxt(f_theo , pur_theo , delimiter = ',')

    # close files
    f_pur.close()
    f_int.close()
102  f_theo.close()

```

## Different thicknesses

```

1  if loop_thickness == True:
    I_plus = [0 for x in range(len(thicknesses_loop))]
    I_minus = [0 for x in range(len(thicknesses_loop))]
    purity = [0 for x in range(len(thicknesses_loop))]

    for i in range(len(thicknesses_loop)):
        # create samples
        lFe1 = define_sample(thicknesses_loop[i], magnetisation_pol1, internal_magnetic_field)
        lFe2 = define_sample(thicknesses_loop[i], magnetisation_pol2, internal_magnetic_field)

```

## A. Used python-scripts

```
11      # calculate spectra
      Intensity, purity[i], purity-T, Detuning, Times = calculate_spectra(displacement1,
      ↪ displacement2)

      I_plus[i] = Intensity[1]
      I_minus[i] = Intensity[2]

      # plot results
      style = [':', '-', '-']
21      plt.figure(figsize=(6,4))
      for i in range(len(thicknesses_loop)):
          plt.plot(Detuning, I_plus[i], color = 'red', linestyle = style[i])
          plt.plot(Detuning, I_minus[i], color = 'blue', linestyle = style[i])

      plt.xlim(-100,100)
      plt.xlabel('Detuning_[$\gamma$]')
      plt.ylabel('Intensity_[$I_0$]')
      plt.savefig(save_pic_thicknesses, bbox_inches='tight')
      print('saved')
```

## Different angles

```
if loop_angle == True:
    print("loop_over_angles")
    print(displ1, disp2)
4    print("written_into", file_name_angles)

    # create samples
    phi_theo1 = disp1 *2*np.pi
    phi_theo2 = disp2 *2*np.pi
    offset_alpha1 = np.pi/2
    offset_alpha2 = np.pi/2
    lin = np.array([1, 0])

14    maxima_det = [[0 for x in range(len(angle2))] for y in range(len(angle1))]
    purity = [[0 for x in range(len(angle2))] for y in range(len(angle1))]
    purity_theo = [[0 for x in range(len(angle2))] for y in range(len(angle1))]

    for i in range(len(angle1)):
        for j in range(len(angle2)):
            # nice control in terminal
            end = time.time()
            current_time = round(end-start)
            current_time_form = str(datetime.timedelta(seconds=current_time))
24            percentage = (i*len(angle1)+j)/(len(angle1)*len(angle2))*100
            if percentage != 0:
                remaining_time = round((100/percentage-1)*current_time)
                remaining_time_form = str(datetime.timedelta(seconds=remaining_time))
            else:
                remaining_time_form = ''
            print('completed:_{0:.3f}%,_running:_{1:6}s,_remaining:_{2:6}s_'.format(percentage
            ↪ , current_time_form, remaining_time_form))

            # create samples
34            alpha1 = angle1[i]
            alpha2 = angle2[j]
            mag_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
            mag_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

            lFe1 = define_sample(layer_thickness1, magnetisation_pol1, internal_magnetic_field)
            lFe2 = define_sample(layer_thickness2, magnetisation_pol2, internal_magnetic_field)

            # Calculate spectra
            Intensity, purity_F_array, purity-T, Detuning, Times = calculate_spectra(displ1, disp2
44            ↪ )
```

```

Intensity_F = Intensity[0]
Intensity_F_dyn_plus = Intensity[1]
Intensity_F_dyn_minus = Intensity[2]
Intensity_F_stat = Intensity[3]
Intensity_T_dyn = Intensity[4]
Intensity_T_dyn_plus = Intensity[5]
Intensity_T_dyn_minus = Intensity[6]
Intensity_T_stat = Intensity[7]

54 # calcualte puridity at given position

position_Det = -(39.7+22.4)/2 # theoretical prediction
position = int(len(Detuning)/2 + position_Det * 2) # position index

position, maxima_det[i][j] = define_position(method, pos_of_purity, Intensity_F,
↪ purity_F_array, Detuning)

64 # calculate theoretical purity
lin2 = waveplate(angle1[i]+offset_alpha1, phi_theo1).dot(lin)
lin4 = waveplate(angle2[j]+offset_alpha2, phi_theo2).dot(lin2)

pol_plus = pol_filter(+1).dot(lin4)
pol_minus = pol_filter(-1).dot(lin4)
I_plus = np.sqrt(abs(pol_plus[0])**2 + abs(pol_plus[1])**2)
I_minus = np.sqrt(abs(pol_minus[0])**2 + abs(pol_minus[1])**2)

purity_theo[i][j] = (I_plus-I_minus) / (I_plus + I_minus)

74 # write into file
f_pur = open(file_name_angles+'_purity.txt','a')
f_int = open(file_name_angles+'_intensity.txt','a')
f_theo = open(file_name_angles+'_purity-theo.txt','a')

# settings
settings = ["#_displacement_1:_ " + str(displ1), "#_displacement_2:_ " + str(displ2), "#_thickness
↪ _1:_"+str(layer_thickness1), "#_thickness_2:_ " + str(layer_thickness2), "#_real_motion:
↪ _"+str(motion_real), "#_method_to_find_position:_"+str(method), "#_used_code:_"+str(
↪ file_name)]

84 f_pur.write("%s\n" % len(settings))
f_int.write("%s\n" % len(settings))
f_theo.write("%s\n" % len(settings))

for item in settings:
    f_pur.write("%s\n" % item)
    f_int.write("%s\n" % item)
    f_theo.write("%s\n" % item)

# x-axis
94 np.savetxt(f_pur, np.array(angle2)[None], delimiter=',')
np.savetxt(f_int, np.array(angle2)[None], delimiter=',')
np.savetxt(f_theo, np.array(angle2)[None], delimiter=',')
# y-axis
np.savetxt(f_pur, np.array(angle1)[None], delimiter=',')
np.savetxt(f_int, np.array(angle1)[None], delimiter=',')
np.savetxt(f_theo, np.array(angle1)[None], delimiter=',')

104 pur = np.array(purity)
inten = np.array(maxima)
pur_theo = np.array(purity_theo)
np.savetxt(f_pur, pur, delimiter = ',')
np.savetxt(f_int, inten, delimiter = ',')
np.savetxt(f_theo, pur_theo, delimiter = ',')

# close files
f_pur.close()
f_int.close()
f_theo.close()

```

## Different angle and displacement

```

if loop_angle_dis == True:
    print('angle_and_dis')
    print('saved_into', file_name_angle_dis)

5
    # create samples
    if one_sample == True:
        first_fixed = False

    if first_fixed == True:
        dis1 = displacement_fixed
        alpha1 = angle_fixed
    else:
        dis2 = displacement_fixed
        alpha2 = angle_fixed

15
    maxima_det = [[0 for x in range(len(displacement))] for y in range(len(angle))]
    purity = [[0 for x in range(len(displacement))] for y in range(len(angle))]

    for i in range(len(angle)):
        for j in range(len(displacement)):
            # nice control in terminal
            end = time.time()
            current_time = round(end-start)
            current_time_form = str(datetime.timedelta(seconds=current_time))
            percentage = (i*len(angle)+j)/(len(angle)*len(displacement))*100
            if percentage != 0:
                remaining_time = round((100/percentage-1)*current_time)
                remaining_time_form = str(datetime.timedelta(seconds=remaining_time))
            else:
                remaining_time_form = ''
            print('completed:_{0:.3f}%,_running:_{1:6}s,_remaining:_{2:6}s_\'r\'.format( percentage
                ↪ , current_time_form , remaining_time_form))

