PeldorFit 2022

User Manual

Dinar Abdullin

Pablo Rauh Corro

Olav Schiemann

Bonn 2022

Table of contents

1 Introduction

1.1 What is it for?

Electron paramagnetic resonance spectroscopy (EPR) provides several pulsed techniques for measuring the strength of dipolar coupling between electron spin centers. These techniques include pulsed electron-electron double resonance (PELDOR or DEER, Figure 1.1),[1–6] relaxation-induced dipolar modulation enhancement (RIDME, Figure 1.1),[7–9] double quantum coherence EPR (DQC),[10] and single-frequency technique for refocusing dipolar couplings (SIFTER)[11,12]. Together, they are denoted as pulsed EPR dipolar spectroscopy (PDS).[13–16] In PDS, the strength of dipolar coupling is determined in frequency units, as a dipolar frequency. For two electron spin centers with isotropic g-factors g_A and g_B, respectively, the dipolar frequency is given by

$$\nu_{dd} = \frac{\mu_0 \beta_e^2 g_A g_B}{4\pi h} \cdot \frac{(1 - 3\cos^2\theta)}{r^3}, \tag{1.1}$$

where μ_0 is the vacuum permeability, β_e is the Bohr magneton, h is the Planck constant, r is the inter-spin distance, and θ is the angle between the inter-spin vector $\vec{r}$ and the applied static magnetic field $\vec{B}_0$ (Figure 1.1a). Since the dipolar frequency is inversely proportional to the third power of the distance between spin centers, PDS can be used to measure inter-spin distances. To date, such distance measurements are widely applied in structural biology to obtain secondary structure restraints in the range of 15-160 Å.

The PDS measurements yield a signal in the form of a time trace modulated by dipolar frequencies. Since the majority of PDS experiments are done on disordered samples, such as frozen solutions or powders, the time trace is usually modulated by many dipolar frequencies rather than a single dipolar frequency, because of the distribution of r and θ in a sample (see Equation 1.1 and Figure 1.1d). The distribution of dipolar frequencies, called a dipolar spectrum, can be readily determined from a PDS time trace by applying the Fourier transform. However, it is usually the inter-spin distance distribution, rather than the dipolar spectrum, that is of primary interest to PDS users. Extraction of the distance distributions from the PDS time traces, the procedure called PDS data analysis, is challenging because it is an ill-posed problem. Several approaches to tackle this problem have been proposed, some of which have been implemented in the software packages DeerAnalysis,[17] DeerNet,[18] DeerLab,[19] DD,[20] and LongDistances[21]. Most of these programs use the following assumptions:

1) The PDS time trace is averaged over all possible orientations the inter-spin vector $\vec{r}$ with respect to the applied static magnetic field $\vec{B}_0$, i.e., no orientation selectivity.

2) The electron spin centers have nearly isotropic g-factors, i.e., $g_A \approx g_B \approx 2$ (Equation 1.1).
3) The exchange interaction between electron spin centers is negligibly small (Exception: DeerNet and DeerLab).
4) The PDS time trace is acquired on a two-spin system (Exception: DeerAnalysis and DeerLab).

These assumptions are usually satisfied for biomolecules labeled with two flexible organic spin labels, such as nitroxide spin labels.[22] However, there are cases where the above assumptions are violated. For example, assumption 1) is often violated for rigid or motionally restricted spin labels,[23–27] as well as for organic[28,29] and metal[30–35] cofactors that are tightly bound to biomolecules. For the latter spin centers, the directions of $\vec{r}$ correlate with the directions of the spin centers, and the microwave pulses excite only certain directions of the spin centers with respect to $\vec{B}_0$. As a result, the corresponding PDS time trace depends on the mutual orientations the electron spin centers, and in order to extract the distance distribution from such time traces, these orientations have to be taken into account in the PDS data analysis.[31,36–42]