            # create sample
            if first_fixed == True:
                alpha2 = angle[i]
                dis2 = displacement[j]
            else:
                alpha1 = angle[i]
                dis1 = displacement[j]

            mag_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
            mag_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

            lFe1 = define_sample(layer_thickness1 , mag_pol1 , internal_magnetic_field)
            lFe2 = define_sample(layer_thickness2 , mag_pol2 , internal_magnetic_field)

            # Calculate spectra
            Intensity , purity_F_array , purity_T , Detuning , Times = calculate_spectra(dis1 , dis2)

            Intensity_F = Intensity[0]
            Intensity_F_dyn_plus = Intensity[1]
            Intensity_F_dyn_minus = Intensity[2]
            Intensity_F_stat = Intensity[3]
            Intensity_T_dyn = Intensity[4]
            Intensity_T_dyn_plus = Intensity[5]
            Intensity_T_dyn_minus = Intensity[6]
            Intensity_T_stat = Intensity[7]

            position_Det = -(39.7+22.4)/2 # theoretical prediction
            position = int(len(Detuning)/2 + position_Det * 2) # position index

            position , maxima_det = define_position(method, pos_of_purity , Intensity_F ,
                ↪ purity_F_array , Detuning)

            purity[i][j] = purity_F_array[position]

```

```

# write into file
f_pur = open(file_name_angle_dis+'_purity.txt','a')
f_int = open(file_name_angle_dis+'_intensity.txt','a')

75 # settings
settings = ["#_angle_fixed:_ " + str(angle_fixed), "#_displacement_fixed:_ " + str(
    ↪ displacement_fixed), "#_one_sample:_ "+str(one_sample), "#_first_fixed:_ "+str(
    ↪ first_fixed), "#_thickness_1:_ "+str(layer_thickness1), "#_thickness_2:_ " + str(
    ↪ layer_thickness2), "#_real_motion:_ "+str(motion_real), "#_method_to_find_position:_ "+
    ↪ str(method), "#_used_code:_ "+str(file_name)]

f_pur.write("%s\n" % len(settings))
f_int.write("%s\n" % len(settings))

for item in settings:
    f_pur.write("%s\n" % item)
    f_int.write("%s\n" % item)

85 # x-axis
np.savetxt(f_pur, np.array(displacement)[None], delimiter=',')
np.savetxt(f_int, np.array(displacement)[None], delimiter=',')
# y-axis
np.savetxt(f_pur, np.array(angle)[None], delimiter=',')
np.savetxt(f_int, np.array(angle)[None], delimiter=',')

pur = np.array(purity)
inten = np.array(maxima)
95 np.savetxt(f_pur, pur, delimiter = ',')
np.savetxt(f_int, inten, delimiter = ',')

# close files
f_pur.close()
f_int.close()

```

## Optimization

```

if loop_everything == True:

    max_angle1_array = [[0 for x in range(len(min_purity))] for y in range(len(thicknesses))]
    max_angle2_array = [[0 for x in range(len(min_purity))] for y in range(len(thicknesses))]
    max_dis1_array = [[0 for x in range(len(min_purity))] for y in range(len(thicknesses))]
    max_dis2_array = [[0 for x in range(len(min_purity))] for y in range(len(thicknesses))]
    max_value_array = [[0 for x in range(len(min_purity))] for y in range(len(thicknesses))]

9   for i in range(len(thicknesses)):
        print(thicknesses[i])
        print("written_into", file_name_everything)

        max_angle1 = [0 for x in range(len(min_purity))]
        max_angle2 = [0 for x in range(len(min_purity))]
        max_dis1 = [0 for x in range(len(min_purity))]
        max_dis2 = [0 for x in range(len(min_purity))]
        max_value = [0 for x in range(len(min_purity))]

19   for k in range(len(angles1)):
        for l in range(len(angles2)):
            # nice control in terminal
            end = time.time()
            current_time = round(end-start)
            current_time_form = str(datetime.timedelta(seconds=current_time))
            percentage = (k*len(angles2)+l)/(len(angles1)*len(angles2))*100
            if percentage != 0:
                remaining_time = round((100/percentage-1)*current_time)
                remaining_time_form = str(datetime.timedelta(seconds=remaining_time))

29   else:
            remaining_time_form = ''
        print('completed:_{0:.3f}%_running:_{1:6}_s_ _remaining:_{2:6}s_ \r'.format(
            ↪ percentage, current_time_form, remaining_time_form))

```

## A. Used python-scripts

```
39     # create samples
    alpha1 = angles1[k]
    alpha2 = angles2[l]
    mag_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
    mag_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

    lFe1 = define_sample(layer_thickness1, magnetisation_pol1, internal_magnetic_field
        ↪ )
    lFe2 = define_sample(layer_thickness2, magnetisation_pol2, internal_magnetic_field
        ↪ )

    for m in range(len(displacements1)):
        for n in range(len(displacements2)):
            # Calculate spectra
            Intensity, purity_F_array, purity_T, Detuning, Times = calculate_spectra(
                ↪ displacements1[m], displacements2[n])

            Intensity_F = Intensity[0]
            Intensity_F_dyn_plus = Intensity[1]
            Intensity_F_dyn_minus = Intensity[2]
            Intensity_F_stat = Intensity[3]
            Intensity_T_dyn = Intensity[4]
            Intensity_T_dyn_plus = Intensity[5]
            Intensity_T_dyn_minus = Intensity[6]
            Intensity_T_stat = Intensity[7]

            position, maxima_det = define_position(method, pos_of_purity, Intensity_F,
                ↪ purity_F_array, Detuning)

59     # find max intensity for given purity
    for o in range(len(min_purity)):
        if (Intensity_F[position] > max_value[o]) and (abs(purity_F_array[
            ↪ position]) > min_purity[o]):
            max_angle1[o] = angles1[k]
            max_angle2[o] = angles2[l]
            max_dis1[o] = displacements1[m]
            max_dis2[o] = displacements2[n]
            max_value[o] = Intensity_F[position]

69     # write into file
    f = open(file_name_everything, "a")
    for o in range(len(min_purity)):
        results = [thicknesses[i], min_purity[o], max_angle1[o], max_angle2[o], max_dis1[o],
            ↪ max_dis2[o], max_value[o]]

        results = np.array(results)

79     # actually write into file
    np.savetxt(f, results[None], delimiter=',')
    f.close()
```

Due to running long times, the optimization for two samples was parallized and done on the cluster of MPI for Nuclear Physics.

## Highly precise displacement measurement

```
3 if loop_dis_1D == True:
    one_sample = False
    print('dis_1D')
    print('saved_into', file_name_dis_1D)

    purity = [0 for x in range(len(dis_1D))]
    intensity_value = [0 for x in range(len(dis_1D))]
    intensity_plus = [0 for x in range(len(dis_1D))]
```

```

# create samples
mag-pol1 = [np.cos(alpha1_dis_1D), np.sin(alpha1_dis_1D), 0]
mag-pol2 = [np.cos(angle_dis_1D), np.sin(angle_dis_1D), 0]
lFe1 = define_sample(layer_thickness1, mag-pol1, internal_magnetic_field)
lFe2 = define_sample(layer_thickness2, mag-pol2, internal_magnetic_field)

for i in range(len(dis_1D)):
    # nice control in terminal
    end = time.time()
    current_time = round(end-start)
    current_time_form = str(datetime.timedelta(seconds=current_time))
    percentage = i/len(dis_1D) * 100
    if percentage != 0:
        remaining_time = round((100/percentage-1)*current_time)
        remaining_time_form = str(datetime.timedelta(seconds=remaining_time))
    else:
        remaining_time_form = ''
    print('completed: {0:3f}%, running: {1:6}s, remaining: {2:6}s\r'.format(percentage,
        ↪ current_time_form, remaining_time_form))

    # Calculate spectra
    Intensity, purity_F_array, purity_T, Detuning, Times = calculate_spectra(dis1_dis_1D,
        ↪ dis_1D[i])

    Intensity_F = Intensity[0]
    Intensity_F_dyn_plus = Intensity[1]
    Intensity_F_dyn_minus = Intensity[2]
    Intensity_F_stat = Intensity[3]
    Intensity_T_dyn = Intensity[4]
    Intensity_T_dyn_plus = Intensity[5]
    Intensity_T_dyn_minus = Intensity[6]
    Intensity_T_stat = Intensity[7]

    position, maxima_det = define_position(method, pos_of_purity, Intensity_F, purity_F_array,
        ↪ Detuning)

    purity[i] = purity_F_array[position]
    intensity_value[i] = Intensity_F[position]
    intensity_plus[i] = Intensity_F_dyn_plus[position]

    # minpos - maxpos
    min_pos = np.argmin(purity)
    max_pos = np.argmax(purity)
    diff = round((min_pos-max_pos)*dis_1D_step,3)
    print('\n min_pos - max_pos: ', diff)

    # plot
    plt.figure(figsize = (5,5))
    plt.plot(dis_1D, purity, label = "purity", color = 'red')
    plt.plot(dis_1D, intensity_value, label = "intensity", color = 'blue')
    plt.plot(dis_1D, intensity_plus, color = 'green')
    plt.plot([dis_1D[min_pos], dis_1D[max_pos]], [-1.1,-1.1], marker = '+', color = 'black')
    plt.text(dis_1D[(max_pos+min_pos)//2], -1, str(diff)+'-\lambda$', horizontalalignment='center'
        ↪ ')

    plt.xlabel('\Delta_z_2\left[-\lambda_\right]$')
    plt.ylabel('Purity_/Intensity[$I_0$]')

    plt.savefig(file_name_dis_1D, bbox_inches="tight")

```

## A.1.2. Time space

```

import functools
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm
import scipy.special
import pyuass

```

## A. Used python-scripts

```
7 from matplotlib.cm import get_cmap
  from matplotlib.colors import Normalize
  from scipy.optimize import curve_fit
  from scipy.signal import find_peaks
  from scipy.fftpack import fft, ifft, fftfreq, fftshift
  from scipy.integrate import.simps

  # get current filename
  import sys
17 import os
  file_name = os.path.basename(sys.argv[0])

  # measure running time
  import time
  start = time.time()

  ### PARAMETERS ###

27 #####

  # consider only one sample
  one_sample = False

  # real or ideal motion
  motion_real = False
  t_rise = 15.

37 # polarisations
  angle_coord = -np.pi * 0
  alpha1 = 1.099557 #np.pi/4.0
  alpha2 = -0.942478 #-np.pi/4.0

  # samples
  layer_thickness1 = 1e-6 # m
  layer_thickness2 = 1e-6 # m
  layer_thickness3 = 1e-12 #m (no sample)
47 internal_magnetic_field1 = 33 # Tesla
  internal_magnetic_field2 = 33 # Tesla

  displacement1 = 0.25#0.5 # times resonant wavelength
  displacement2 = 0.65#0.75

  slice_freq = -(39.7+22.4)/2# theoretical value of abs line
  #range_freq = np.arange(0,2,0.25) # gamma around slice_freq
  range_freq = [1] # gamma around slice_freq

57 #####

  ### CALCUALTIONS AND PLOTTING ###

  # polarisations
  beam_direction = [0, 0, 1]
67 beam_pol_lin = [np.cos(angle_coord), np.sin(angle_coord), 0]
  detector_pol_circ1 = +1 #(+ is left)
  detector_pol_circ2 = -1 #(+ is left)
  magnetisation_pol1 = [np.cos(alpha1 + angle_coord), np.sin(alpha1 + angle_coord), 0]
  magnetisation_pol2 = [np.cos(alpha2 + angle_coord), np.sin(alpha2 + angle_coord), 0]

  ### set up beam and detector
  Beam = pynuss.Beam(beam_direction)
  Detector1 = pynuss.Detector(Beam)
77 Detector2 = pynuss.Detector(Beam) # no polarisations
  Detector_minus = pynuss.Detector(Beam)
```

```

# set polarizations
Beam.SetLinearPolarization(beam_pol_lin)
beam_pol = str(beam_pol_lin)
Detector1.SetCircularFilter(detector_pol_circ1)
Detector_minus.SetCircularFilter(detector_pol_circ2)
detector_poll1 = str(detector_pol_circ1)
EVectorOut = [1, detector_pol_circ1*complex(0,1), 0]
87
#### define samples
def define_samples(thickness1, thickness2, thickness3, mag1, mag2):
    # sample 1
    eFe1 = pynuss.ResonantElement.fromTemplate('Fe57')
    eFe1.MagneticHyperfineField = internal_magnetic_field1
    eFe1.SetMagnetizationDirection(mag1)
    mFe1 = pynuss.Material.fromElement(eFe1)
    lFe1 = pynuss.Layer(mFe1, thickness1)

97
    # sample 2
    eFe2 = pynuss.ResonantElement.fromTemplate('Fe57')
    eFe2.MagneticHyperfineField = internal_magnetic_field2
    eFe2.SetMagnetizationDirection(mag2)
    mFe2 = pynuss.Material.fromElement(eFe2)
    lFe2 = pynuss.Layer(mFe2, thickness2)

    # sample 3 (nearly empty sample)
    lFe3 = pynuss.Layer(mFe2, thickness3)
107
    return lFe1, lFe2, lFe3

#####
### Calculate spectrum
#####

lFe1, lFe2, lFe3 = define_samples(layer_thickness1, layer_thickness2, layer_thickness3,
    ↪ magnetisation_poll1, magnetisation_poll2)

117
def MotionSmoothStep1(t):
    # Times in ns
    tRise = t_rise
    tShift = 10.
    lambda0 = 8.60254801e-11 # m
    x0 = displacement1 * lambda0
    if motion_real == True:
        return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
    else:
        return x0 * np.heaviside((t - tShift - 1), 0.5)

127
def MotionSmoothStep2(t):
    # Times in ns
    tRise = t_rise
    tShift = 10.
    lambda0 = 8.60254801e-11 # m
    x0 = displacement2 * lambda0
    if motion_real == True:
        return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
137
    else:
        return x0 * np.heaviside((t - tShift - 1), 0.5)