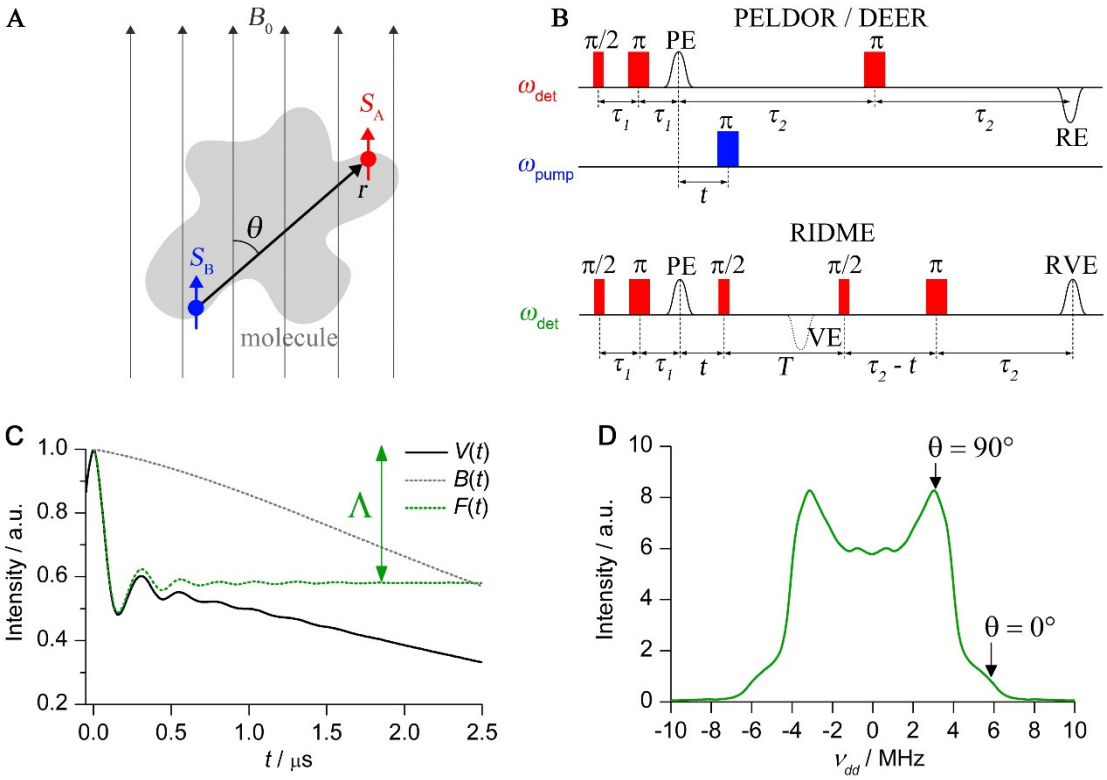

Figure 1.1. General concepts of PDS spectroscopy: A) The model of a spin pair in the applied static magnetic field. The parameters r and θ are explained in the figure. B) Four-pulse PELDOR and five-pulse RIDME pulse sequences. C) The example of a PDS time trace $V(t)$, its background contribution $B(t)$, and its dipolar contribution $F(t)$. The arrow and the symbol Λ denote the modulation depth parameter. D) The example of a dipolar spectrum obtained after the FFT of $D(t)$. The dipolar frequency regions that correspond to $\theta = 0°$ and $\theta = 90°$ are marked by arrows.

Assumption 2) is violated for spin systems in which at least one of the electron spin centers has large g-factor anisotropy, like Fe^{3+} ions or iron-sulfur clusters.[43–45] For such spin systems, Equation 1.1 does not hold and has to be replaced by the equation, in which ν_{dd} depends on the principal values of each anisotropic g-factor and the orientation of each anisotropic g-factor with respect to $\vec{r}$.[43,46]

Assumption 3) is violated for spin systems with non-negligible exchange interaction. Although significant exchange interaction is quite rare in spin systems with the inter-spin distance above 1.5 nm, there are still exceptions, most of which correspond to artificially synthesized biradicals.[38,47–50] If exchange coupling constant J corresponds to the weak coupling limit, the corresponding PDS time traces are modulated by the sum frequencies ($\nu_{dd} + J$).[51] To extract the distance distributions from the corresponding PDS time traces, J and sometimes its distribution have to be included in the PDS data analysis.[38,47,48] If exchange coupling constant J corresponds to the strong coupling limit, all dipolar frequencies scaled by factor 3/2, i.e. $3/2\ \nu_{dd}$,[51] but the corresponding PDS time traces do not depend on J.[49] Finally, if exchange coupling constant J corresponds to the intermediate coupling limit, the effect of J on the PDS time trace can not be described by simple expressions, which makes the extraction of the distance distribution difficult.[50,51]

Assumption 4) is violated for multi-spin systems, e.g., multimeric proteins consisting of three or more identical monomers, each containing one spin label or one paramagnetic cofactor.[52–55] In such spin systems, the PDS time traces are given by the product of two-spin contributions and are modulated not only by dipolar frequencies corresponding to the different spin pairs, but also by the combination frequencies corresponding to the sums and differences of the dipolar frequencies. Since the number of dipolar frequencies and the corresponding combination frequencies increases rapidly with the number of spin centers in the spin system, the complexity of the PDS data analysis increases dramatically with the number of spin centers.[52–58]

PeldorFit is a software that facilities the PDS data analysis in the challenging cases described above. The first version of the program have been developed in 2015[40] and enabled the analysis of orientation-selective PELDOR time traces, i.e. the time traces for which assumption 1) is violated. The core of this analysis was the model-based global fitting of orientation-selective time traces. The current version of the program is extended to RIDME time traces as well the PELDOR time traces acquired with chirp pulses. Moreover, in contrast to the previous version, the new version of PeldorFit does not require any pre-processing (e.g., phase correction, zero point correction, normalization) and preliminary background correction. Instead, the pre-processing is done automatically, while the background correction is included in the main

fitting. Furthermore, PeldorFit was extended to spin systems for which assumptions 2)-4) are violated. In particular, PeldorFit supports all types of spin-1/2 centers, including the ones with strongly anisotropic *g*-factors. The routine that allows considering anisotropic *g*-factors in the PDS data analysis was adapted from the program AnisoDipFit[46] (initially called DipFit[44,45]). Starting from the first version, PeldorFit is applicable to two-spin systems that have a weak Heisenberg exchange coupling. The exchange coupling constant and even its distribution can be included in the fitting of the PDS time traces. Finally, PeldorFit can now be applied to spin systems with more than two spins (multi-spin systems). Since the PeldorFit is model-based, the data analysis for multi-spin systems differs from the data analysis for two-spin systems only by increased number of model parameters.

1.2 Models of a spin system

PeldorFit performs the fitting of the PDS time traces without any prior knowledge of the spin system geometry. Only the number of spins in the spin system needs to be known in advance. Since there is an infinite number of possible spin system geometries, it is impossible to determine the exact geometric model of a spin system, and some simplifying assumptions have to be made. Three different cases of such simplifying assumptions are considered below.

1.2.1 Two-spin model

A simplest model of a two-spin system is depicted in Figure 1.2a. This model consists of two coordinate frames associated with the *g*-tensor principle axes of the two spins denoted here as spin A and spin B. The reference coordinate system of the model is set to coincide with the *g*-tensor principle axes of spin A. The orientation of the *g*-tensor principle axes corresponding to spin B is described by three Euler angles α, β, and γ. By default, the Euler angles are defined according to the *ZXZ* convention, but the user can choose other conventions if necessary. A vector connecting the origins of these two frames defines the inter-spin vector. It is described by three spherical coordinates, namely a length *r,* a polar angle ξ, and an azimuthal angle φ. To account for the conformational flexibility of the spin system, geometric parameters *r,* ξ, φ, α, β, and γ are allowed to have distributions. In the simplest case, all distributions are approximated by either a uniform distribution (Figure 1.3a) or a Gaussian distribution (Figure 1.3b). This allows to describe each distribution $P(p)$, where $p = r, \xi, \varphi, \alpha, \beta$, or γ, by only two parameters, a mean value $\langle p \rangle$ and a width Δp. In the case of the Gaussian distribution, the width parameter is given by a standard deviation. For the angular parameters, Gaussian distribution is replaced by an analogous von Mises distribution, which is described by the same