# set up frequency domain calculation
fw1 = pynuss.ForwardScattering(Beam, Detector1, lFe1) # normal spectrum
fw2 = pynuss.ForwardScattering(Beam, Detector1, lFe2)
fw_minus = pynuss.ForwardScattering(Beam, Detector_minus, lFe1) # spectrum with other handedness
fw4 = pynuss.ForwardScattering(Beam, Detector_minus, lFe2)
147
fw_initial = pynuss.ForwardScattering(Beam, Detector1, lFe3) # initial beam
fw_no_poll1 = pynuss.ForwardScattering(Beam, Detector2, lFe1) # no polarisations -> whole spectrum
fw_no_poll2 = pynuss.ForwardScattering(Beam, Detector2, lFe2) # no polarisations -> whole spectrum

if one_sample == True:

```

## A. Used python-scripts

```

# combine samples, detector 1 (
cs = pynuss.tools.CombineSamples(
    (fw1.TransmissionMatrix, MotionSmoothStep1), # moving sample
)
157 # combine samples, detector 3 (
cs_minus = pynuss.tools.CombineSamples(
    (fw_minus.TransmissionMatrix, MotionSmoothStep1), # moving sample
)
cs_none = pynuss.tools.CombineSamples(
    (fw_no_pol1.TransmissionMatrix, MotionSmoothStep1), # moving sample
)

else:
# combine samples, detector 1 (+)
167 cs = pynuss.tools.CombineSamples(
    (fw1.TransmissionMatrix, MotionSmoothStep1), # moving sample
    (fw2.TransmissionMatrix, MotionSmoothStep2) # moving sample
)

# combine samples, detector 3 (-)
cs_minus = pynuss.tools.CombineSamples(
    (fw_minus.TransmissionMatrix, MotionSmoothStep1), # moving sample
    (fw4.TransmissionMatrix, MotionSmoothStep2) # moving sample
)
177 # combine samples, detector 2 (none)
cs_none = pynuss.tools.CombineSamples(
    (fw_no_pol1.TransmissionMatrix, MotionSmoothStep1), # moving sample
    (fw_no_pol2.TransmissionMatrix, MotionSmoothStep2) # moving sample
)

# compute ResponseMatrix
Detuning = cs.DetuningGrid(40, 0.5, 800, 0.2)
187 DetuningStep = Detuning[1] - Detuning[0]
RM_F_stat = cs.ResponseStatic(Detuning) # static response
RM_F_dyn = cs.ResponseWithMotion(Detuning) # dynamic response
RM_F_dyn_minus = cs_minus.ResponseWithMotion(Detuning) # dynamic response
RM_initial = fw_initial.TransmissionMatrix(Detuning)
RM_no_pol = fw_no_pol1.TransmissionMatrix(Detuning)

# compute intensity
Intensity_F_stat = RM_F_stat.Intensity()
Intensity_F_dyn = RM_F_dyn.Intensity()
197 Intensity_F_dyn_minus = RM_F_dyn_minus.Intensity()
Intensity_F_initial = RM_initial.Intensity()
Intensity_F_no_pol = RM_no_pol.Intensity()
Intensity_F_initial = fw_initial.TransmissionIntensity(Detuning)

# Transform into time domain
RM_T_stat, TStep = pynuss.tools.FreqToTime(RM_F_stat, DetuningStep)
RM_T_dyn, TStep = pynuss.tools.FreqToTime(RM_F_dyn, DetuningStep)
RM_T_dyn_minus, TStep = pynuss.tools.FreqToTime(RM_F_dyn_minus, DetuningStep)
RM_T_initial, TStep = pynuss.tools.FreqToTime(RM_initial, DetuningStep)
207 RM_T_no_pol, TStep = pynuss.tools.FreqToTime(RM_no_pol, DetuningStep)
Times1 = np.arange(len(Detuning)) * (TStep)

# compute intensity in time domain
Intensity_T_stat = RM_T_stat.Intensity()
Intensity_T_dyn = RM_T_dyn.Intensity()
Intensity_T_dyn_minus = RM_T_dyn_minus.Intensity()
Intensity_T_initial = RM_T_initial.Intensity()
Intensity_T_no_pol = RM_T_no_pol.Intensity()

217 Intensity = [Intensity_F_initial, Intensity_T_initial, Intensity_F_no_pol, Intensity_T_no_pol,
    ↪ Intensity_F_stat, Intensity_T_stat, Intensity_F_dyn, Intensity_F_dyn_minus, Intensity_T_dyn
    ↪ , Intensity_T_dyn_minus]

purity_all = (Intensity_T_dyn - Intensity_T_dyn_minus)/(Intensity_T_dyn+Intensity_T_dyn_minus)

# angle between J and J+
length = np.sqrt(Intensity_T_dyn**2 + Intensity_T_dyn_minus**2)

```

```

angle = np.arccos(Intensity-T_dyn/length)

### analyze the peaks
step_time = Times1[1]-Times1[0]
227 peaks, _ = find_peaks(-purity_all[0:int(200/step_time)], prominence = 0.8, distance = 4, height =
    ↪ 0.5)
Times_peaks = np.zeros(len(peaks))
purity_peaks = np.zeros(len(peaks))
for i in range(len(peaks)):
    Times_peaks[i] = Times1[peaks[i]]
    purity_peaks[i] = purity_all[peaks[i]]

# distance between peaks
237 distance = np.diff(Times_peaks)
mean_dist = np.mean(distance)
std_dist = np.std(distance)
print('\Delta t: ', mean_dist, '\pm', std_dist)

### Fourier analysis of total spectrum

# define variables
scaling = 1/(Times1[-1]-Times1[0])*len(Times1) #time -> index
# exclude prompt pulse
247 xlim_u = int(10*scaling)
xlim_o = len(Times1)

N = len(Times1[xlim_u:xlim_o])
T = Times1[1]-Times1[0]
x = np.linspace(0.0, N*T, N)
y = Intensity-T_dyn[xlim_u:xlim_o]
y2 = (np.sin(2*np.pi/28*x-2.3*scaling)**2)
y3 = purity_all[xlim_u:xlim_o]
y4 = Intensity-T_dyn_minus[xlim_u:xlim_o]
257 # calc FT
tstep = T / 141
fstep = 2*np.pi / (len(y)*tstep)
norm = np.sqrt(2*np.pi) / fstep
yf = ifft(y)*norm
yf2 = ifft(y2)*norm
yf3 = ifft(y3)*norm
yf4 = ifft(y4)*norm
267 xf = np.arange(-N/2, N/2, 1)*fstep

# shift 0 to middle
xf = fftshift(xf)
yf = fftshift(yf)
yf2 = fftshift(yf2)
yf3 = fftshift(yf3)
yf4 = fftshift(yf4)

# plot
277 fig, axes = plt.subplots(2, figsize=(7,6.5))
axes[0].plot(x, y3, color = 'red', label = '$P$')
axes[1].plot(xf, np.abs(yf3), color = 'red', label = '$P$')
axes[1].set_xlim([-100,100])
axes[1].set_ylim([0,0.2])
axes[0].set_xlim([0,200])
axes[0].set_xlabel('Time_[ns]')
axes[0].set_ylabel('Purity')
axes[1].set_ylabel('Amplitude_[a.u.]')
axes[1].set_xlabel('Detuning_[\gamma]')
287 plt.savefig('time_spec_all', bbox_inches='tight')

#####
### Energy time spectra ###
#####

```

## A. Used python-scripts

```

297  ### create analyzer
mSS = pynuss.Material.fromChemicalFormula( '{57Fe}55Cr25Ni20', 7900)
Analyzer = pynuss.Layer(mSS, 6e-6)
fw_plus = pynuss.ForwardScattering(Beam, Detector1, Analyzer)
fw_none = pynuss.ForwardScattering(Beam, Detector2, Analyzer)
fw_minus = pynuss.ForwardScattering(Beam, Detector_minus, Analyzer)

T_plus = fw_plus.TransmissionMatrix
T_none = fw_none.TransmissionMatrix
T_minus = fw_minus.TransmissionMatrix