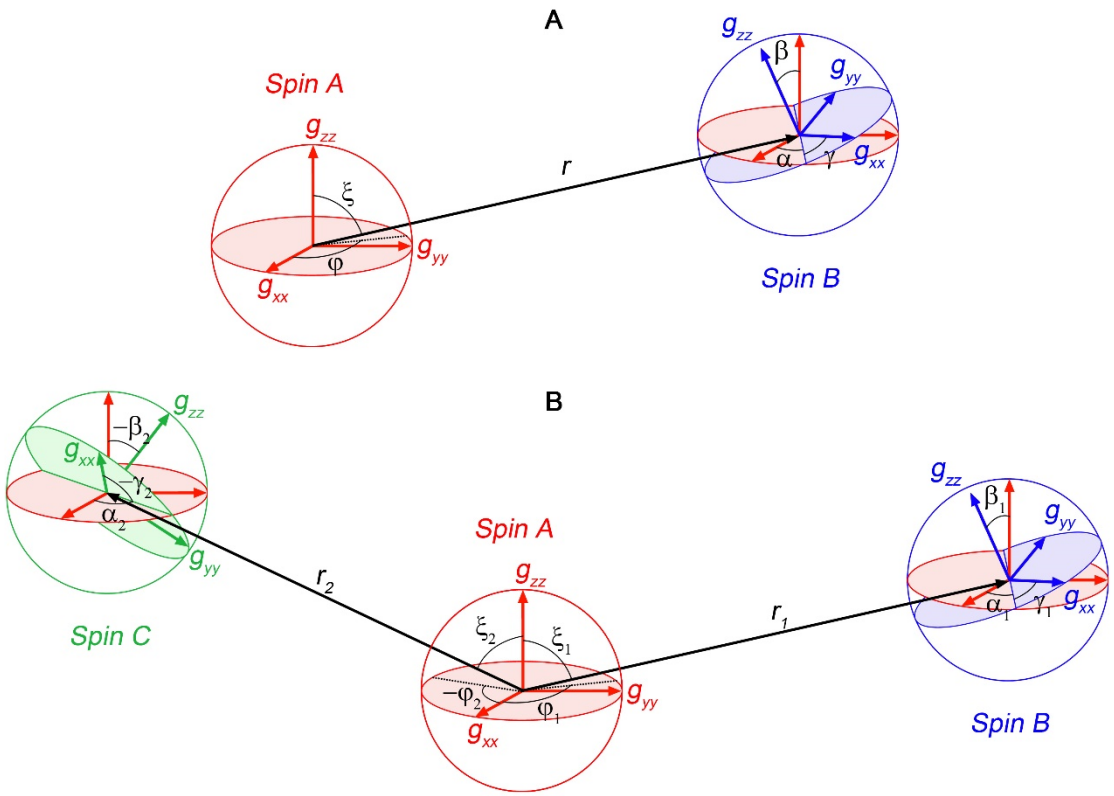

Figure 1.2. PeldorFit models of A) two-spin and B) three-spin systems.

mean value and the same standard deviation as the Gaussian distribution, but takes into account the periodicity of the angles. Since information about possible correlations between the geometric parameters is usually not available, it is assumed that the correlations between all geometric parameters are zero. Thus, the model of a two-spin system with the uncorrelated, unimodally distributed geometric parameters is described by six mean values, $\langle r \rangle$, $\langle \xi \rangle$, $\langle \varphi \rangle$, $\langle \alpha \rangle$, $\langle \beta \rangle$, and $\langle \gamma \rangle$, and six widths, Δr, $\Delta \xi$, $\Delta \varphi$, $\Delta \alpha$, $\Delta \beta$, and $\Delta \gamma$.

1.2.2 Two-spin model with multiple states

Some two-spin systems may have two or more states, e.g., because of several distinct conformational sub-ensembles of a spin label attached to a biomolecule.[59] For such cases, PeldorFit allows the distributions of geometric parameters r, ξ, φ, α, β, and γ to be approximated by multimodal uniform or Gaussian distributions (Figure 1.3c,d). Each mode of the multimodal distributions corresponds to a particular state of the spin system and is described by three parameters, namely a mean value, a width, and a relative weight w. Consequently, the model of two-spin systems with n states is described by $6n$ means, $6n$ widths, and $(n - 1)$ relative weights.

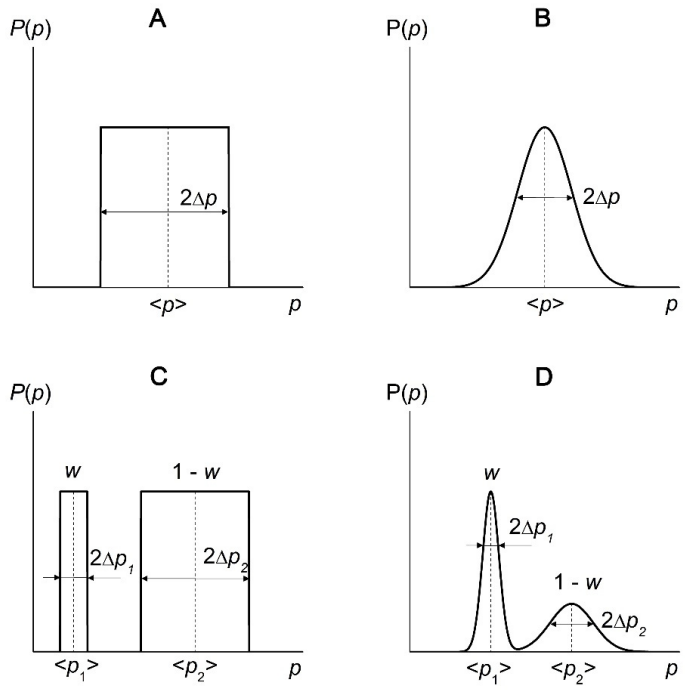

Figure 1.3. The distributions used for the PeldorFit parameters. A) Uniform distribution with a mean value $\langle p \rangle$ and a width Δp. B) Gaussian distribution with the mean value $\langle p \rangle$ and the standard deviation Δp. C) Bimodal Gaussian distribution with the mean values $\langle p_1 \rangle$ and $\langle p_2 \rangle$, the standard deviations Δp_1 and Δp_2, and a weight w.