307  ### perform actual calculation of EnergyTimeSpectrum

# +
MBDetuningMax = 150 # gamma
MBDetuningStep = 0.25
TimeRange = 800 # ns
TimeStep = 0.5
MBDetunings, Times, RM = cs_none.EnergyTimeSpectrum(
    T_none, MBDetuningMax, MBDetuningStep, TimeRange, TimeStep)

317  RM.Beam.SetLinearPolarization([1, 0, 0])
Grid_none = RM.Intensity()

RM.Detector.SetCircularFilter(+1)
Grid_plus = RM.Intensity()

RM.Detector.SetCircularFilter(-1)
Grid_minus = RM.Intensity()

327  fig, axes = plt.subplots(2, figsize=(7,6.5))
colors = ['red', 'orange', 'lawngreen', 'green', 'blue']
### slice to have time dependence at specific frequency
for j in range(len(range_freq)):
    # slice_freq in step units of MBDetunings
    step_det = (MBDetunings[1]-MBDetunings[0])
    slice_range = np.array([slice_freq - range_freq[j], slice_freq +range_freq[j]])
    pos_slice_freq = len(MBDetunings)/2.0 + slice_range / step_det

337  Time_spec_none = Grid_none[:,int(pos_slice_freq[0])]
Time_spec_plus = Grid_plus[:,int(pos_slice_freq[0])]
Time_spec_minus = Grid_minus[:,int(pos_slice_freq[0])]

for i in range(len(pos_slice_freq)-1):
    Time_spec_none += Grid_none[:,int(pos_slice_freq[i+1])]
    Time_spec_plus += Grid_plus[:,int(pos_slice_freq[i+1])]
    Time_spec_minus += Grid_minus[:,int(pos_slice_freq[i+1])]

purity = (Time_spec_plus-Time_spec_minus)/(Time_spec_plus+Time_spec_minus)

347  # Fourier analysis of one frequency

# define variables
scaling = 1/(Times[-1]-Times[0])*len(Times) #time -> index
xlim_u = int(10*scaling)
xlim_o = len(Times)
N = len(Times[xlim_u:xlim_o])
# sample spacing
T = Times[1]-Times[0]

357  x = np.linspace(0.0, N*T, N)
y = Time_spec_plus[xlim_u:xlim_o]/Time_spec_plus[int(10*scaling)]
y2 = (np.sin(2*np.pi/28*x-2.3*scaling)**2)
y4 = Time_spec_minus[xlim_u:xlim_o]/Time_spec_minus[int(10*scaling)]
y3 = purity[xlim_u:xlim_o]

# calc FT
tstep = T / 141
fstep = 2*np.pi / (len(y)*tstep)

367  norm = np.sqrt(2*np.pi) / fstep

```

```

yf = ifft(y)*norm
yf2 = ifft(y2)*norm
yf3 = ifft(y3)*norm
yf4 = ifft(y4)*norm
xf = np.arange(-N/2, N/2, 1)*fstep

# shift 0 to middle
xf = fftshift(xf)
yf = fftshift(yf)
377 yf2 = fftshift(yf2)
yf3 = fftshift(yf3)
yf4 = fftshift(yf4)

axes[0].plot(x, y3, color = colors[j%len(colors)], label =str(range_freq[j]))

axes[1].plot(xf, np.abs(yf3), color = colors[j%len(colors)], label = str(range_freq[j]))

387 axes[1].set_xlim([-100,100])
axes[1].set_ylim([0,0.08])
axes[0].set_xlim([0,200])

axes[0].set_xlabel('Time_[ns]')
axes[0].set_ylabel('Purity')
axes[1].set_ylabel('Amplitude_[a.u.]')
axes[1].set_xlabel('Detuning_[ $\gamma$ ]')
if (len(range_freq)>1):
    axes[1].legend(loc='best')
397 plt.savefig('time_spec_freq_range', bbox_inches='tight')

fig, axes = plt.subplots(1, figsize=(7,3))
axes.semilogy(x, y, label = '$I_+$', color = 'blue')
axes.semilogy(x, y4, label = '$I_-$', color = 'green', linestyle = '--')
axes.set_xlim([0,200])

axes.set_xlabel('Time_[ns]')
axes.set_ylabel('Intensity_[ $I_0$ ]')
407 plt.savefig('time_spec_freq_int', bbox_inches='tight')

# pictures for thesis
fig = plt.figure(figsize=(7,3))
plt.semilogy(Times1, Intensity_T_dyn + Intensity_T_dyn_minus, color = 'blue', linestyle = '--')
plt.semilogy(Times1, Intensity_T_dyn_minus, color = 'green', linestyle = ':')
plt.semilogy(Times1, Intensity_T_dyn, color = 'red')

plt.xlim(0, 200)
417 plt.xlabel('Time_t_[ns]')
plt.ylabel('Intensity_[a.u.]')

plt.savefig('all_time_int', bbox_inches='tight')

fig, ax1 = plt.subplots(figsize=(7,3))
ax1.semilogy(Times1, Intensity_T_dyn + Intensity_T_dyn_minus, color = 'blue', linestyle = '--')
ax1.set_xlim(0, 200)
ax1.set_xlabel('Time_t_[ns]')
427 ax1.set_ylabel('Intensity_[a.u.]')

ax2 = ax1.twinx()
ax2.plot(Times1, purity_all, color = 'red')
ax2.set_ylabel('Purity')

plt.savefig('all_time_pur', bbox_inches='tight')

```

## A.2. Applications

With these scripts, the additional applications of our setup where calculated.

### A.2.1. Circular polarization filter

```
import functools
import numpy as np
import matplotlib.pyplot as plt
import scipy.special
import pynuss
6 from scipy.optimize import curve_fit
import datetime

# get current filename
import sys
import os
file_name = os.path.basename(sys.argv[0])

# measure running time
import time
16 start = time.time()

### PARAMETERS ###

#####

26 ### general parameters

# consider only one sample
one_sample = False

# real or ideal motion
motion_real = False
t_rise = 2. #15. experimental

36 # samples
layer_thickness1 = 1e-6 # m
layer_thickness2 = 1e-6 # m
internal_magnetic_field = 33 # Tesla

### parameters for the different parts (if not set, the paramters of "no loop" are taken)

# loops
no_loop = True

46 # no loop
alpha1 = np.pi/4.0
alpha2 = -np.pi/4.0

displacement1 = 0.5 # times resonant wavelength
displacement2 = 0.25

save_pic = "spectrum_for_thesis_polfilter_2samples"

56

#####

### CALCUALTIONS AND PLOTTING ###

### combine paramtere acoordingly for calculation
```

```

66 ##### set up beam and detector
Beam_plus = pynuss.Beam([0,0,1])
Beam_minus = pynuss.Beam([0,0,1])

# set polarizations
Beam_plus.SetCircularPolarization(+1)
Beam_minus.SetCircularPolarization(-1)

beam_direction = [0,0,1]

76 # polarisations
magnetisation_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
magnetisation_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

def define_sample(thickness, magnetization, int_field):
    eFe = pynuss.ResonantElement.fromTemplate('Fe57')
    eFe.MagneticHyperfineField = int_field
    eFe.SetMagnetizationDirection(magnetization)
86 mFe = pynuss.Material.fromElement(eFe)
lFe = pynuss.Layer(mFe, thickness)
return lFe