1.2.3 Multi-spin model

So far, two models of a two-spin system were considered. Next, the models of spin systems with three or more electron spin centers have to be discussed. To build the model of a three-spin system, the model of a two-spin system is taken as a basis and extended by an additional spin, e.g., spin C. Since the reference coordinate system of the model has been set to coincide with the *g*-tensor principle axes of spin A, the position and orientation of the *g*-tensor principle axes of spin C are described in the same way as in the case of spin B. Thus, in addition to the six geometric parameters used for two-spin systems, six additional geometric parameters have to be included. To distinguish the two sets of geometric parameters r, ξ, φ, α, β, and γ, they are labelled with the indices "1" and "2" for spins B and C, respectively (Figure 1.2b). As in the case of the two-spin model, each geometric parameter can have a distribution approximated by a uniform distribution or a Gaussian distribution (Figure 1.3), and the correlation between individual geometric parameter is set to zero. Further extension of the spin system to 4 or more spins is done analogously to above. Thus, the model of an m-spin system with the uncorrelated, unimodally distributed geometric parameters is described by $6(m - 1)$ mean values and $6(m - 1)$ widths.

1.3 Geometric considerations

There are two different ways to assign spins A and B in the two-spin system and nine different ways to assign spins A, B, and C in the three-spin system. The values of the geometric parameters describing the model depend on how the assignment of the spins was made. Therefore, for the two-spin system, two equivalent sets of parameters r, ξ, φ, α, β, and γ describe the same geometry. Similarly, the geometry of the three-spin system can be described by nine equivalent sets of parameters r_1, ξ_1, φ_1, α_1, β_1, γ_1, r_2, ξ_2, φ_2, α_1, β_1, and γ_1 (Figure 1.4). PeldorFit handles only one set of geometric parameters and assumes that the user can readily derive all equivalent parameter sets.

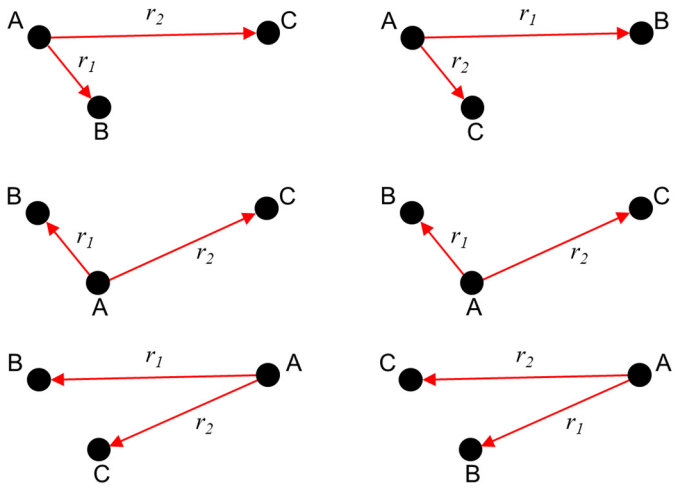

Figure 1.4. Nine different ways to assign spins A, B, and C in the tree-spin system and the corresponding nine equivalent sets of parameters r_1 and r_2.

1.4 Symmetry considerations

For simplicity, we will focus on two-spin systems. There are at least two different types of symmetry relevant to the analysis of PDS time traces. The first is the symmetry of the magnetic tensors (g and hyperfine tensors) of the electron spin centers. This symmetry is determined by the electronic configuration of the individual electron spin centers and their coordination in a particular molecule. Depending on this symmetry, all five angular parameters (ξ, φ, α, β, γ) or only a subset of them are required to simulate PDS time traces. A summary of the different spin center symmetries and the corresponding angular parameters is given in Table 1.1. For example, if the magnetic tensors of both spins are rhombic, the full set of angles is required. In contrast, three angles are sufficient if the magnetic tensors of both spins are axial. In this case, angles φ

Table 1.1. Number and ranges of the angular parameters ξ, φ, α, β, and γ required to simulate the PELDOR time traces in the cases of isotropic, axial, and rhombic spins A and B.

Spin A	**Spin B**	ξ	φ	α	β	γ
isotropic	isotropic	-	-	-	-	-
axial	isotropic	[0°, 90°]	-	-	-	-
axial	axial	[0°, 90°]	-	[0°, 90°]	[0°, 90°]	-
rhombic	isotropic	[0°, 90°]	[0°, 90°]	-	-	-
rhombic	axial	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]	-
rhombic	rhombic	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]

and γ can be excluded from consideration. In general, when the magnetic tensors of spins A and B are axial or isotropic, the number of fitting parameters in PeldorFit can be reduced.

The second type of symmetry is the inversion symmetry of the *g*-tensors. Due to this symmetry, the PDS time traces remain invariant when one of the *g*-tensor axes of spin A or B is rotated by 180°. A total of 16 combinations of such axis rotations are possible. Thus, for each set of angles (ξ, φ, α, β, γ), there are 15 symmetry-related sets of angles that cannot be distinguished in the analysis of the PDS time traces. To account for this, PeldorFit computes all 16 symmetry-related sets of angles in each fitting run.

1.5 Block diagram of the program

The block diagram of PeldorFit is depicted in Figure 1.5. PeldorFit uses a configuration file as an input. The configuration file contains all the information required for a particular PDS data analysis. The description of this file can be found in Chapter 4. The first block of the program is responsible for reading the configuration file and importing the input data into the program memory. Experimental PDS time traces are always among the input data. They are preprocessed in the second block of the program. The preprocessing includes the following steps:

1) The phase of the in-phase and quadrature components of the PDS time traces is adjusted to bring the entire signal into the in-phase component. After such adjustment, the quadrature component should look like a horizontal line. In practice, slight nonlinearity can be sometimes observed for the quadrature component, especially, at the beginning of the time trace. Usually, this has technical reasons, e.g., small phase drifts during the PDS experiment.
2) The time axis (the abscissa) of the PDS time traces is shifted to set the correct zero time. The zero time is determined using the method based on the calculation of the first moment.[17] In this method, the PDS time trace is assumed to be symmetric about the zero time and therefore its first moment has a minimum at the zero point.
3) The in-phase component of the PDS time trace is normalized to 1.
4) The noise level of each PDS time trace (the standard deviation of the noise, σ_N) is estimated using its quadrature component.[60] To extract noise from the quadrature time trace, the last 2/3 of this time trace is fitted by a linear function and then the fit is subtracted from the time trace. The first 1/3 of the quadrature time trace is excluded for the reasons discussed in 1).

All these steps are performed automatically. After the preprocessing, the in-phase components of the PDS time traces, hereafter called just the PDS time traces, are used for either simulation or fitting. The user can select only one of these two options in the configuration file.

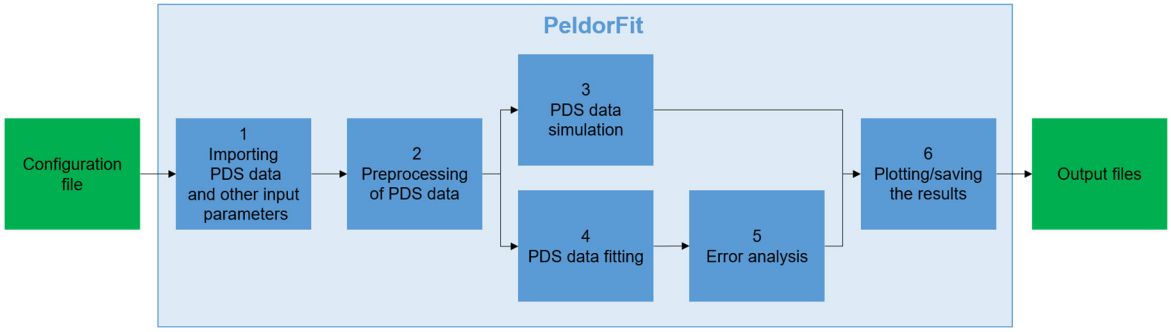

Figure 1.5. The block diagram of PeldorFit.

The simulation of the PDS time traces is performed by the third block of the program. Here, the PDS time traces are simulated using a user-defined model of the spin system and user-defined parameters of the model. The theory and technical aspects of the simulations are described in Chapters 1.6 and 1.7.

The fourth block of the program deals with the fitting of the PDS time traces. As the name suggests, this block searches for the global fit to the PDS time traces by optimizing the parameters of a user-defined model of the spin system. The details of the fitting procedure can be found in Chapter 1.8. The fitting block is followed by the error analysis block (sixth block) that allows to estimate the errors of the optimized fitting parameters. The details of the error analysis are described in Chapter 1.9.

Finally, the output data of the simulation, fitting, or error analysis are gathered in the sixth block of program, which is responsible for the graphical representation and saving of this data. The description of the output data can be found in Chapter 5.

1.6 Mathematical model of the PDS time trace

The simulation of the PDS time traces is based on the theory originally developed for PELDOR time traces.[13,61] This theory considers a disordered sample containing a large number of molecules with a particular spin system. The PDS time trace of this sample, $V(t)$, is given by the product of two contributions, a gradually decaying background, $B(t)$, and an form factor, $F(t)$:

$$V(t) = B(t) \cdot F(t). \tag{1.2}$$

In PELDOR, $B(t)$ describes the time decay of the signal caused by the dipole-dipole interaction between electron spin centers located on different particles/molecules. Therefore, $B(t)$ is often called the intermolecular component of the PELDOR time traces. The shape of $B(t)$ depends on the spatial distribution of electron spin centers in the sample. In the case of a homogeneous distribution, $B(t)$ is a monoexponential decay function,[62]

$$B(t) = \exp(-kt), \tag{1.3}$$

where k quantifies the density of the electron spin centers in the sample. Deviations from the homogeneous distribution lead to $B(t)$ which is better described by a stretched exponential function,[63]

$$B(t) = \exp\left(-kt^{d/3}\right), \tag{1.4}$$

where d is the so-called dimensionality factor. Note that when $d = 3$, Equation 1.3 transforms into Equation 1.2.

In RIDME, the shape of $B(t)$ is affected by spin diffusion, which makes it more complicated compared to the PELDOR background. Keller et al. derived a theoretical expression for the five-pulse RIDME background that is given by the product of monoexponential and Gaussian decays.[64] However, experimental studies emphasized that the background decay is often better described by one or two stretched exponential decays.[64] In addition to the stretched exponential functions, 2nd, 3rd, and 4th order polynomial functions have been shown to provide adequate fits to RIDME backgrounds.[9,44,45,65]

$F(t)$ describes the modulation of the measured echo intensity by the dipolar frequencies which correspond to the dipole-dipole interaction between the electron spin centers within individual molecules. Therefore, $F(t)$ is often called the intramolecular component of the PDS time traces. $F(t)$ depends on the number of spins in the spin system. In the case of two-spin systems, it can be described by the following equation:

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{4\pi}\int_0^{+\infty} P(r)dr \int_0^{\pi} P(\xi)\,sin(\xi)d\xi \int_0^{2\pi} P(\varphi)d\varphi \int_0^{2\pi} P(\alpha)d\alpha \times$$
$$\times \int_0^{\pi} P(\beta)\sin(\beta)d\beta \int_0^{2\pi} P(\gamma)\,d\gamma \int_0^{\pi} \sin(\xi_{B_0})d\xi_{B_0} \int_0^{2\pi} d\varphi_{B_0}\,\lambda(1-\cos(\nu_{dd}t)). \quad (1.5)$$

$F(0)$ is the PDS signal intensity at the zero point, $t = 0$. r, ξ, φ, α, β, and γ are defined in Figure 1.2a, and $P()$ denotes their distributions. It is assumed that all distributions are normalized to 1, i.e.,

$$\int_0^{+\infty} P(r)dr = 1, \quad \int_0^{\pi} P(\xi)\,sin(\xi)d\xi = 1, \quad \int_0^{2\pi} P(\varphi)d\varphi = 1,$$
$$\int_0^{2\pi} P(\alpha)d\alpha = 1\,, \quad \int_0^{\pi} P(\beta)\sin(\beta)d\beta = 1, \quad \int_0^{2\pi} P(\gamma)\,d\gamma = 1. \quad (1.6)$$

ξ_{B_0} and φ_{B_0} are the polar and azimuthal angles, respectively, which describe the orientation of the magnetic field $\vec{B}_0$ in the reference coordinate system fixed at one of the spins (see Chapter 1.2). ν_{dd} is the dipolar frequency introduced in Chapter 1.1. λ is called the modulation depth parameter. It defines the probability with which each spin pair contributes to the dipolar modulation of the PDS time trace. In Equation 1.5, λ is averaged over all possible orientations of the spin system with respect to the magnetic field $\vec{B}_0$, yielding a total modulation depth parameter Λ. Λ can be determined as a decay of the $F(t)$ amplitude from 1, corresponding to $t = 0$, to 1 - Λ, corresponding to the t values at which the dipolar oscillations are fully damped Figure 1.1c). Importantly, both ν_{dd} and λ depend on integration variables. The mathematical

description of these dependencies is quite sophisticated and therefore will be discussed in separate Chapters 1.6.1 and 1.6.2.

If in addition to the dipole-dipole coupling the electron spin centers exhibit a weak exchange coupling and the corresponding exchange coupling constant J has a distribution $P(J)$, Equation 1.5 can be extended to[61]

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{4\pi}\int_{-\infty}^{+\infty} P(J)dJ \int_{0}^{+\infty} P(r)dr \int_{0}^{\pi} P(\xi)\sin(\xi)d\xi \int_{0}^{2\pi} P(\varphi)d\varphi \int_{0}^{2\pi} P(\alpha)d\alpha \times$$
$$\times \int_{0}^{\pi} P(\beta)\sin(\beta)d\beta \int_{0}^{2\pi} P(\gamma)\,d\gamma \int_{0}^{\pi} \sin(\xi_{B_0})d\xi_{B_0} \int_{0}^{2\pi} d\varphi_{B_0}\,\lambda(1-\cos(\{\nu_{dd}+J\}t)). \quad (1.7)$$

For the m-spin system ($m \geq 3$), $F(t)$ is given by the following equation[62]

$$\frac{F(t)}{F(0)} = \frac{1}{m}\sum_{i=1}^{m}\prod_{\substack{j=1 \\ j\neq i}}^{m} F_{ij}(t), \quad (1.8)$$

where $F_{ij}(t)$ corresponds to $F(t)$ of a two-spin subsystem, in which spin i is the detected spin and spin j is the pumped spin. Thus, $F_{ij}(t)$ can be calculated using Equation 1.5, omitting the second summand in the equation for λ (Equation 1.11).

1.6.1 Dipolar frequency

There are several equations for the dipolar frequency ν_{dd}. The choice of an appropriate equation depends on the g-anisotropy of the interacting electron spin centers. The most common case is the electron spin centers with isotropic or nearly isotropic g-factors, such as nitroxide radicals. For these spin centers, ν_{dd} is given by Equation 1.1 (see also Table 1.2). Apart from the constants and the g-values, ν_{dd} in Equation 1.1 depends on two parameters, the inter-spin distance and angle θ.

When one of the electron spin centers or both electron spin centers exhibit significant g-anisotropy (e.g., Cu^{2+} and Fe^{3+} ions), ν_{dd} needs to be calculated using Equations 1.9 or 1.10, respectively (Table 1.2).[66,67] The latter two equations take into account the fact that the quantization axes of anisotropic spin centers are not necessarily collinear with the applied magnetic field $\vec{B}_0$. Consequently, ν_{dd} becomes dependent on the relative orientation of one or both spin centers and the inter-spin vector $\vec{r}$. This dependence can be expressed in terms of the geometric parameters introduced in Figure 1.2a. When spin A is anisotropic, ν_{dd} depends not only on r and θ, but also on angles ξ and φ. When spin B is anisotropic too, ν_{dd} additionally depends on α, β, and γ.