##### define motion
def MotionSmoothStep1(t):
    # Times in ns
    tRise = t_rise
    tShift = 10.
96 lambda0 = 8.60254801e-11 # m
x0 = displacement1 * lambda0
if motion_real == True:
    return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
else:
    return x0 * np.heaviside((t - (tShift+1)), 0.5)

def MotionSmoothStep2(t):
    # Times in ns
106 tRise = t_rise
tShift = 10.
lambda0 = 8.60254801e-11 # m
x0 = displacement2 * lambda0
if motion_real == True:
    return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
else:
    return x0 * np.heaviside((t - tShift-1), 0.5)

def Fit_gaussian(x, A, mu, sigma, offset):
116 return A*np.exp(-0.5*((x-mu)/sigma)**2) + offset

def calculate_spectra(Beam, displacement_1, displacement_2):
    # define detector
    Detector_lin = pynuss.Detector(Beam)
    Detector_lin_perp = pynuss.Detector(Beam)
    Detector_lin.SetLinearFilter([1,0,0])
    Detector_lin_perp.SetLinearFilter([0,1,0])

126 def MotionSmoothStep1(t):
    # Times in ns
    tRise = t_rise
    tShift = 10.
    lambda0 = 8.60254801e-11 # m
    x0 = displacement_1 * lambda0
    if motion_real == True:
        return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
    else:
        return x0 * np.heaviside((t - tShift-1), 0.5)

136 def MotionSmoothStep2(t):

```

## A. Used python-scripts

```

# Times in ns
tRise = t_rise
tShift = 10.
lambda0 = 8.60254801e-11 # m
x0 = displacement_2 * lambda0
if motion_real == True:
    return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
146 else:
    return x0 * np.heaviside((t - tShift - 1), 0.5)

# set up frequency domain calculation
fw1_lin = pynuss.ForwardScattering(Beam, Detector_lin, lFe1) # normal spectrum
fw2_lin = pynuss.ForwardScattering(Beam, Detector_lin, lFe2)
fw1_lin_perp = pynuss.ForwardScattering(Beam, Detector_lin_perp, lFe1) # spectrum with other
    ↪ handedness
fw2_lin_perp = pynuss.ForwardScattering(Beam, Detector_lin_perp, lFe2)

156 if one_sample == True:
    # combine samples, detector 1 (
    cs_lin = pynuss.tools.CombineSamples(
        (fw1_lin.TransmissionMatrix, MotionSmoothStep1), # moving sample
    )

    # combine samples, detector 3 (
    cs_lin_perp = pynuss.tools.CombineSamples(
        (fw1_lin_perp.TransmissionMatrix, MotionSmoothStep1), # moving sample
    )
166 else:
    # combine samples, detector 1 (
    cs_lin = pynuss.tools.CombineSamples(
        (fw1_lin.TransmissionMatrix, MotionSmoothStep1), # moving sample
        (fw2_lin.TransmissionMatrix, MotionSmoothStep2) # moving sample
    )

    # combine samples, detector 3 (
    cs_lin_perp = pynuss.tools.CombineSamples(
        (fw1_lin_perp.TransmissionMatrix, MotionSmoothStep1), # moving sample
        (fw2_lin_perp.TransmissionMatrix, MotionSmoothStep2) # moving sample
176 )

# compute ResponseMatrix
Detuning = cs_lin.DetuningGrid(100, 0.5, 800, 0.2)
DetuningStep = Detuning[1] - Detuning[0]
RM_F_stat = cs_lin.ResponseStatic(Detuning) # static response
RM_F_stat_perp = cs_lin_perp.ResponseStatic(Detuning) # static response
RM_F_dyn_lin = cs_lin.ResponseWithMotion(Detuning) # dynamic response
RM_F_dyn_lin_perp = cs_lin_perp.ResponseWithMotion(Detuning) # dynamic response
186

# compute intensity
Intensity_F_stat = RM_F_stat.Intensity() + RM_F_stat_perp.Intensity()
Intensity_F_dyn_lin = RM_F_dyn_lin.Intensity()
Intensity_F_dyn_lin_perp = RM_F_dyn_lin_perp.Intensity()

# Transform into time domain
RM_T_stat, TStep = pynuss.tools.FreqToTime(RM_F_stat, DetuningStep)
RM_T_stat_perp, TStep = pynuss.tools.FreqToTime(RM_F_stat_perp, DetuningStep)
196 RM_T_dyn_lin, TStep = pynuss.tools.FreqToTime(RM_F_dyn_lin, DetuningStep)
RM_T_dyn_lin_perp, TStep = pynuss.tools.FreqToTime(RM_F_dyn_lin_perp, DetuningStep)
Times = np.arange(len(Detuning)) * (TStep)

# compute intensity in time domain
Intensity_T_stat = RM_T_stat.Intensity() + RM_T_stat_perp.Intensity()
Intensity_T_dyn_lin = RM_T_dyn_lin.Intensity()
Intensity_T_dyn_lin_perp = RM_T_dyn_lin_perp.Intensity()

Intensity_F_dyn = Intensity_F_dyn_lin + Intensity_F_dyn_lin_perp
206 Intensity_T_dyn = Intensity_T_dyn_lin + Intensity_T_dyn_lin_perp

purity_F = (Intensity_F_dyn_lin - Intensity_F_dyn_lin_perp) / Intensity_F_dyn
purity_T = (Intensity_T_dyn_lin - Intensity_T_dyn_lin_perp) / Intensity_T_dyn

```

```

Intensity = [Intensity_F_dyn , Intensity_F_dyn_lin , Intensity_F_dyn_lin_perp , Intensity_F_stat ,
            ↪ Intensity_T_dyn , Intensity_T_dyn_lin , Intensity_T_dyn_lin_perp , Intensity_T_stat]

return Intensity , purity_F , purity_T , Detuning , Times

216 #####
    ### compute spectra , one set of parameter
    #####

if no_loop == True:
    # create samples
    lFe1 = define_sample(layer_thickness1 , magnetisation_pol1 , internal_magnetic_field)
    lFe2 = define_sample(layer_thickness2 , magnetisation_pol2 , internal_magnetic_field)

    # Calculate spectra
226 Intensity_plus , purity_F_array , purity_T , Detuning , Times = calculate_spectra(Beam_plus ,
    ↪ displacement1 , displacement2)

    Intensity_F = Intensity_plus [0]
    Intensity_F_dyn_lin = Intensity_plus [1]
    Intensity_F_dyn_lin_perp = Intensity_plus [2]
    Intensity_F_stat = Intensity_plus [3]
    Intensity_T_dyn = Intensity_plus [4]
    Intensity_T_dyn_lin = Intensity_plus [5]
    Intensity_T_dyn_lin_perp = Intensity_plus [6]
    Intensity_T_stat = Intensity_plus [7]

236 Intensity_minus , purity_F_array , purity_T , Detuning , Times = calculate_spectra(Beam_minus ,
    ↪ displacement1 , displacement2)

    Intensity_F_minus = Intensity_minus [0]
    Intensity_F_dyn_lin_minus = Intensity_minus [1]
    Intensity_F_dyn_lin_perp_minus = Intensity_minus [2]
    Intensity_F_stat_minus = Intensity_minus [3]
    Intensity_T_dyn_minus = Intensity_minus [4]
    Intensity_T_dyn_lin_minus = Intensity_minus [5]
    Intensity_T_dyn_lin_perp_minus = Intensity_minus [6]
246 Intensity_T_stat_minus = Intensity_minus [7]

    ### plot
    # spectra for thesis
    fig , axes = plt.subplots(nrows = 1 , ncols = 2 , figsize = (8.5,4))
    ax1 = axes [0]
    ax2 = axes [1]