Table 1.2. Different equations for the dipolar frequency.

g-factor of Spin A	g-factor of Spin B	Equation for the dipolar frequency[a]
isotropic	isotropic	$\nu_{dd} = \frac{\mu_0 \beta_e^2 g_\mathrm{A} g_\mathrm{B}}{4\pi h} \cdot \frac{(1 - 3\cos^2\theta)}{r^3}$ (same as 1.1)
isotropic anisotropic	anisotropic isotropic	$\nu_{dd} = \frac{\mu_0}{4\pi h} \frac{\beta_e^2 g_{\mathrm{A},eff} g_\mathrm{B}}{r^3} \left[1 - 3\left(\frac{\hat{g}_A \hat{g}_A^T}{g_{A,\,eff}^2} \frac{\vec{B}_0}{B_0},\ \frac{\vec{r}}{r}\right) cos(\theta)\right]$ (1.9)
anisotropic	anisotropic	$\nu_{dd} = \frac{\mu_0}{4\pi h} \frac{\beta_e^2 g_{\mathrm{A},eff} g_{\mathrm{B},eff}}{r^3} \left[\left(\frac{\hat{g}_A \hat{g}_A^T}{g_{A,\,eff}^2} \frac{\vec{B}_0}{B_0}, \frac{\hat{g}_B \hat{g}_B^T}{g_{B,\,eff}^2} \frac{\vec{B}_0}{B_0}\right) - 3\left(\frac{\hat{g}_A \hat{g}_A^T}{g_{A,\,eff}^2} \frac{\vec{B}_0}{B_0},\ \frac{\vec{r}}{r}\right)\left(\frac{\hat{g}_B \hat{g}_B^T}{g_{B,eff}^2} \frac{\vec{B}_0}{B_0},\ \frac{\vec{r}}{r}\right)\right]$ (1.10)

[a] μ_0 is the vacuum permeability, β_e is the Bohr magneton, h is the Planck constant, $\vec{r}$ is the inter-spin vector with the length of r, $\vec{B}_0$ is the magnetic field vector with the magnitude of B_0, θ is the angle between $\vec{r}$ and $\vec{B}_0$, g_A and g_B are *g*-factors of the isotropic spins A and B, respectively, $\hat{g}_A$ and $\hat{g}_B$ are the *g*-tensors, and $g_{\mathrm{A},eff}$ and $g_{\mathrm{B},eff}$ are effective *g*-values of the anisotropic spins A and B, respectively.

1.6.2 Modulation depth parameter

The modulation depth parameter λ for a single orientation of the two-spin system in the applied magnetic field $\vec{B}_0$ is given by

$$\lambda = \frac{p_{det}(\omega_A) \cdot p_{pump}(\omega_B) + p_{det}(\omega_B) \cdot p_{pump}(\omega_A)}{F(0)} \tag{1.11}$$

where

$$F(0) = \frac{1}{4\pi} \int_0^{\pi} \sin(\xi_{B_0}) d\xi_{B_0} \int_0^{2\pi} d\varphi_{B_0}\ \{p_{det}(\omega_A) + p_{det}(\omega_B)\} \tag{1.12}$$

Here, ω_A and ω_B are the resonance angular frequencies of spins A and B, respectively. $p_{det}(\omega)$ is the probability that the spin with resonance frequency ω is excited by the detection pulses and thereby contributes to the net transverse magnetization. $p_{pump}(\omega)$ is the probability that the spin with resonance frequency ω is flipped by the pump pulse (in PELDOR) or the relaxation (in RIDME) and thereby contributes to the dipolar modulation of the PDS signal.

If several different resonance angular frequencies ω_A and ω_B correspond to the same orientation of the magnetic field $\vec{B}_0$ (e.g., due to the hyperfine interaction of spin A and/or B with nuclear spins), Equations 1.11 and 1.12 have to be slightly modified by introducing the sum over different values of ω_A and ω_B:

$$\lambda = \frac{\sum_{\omega_A} \sum_{\omega_B} w_{\omega_A} w_{\omega_B} \{p_{det}(\omega_A) \cdot p_{pump}(\omega_B) + p_{det}(\omega_B) \cdot p_{pump}(\omega_A)\}}{F(0)} \tag{1.13}$$

with

$$F(0) = \frac{1}{4\pi}\int_0^{\pi} \sin(\xi_{B_0}) d\xi_{B_0} \int_0^{2\pi} d\varphi_{B_0} \sum_{\omega_A}\sum_{\omega_B} w_{\omega_A} w_{\omega_B} \{p_{det}(\omega_A) + p_{det}(\omega_B)\}, \quad (1.14)$$

where w_{ω_A} and w_{ω_B} are the relative weights of different ω_A and ω_B values, respectively. The sum of all w_{ω_A}, as well as the sum of all w_{ω_B}, equals 1.

The equations for p_{det} and p_{pump} depend on the PDS pulse sequence. In the case of four-pulse PELDOR with rectangular pulses (Figure 1.1b), they can be calculated as follows[68]

$$\begin{aligned} p_{det}(\omega) &= \frac{\omega_{1,\pi/2}}{\Omega_{\pi/2}} \sin\left(\Omega_{\pi/2} t_{\pi/2}\right) \frac{\omega_{1,\pi}^4}{4\Omega_{\pi}^4} \{1 - \cos(\Omega_{\pi} t_{\pi})\}^2, \\ p_{pump}(\omega) &= \frac{\omega_{1,pump}^2}{2\Omega_{pump}^2} \left\{1 - \cos\left(\Omega_{pump} t_{pump}\right)\right\}, \end{aligned} \quad (1.15)$$

where

$$\begin{aligned} \omega_{1,\pi/2} &= \frac{\pi/2}{t_{\pi/2}}, & \Omega_{\pi/2}^2 &= \omega_{1,\pi/2}^2 + (\omega - \omega_{det})^2, \\ \omega_{1,\pi} &= \frac{\pi}{t_{\pi}}, & \Omega_{\pi}^2 &= \omega_{1,\pi}^2 + (\omega - \omega_{det})^2, \\ \omega_{1,pump} &= \frac{\pi}{t_{pump}}, & \Omega_{pump}^2 &= \omega_{1,pump}^2 + (\omega - \omega_{pump})^2. \end{aligned} \quad (1.16)$$

Here $t_{\pi/2}$, t_{π}, and t_{pump} are the lengths of the π/2, π, and pump pulses, respectively. ω_{det} and ω_{pump} are the detection and pump angular frequencies, respectively.