256 ax1.set_xlabel('Detuning_[$\gamma$]')
    ax1.set_ylabel('Intensity_[$I_0$]')
    ax2.set_xlabel('Detuning_[$\gamma$]')
    ax2.set_ylabel('Intensity_[$I_0$]')

    ax1.set_xlim([-100, 100])
    ax2.set_xlim([-100, 100])

    ax1.plot(Detuning , Intensity_F_dyn_lin , label='$||$', linestyle = "—", color = "blue")
    ax1.plot(Detuning , Intensity_F_dyn_lin_perp , label='$\perp$', linestyle = ":", color = "red")

266 ax2.plot(Detuning , Intensity_F_dyn_lin_minus , label='$||$', linestyle = "—", color = "blue")
    ax2.plot(Detuning , Intensity_F_dyn_lin_perp_minus , label='$\perp$', linestyle = ":", color =
    ↪ red")

    fig.subplots_adjust(wspace=0.4)
    fig.savefig(save_pic , bbox_inches='tight')

```

## A.2.2. $\frac{\lambda}{2}$ -wave plate and light switch

```

import functools
import numpy as np

```

## A. Used python-scripts

```
import matplotlib.pyplot as plt
import scipy.special
import pynuss
from scipy.optimize import curve_fit
import datetime

9 # get current filename
import sys
import os
file_name = os.path.basename(sys.argv[0])

# measure running time
import time
start = time.time()

19
### PARAMETERS ###

#####

### general parameters

# consider only one sample
29 one_sample = True

# real or ideal motion
motion_real = False
t_rise = 2.

# samples
layer_thickness1 = 1e-6 # m
layer_thickness2 = 1e-6 # m
39 internal_magnetic_field = 33 # Tesla

### parameters for the different parts (if not set, the paramters of "no loop" are taken)

alpha1 = np.pi*0.25 #1.099557#np.pi/4.0
alpha2 = np.pi*0.5 #-0.942478#-np.pi/4.0

displacement1 = 0.5 # times resonant wavelength
displacement2 = 1

49 save_pic = "spectrum_for_thesis_lambda2_one"

#####

### CALCUALTIONS AND PLOTTING ###

59 ### combine paramtere acoordingly for calculation

### set up beam and detector
Beam = pynuss.Beam([0,0,1])

# set polarizations
Beam.SetLinearPolarization([1,0,0])
beam_direction = [0,0,1]

69 magnetisation_pol1 = [np.cos(alpha1), np.sin(alpha1), 0]
magnetisation_pol2 = [np.cos(alpha2), np.sin(alpha2), 0]

# define samples
def define_sample(thickness, magnetization, int_field):
    eFe = pynuss.ResonantElement.fromTemplate('Fe57')
    eFe.MagneticHyperfineField = int_field
```

```

eFe.SetMagnetizationDirection(magnetization)
mFe = pynuss.Material.fromElement(eFe)
lFe = pynuss.Layer(mFe, thickness)
79 return lFe

def Fit_gaussian(x, A, mu, sigma, offset):
    return A*np.exp(-0.5*((x-mu)/sigma)**2) + offset

def calculate_spectra(Beam, displacement_1, displacement_2):
    # define detector
    Detector_lin = pynuss.Detector(Beam)
    Detector_lin_perp = pynuss.Detector(Beam)
    89 Detector_lin.SetLinearFilter([1,0,0])
    Detector_lin_perp.SetLinearFilter([0,1,0])

    def MotionSmoothStep1(t):
        # Times in ns
        tRise = t_rise
        tShift = 10.
        lambda0 = 8.60254801e-11 # m
        x0 = displacement_1 * lambda0
        99 if motion_real == True:
            return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
        else:
            return x0 * np.heaviside((t - tShift -1), 0.5)

    def MotionSmoothStep2(t):
        # Times in ns
        tRise = t_rise
        tShift = 10.
        lambda0 = 8.60254801e-11 # m
        109 x0 = displacement_2 * lambda0
        if motion_real == True:
            return x0 * (1 + scipy.special.erf((t - tShift) / tRise)) / 2.
        else:
            return x0 * np.heaviside((t - tShift -1), 0.5)

    # set up frequency domain calculation
    fw1_lin = pynuss.ForwardScattering(Beam, Detector_lin, lFe1) # normal spectrum
    fw2_lin = pynuss.ForwardScattering(Beam, Detector_lin, lFe2)
    119 fw1_lin_perp = pynuss.ForwardScattering(Beam, Detector_lin_perp, lFe1) # spectrum with other
        ↪ handedness
    fw2_lin_perp = pynuss.ForwardScattering(Beam, Detector_lin_perp, lFe2)

    if one_sample == True:
        # combine samples, detector 1 (
        cs_lin = pynuss.tools.CombineSamples(
            (fw1_lin.TransmissionMatrix, MotionSmoothStep1), # moving sample
        )

        # combine samples, detector 3 (
        129 cs_lin_perp = pynuss.tools.CombineSamples(
            (fw1_lin_perp.TransmissionMatrix, MotionSmoothStep1), # moving sample
        )
    else:
        # combine samples, detector 1 (
        cs_lin = pynuss.tools.CombineSamples(
            (fw1_lin.TransmissionMatrix, MotionSmoothStep1), # moving sample
            (fw2_lin.TransmissionMatrix, MotionSmoothStep2) # moving sample
        )

        # combine samples, detector 3 (
        139 cs_lin_perp = pynuss.tools.CombineSamples(
            (fw1_lin_perp.TransmissionMatrix, MotionSmoothStep1), # moving sample
            (fw2_lin_perp.TransmissionMatrix, MotionSmoothStep2) # moving sample
        )

    # compute ResponseMatrix
    Detuning = cs_lin.DetuningGrid(100, 0.5, 800, 0.2)

```

## A. Used python-scripts

```

149 DetuningStep = Detuning[1] - Detuning[0]
    RM_F_stat = cs_lin.ResponseStatic(Detuning) # static response
    RM_F_stat_perp = cs_lin_perp.ResponseStatic(Detuning) # static response
    RM_F_dyn_lin = cs_lin.ResponseWithMotion(Detuning) # dynamic response
    RM_F_dyn_lin_perp = cs_lin_perp.ResponseWithMotion(Detuning) # dynamic response

    # compute intensity
    Intensity_F_stat = RM_F_stat.Intensity() + RM_F_stat_perp.Intensity()
    Intensity_F_dyn_lin = RM_F_dyn_lin.Intensity()
    Intensity_F_dyn_lin_perp = RM_F_dyn_lin_perp.Intensity()

159 # Transform into time domain
    RM_T_stat, TStep = pynuss.tools.FreqToTime(RM_F_stat, DetuningStep)
    RM_T_stat_perp, TStep = pynuss.tools.FreqToTime(RM_F_stat_perp, DetuningStep)
    RM_T_dyn_lin, TStep = pynuss.tools.FreqToTime(RM_F_dyn_lin, DetuningStep)
    RM_T_dyn_lin_perp, TStep = pynuss.tools.FreqToTime(RM_F_dyn_lin_perp, DetuningStep)
    Times = np.arange(len(Detuning)) * (TStep)

    # compute intensity in time domain
    Intensity_T_stat = RM_T_stat.Intensity() + RM_T_stat_perp.Intensity()
    Intensity_T_dyn_lin = RM_T_dyn_lin.Intensity()
169 Intensity_T_dyn_lin_perp = RM_T_dyn_lin_perp.Intensity()
    Intensity_F_dyn = Intensity_F_dyn_lin + Intensity_F_dyn_lin_perp
    Intensity_T_dyn = Intensity_T_dyn_lin + Intensity_T_dyn_lin_perp

    purity_F = (Intensity_F_dyn_lin - Intensity_F_dyn_lin_perp)/Intensity_F_dyn
    purity_T = (Intensity_T_dyn_lin - Intensity_T_dyn_lin_perp)/Intensity_T_dyn

    Intensity = [Intensity_F_dyn, Intensity_F_dyn_lin, Intensity_F_dyn_lin_perp, Intensity_F_stat,
    ↪ Intensity_T_dyn, Intensity_T_dyn_lin, Intensity_T_dyn_lin_perp, Intensity_T_stat]

179 return Intensity, purity_F, purity_T, Detuning, Times

#####
### compute spectra, one set of parameter
#####

# create samples
lFe1 = define_sample(layer_thickness1, magnetisation_poll, internal_magnetic_field)
lFe2 = define_sample(layer_thickness2, magnetisation_pol2, internal_magnetic_field)
189 # Calculate spectra
Intensity, purity_F_array, purity_T, Detuning, Times = calculate_spectra(Beam, displacement1,
    ↪ displacement2)

Intensity_F = Intensity[0]
Intensity_F_dyn_lin = Intensity[1]
Intensity_F_dyn_lin_perp = Intensity[2]
Intensity_F_stat = Intensity[3]
Intensity_T_dyn = Intensity[4]
Intensity_T_dyn_lin = Intensity[5]
199 Intensity_T_dyn_lin_perp = Intensity[6]
Intensity_T_stat = Intensity[7]

### plot
# spectra for thesis
fig, axes = plt.subplots(figsize = (3.5,5))
ax1 = axes

ax1.set_xlabel('Detuning_[$\gamma$]')
209 ax1.set_ylabel('Intensity_[$I_0$]')

ax1.set_xlim([-100, 100])

ax1.plot(Detuning, Intensity_F_dyn_lin, linestyle = ":", color = "blue")
ax1.plot(Detuning, Intensity_F_dyn_lin_perp, linestyle = "-", color = "green")

fig.savefig(save_pic, bbox_inches='tight')

```

**Highly precise displacement measurement**

See appendix A.1.1.

# List of Figures

2.1. Schematic of wave plates. . . . .	5
2.2. Dependency of one wave plate on angle and phase shift . . . . .	8
2.3. Dependency of two wave plates on phase shift . . . . .	10
2.4. Dependence of two wave plates on angles . . . . .	11
3.1. Level scheme of $^{57}\text{Fe}$ . . . . .	13
3.2. Spectrum of static sample . . . . .	15
3.3. Fano- and Lorentz-lines . . . . .	17
3.4. Fano- and Lorentz line combined . . . . .	18
3.5. Analytical time spectrum . . . . .	19
3.6. Setup of new approach of polarization transformation . . . . .	20
4.1. Spectrum of an unmoved sample . . . . .	25
4.2. Numerical spectrum of one sample . . . . .	26
4.3. Different kinds of motion . . . . .	27
4.4. Spectra for real and ideal motion . . . . .	28
4.5. Numerical spectra for different sample thicknesses . . . . .	29
4.6. Effect of magnetic angle and displacement for one sample . . . . .	30
4.7. Spectra for configuration of two samples . . . . .	32
4.8. Effect of different displacements for two sample . . . . .	33
4.9. Effect of different angles for two sample . . . . .	34
4.10. Optimization . . . . .	36
4.11. Spectrum for optimized parameter . . . . .	36
4.12. Time spectrum, all frequencies . . . . .	38
4.13. Fourier analysis of purity, all frequencies . . . . .	38
4.14. Time spectrum, range of frequencies . . . . .	40
4.15. Fourier analysis of purity, range of frequencies . . . . .	40
4.16. Circular polarization filter . . . . .	41
4.17. $\frac{\lambda}{2}$ -wave plate . . . . .	42
4.18. Light switch . . . . .	43
4.19. Precise displacement measurement . . . . .	44

# List of Tables

2.1. Effect of wave plates . . . . .	7
3.1. Properties of the nuclear transitions of $^{57}\text{Fe}$ . . . . .	13
3.2. Polarizations after one sample in the new setup . . . . .	22
4.1. Standard parameters for numerical calculation . . . . .	25
4.2. Optimized parameters . . . . .	35

# Bibliography

- [1] K. P. Heeg, A. Kaldun, C. Strohm, P. Reiser, C. Ott, R. Subramanian, D. Lentrodt, J. Haber, H.-C. Wille, S. Goerttler, R. Ruffer, C. H. Keitel, R. Röhlsberger, T. Pfeifer, and J. Evers. Spectral narrowing of x-ray pulses for precision spectroscopy with nuclear resonances. *Science*, 357(6349):375–378, 2017.
- [2] A.C. Thompson. *X-ray Data Booklet*. Lawrence Berkeley National Laboratory, University of California, 2009.
- [3] Takaya Mitsui, Yasuhiko Imai, Ryo Masuda, Makoto Seto, and Ko Mibu. 57fe polarization-dependent synchrotron mössbauer spectroscopy using a diamond phase plate and an iron borate nuclear bragg monochromator. *Journal of synchrotron radiation*, 22(2):427–435, 2015.
- [4] Patrick Reiser. Time domain control of x-ray quantum dynamics. Master’s thesis, University of Heidelberg, 2014.
- [5] Kilian Peter Heeg. *X-Ray Quantum Optics With Möessbauer Nuclei In Thin-Film Cavities*. PhD thesis, University of Heidelberg, 2014.
- [6] Wolfgang Demtröder. *Experimentalphysik 2*. SpringerLink : Bücher. Springer Spektrum, Berlin, Heidelberg, 7. aufl. 2017 edition, 2017.
- [7] Eugene Hecht. *Optics*. Pearson, Boston ; New York ; Amsterdam ; Munich, 5 ed, global edition edition, 2017.
- [8] Ralf Röhlsberger. *Nuclear condensed matter physics with synchrotron radiation*. Number ARRAY(0x55d8d634d4f0) in Springer tracts in modern physics. Springer, Berlin ; Heidelberg [u.a.], 2004.
- [9] DESY. Machine parameters petra iii (design values). [http://photon-science.desy.de/facilities/petra\\_iii/machine/parameters/index\\_eng.html](http://photon-science.desy.de/facilities/petra_iii/machine/parameters/index_eng.html). Accessed: 20.06.2019.
- [10] GV Smirnov. Nuclear resonant scattering of synchrotron radiation. *Hyperfine Interactions*, 97(1):551–588, 1996.
- [11] Wolfgang Sturhahn. Conuss. <https://www.nrxs.com/conuss.html>. Accessed: 20.06.2019.
- [12] Hendrik Bernhardt, Berit Marx-Glowna, Kai S Schulze, Benjamin Grabiger, Johann Haber, Carsten Detlefs, Robert Loetzsch, Tino Kämpfer, Ralf Röhlsberger, Eckhart Förster, et al. High purity x-ray polarimetry with single-crystal diamonds. *Applied Physics Letters*, 109(12):121106, 2016.
- [13] Thorlabs. Achromatic wave plates. [https://www.thorlabs.com/navigation.cfm?guide\\_id=2396](https://www.thorlabs.com/navigation.cfm?guide_id=2396). Accessed: 15.06.2019.

- [14] Thorlabs. Pockels cell. [https://www.thorlabs.com/newgrouppage9.cfm?objectgroup\\_id=6149](https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=6149). Accessed: 15.06.2019.



# Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 24. Juni 2019