When the rectangular pump pulse in the PELDOR pulse sequence is replaced by a chirp pump pulse, p_{pump} is calculated using the Landau-Zener-Stückelberg-Majorana equation,[69]

$$p_{pump}(\omega) = \exp\left(-\frac{\pi}{2} Q_{crit}(\omega)\right). \quad (1.17)$$

The critical adiabaticity at the frequency ω, $Q_{crit}(\omega)$, is given by

$$Q_{crit}(\omega) = \min_t Q_{crit}(\omega, t), \quad Q_{crit}(\omega, t) = \frac{\{\omega_1^2(t) + \Delta\omega^2(\omega, t)\}^{\frac{3}{2}}}{\left|\omega_1(t) \frac{d\Delta\omega(\omega, t)}{dt} - \Delta\omega(\omega, t) \frac{d\omega_1(t)}{dt}\right|}, \quad (1.18)$$

where

$$\Delta\omega(\omega, t) = \omega_{pump}(t) - \omega, \quad \omega_{pump}(t) = \omega_{pump,c} - \frac{1}{2}\Delta\omega_{pump} + \Delta\omega_{pump} \frac{t}{t_{pump}}. \quad (1.19)$$

$$\omega_1(t) = \sqrt{\frac{Q_{crit}\Delta\omega_{pump}}{t_{pump}}} \begin{cases} \sin\left(\frac{\pi}{2}\frac{t}{t_{rise}}\right), & t \in [0, t_{rise}) \\ 1, & t \in [t_{rise}, t_{pump} - t_{rise}] \\ \sin\left(\frac{\pi}{2}\frac{t_{pump} - t}{t_{rise}}\right), & t \in (t_{pump} - t_{rise}, t_{pump}] \end{cases} \quad (1.20)$$

Here, t_{pump} and t_{rise} are the length and the rise time of the pump pulse, respectively, $\omega_{pump,c}$ is the central pump frequency, $\Delta\omega_{pump}$ is the frequency sweep width, and Q_{crit} is the critical adiabaticity.

In the case of five-pulse RIDME with rectangular pulses (Figure 1.1b), p_{det} and p_{pump} can be calculated using the following equations[7,70]

$$\begin{aligned} p_{det}(\omega) &= \frac{\omega_{1,\pi/2}^3}{\Omega_{\pi/2}^3}\sin^3\left(\Omega_{\pi/2}t_{\pi/2}\right)\frac{\omega_{1,\pi}^4}{4\Omega_{\pi}^4}\{1-\cos(\Omega_{\pi}t_{\pi})\}^2, \\ p_{pump} &= \frac{1}{2}\left\{1-\exp\left(-\frac{T_{mix}}{T_1}\right)\right\}, \end{aligned} \tag{1.21}$$

where $t_{\pi/2}$, t_{π}, $\omega_{1,\pi/2}$, $\omega_{1,\pi}$, $\Omega_{\pi/2}$, and Ω_{π} are defined above, T_{mix} is the length of the mixing interpulse interval in the RIDME pulse sequence, and T_1 is the longitudinal (spin-lattice) relaxation time of the flipped spins. Note that in RIDME p_{pump} does not depend on the resonance frequency ω of the flipped spin. However, at temperatures comparable to the electron Zeeman energy (usually liquid helium temperatures), p_{pump} depends on the effective g-factor of the flipped spin, g_{eff},[44]

$$p_{pump} = \frac{2f}{(1+f)^2}\left\{1-\exp\left(-\frac{T_{mix}}{T_1}\right)\right\} \quad \text{with} \quad f = \exp\left(-\frac{g_{eff}\beta B_0}{kT}\right), \tag{1.22}$$

where β is the Bohr magneton, k is the Boltzmann constant, and T is the temperature of the RIDME experiment. In the high-temperature limit, Equation 1.22 transforms into Equation 1.21.

1.7 Simulation of PDS time traces

The simulation of the PDS time traces is based on Equation 1.2. According to this equation, the PDS time trace $V(t)$ consists of two contributions, the form factor $F(t)$ and the background $B(t)$. $F(t)$ is calculated using Equation 1.6 and 1.8 for two-spin and m-spin ($m \geq 3$) systems, respectively. Since $F(t)$ of m-spin systems can be expressed in terms of two-spin contributions $F_{ij}(t)$ (Equation 1.8), the following discussion focuses only on two-spin systems. To compute $F(t)$ using Equation 1.6, the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ must be given. Recall that in PeldorFit each of these distributions is approximated by either a uniform distribution or a Gaussian distribution (Chapter 1.2.1). Moreover, all these distributions can be either unimodal (Chapter 1.2.1) or multimodal (Chapter 1.2.2). In the case of the unimodal distributions, the user has to provide the values of 12 parameters including 6 mean values, $\langle r \rangle$, $\langle \xi \rangle$, $\langle \varphi \rangle$, $\langle \alpha \rangle$, $\langle \beta \rangle$, and $\langle \gamma \rangle$, and 6 widths, Δr, $\Delta \xi$, $\Delta \varphi$, $\Delta \alpha$, $\Delta \beta$, and $\Delta \gamma$. In the case of the n-modal distributions, the number of parameters increases to $(13n - 1)$, which includes $6n$ mean values, $6n$ widths and $(n - 1)$ relative weights. In addition, the distribution of exchange coupling constant, $P(J)$, can be included in the simulation via Equation 1.7. This option is currently available only for the two-spin model with unimodal distributions (Chapter 1.2.1). If included, $P(J)$ is also approximated by either a uniform distribution or a Gaussian distribution. This yields two additional simulation parameters, a mean value $\langle J \rangle$ and a width ΔJ. The integral in Equations 1.6 and 1.7 is solved numerically using the Monte-Carlo method and the sampling distributions. The important steps of the numerical integration are:

1) Using the given distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, $P(\gamma)$, and $P(J)$, the sampling distributions $P_s(r) = P(r)$, $P_s(\xi) = P(\xi)\sin(\xi)$, $P_s(\varphi) = P(\varphi)$, $P_s(\alpha) = P(\alpha)$, $P_s(\beta) = P(\beta)\sin(\beta)$, $P_s(\gamma) = P(\gamma)$, $P_s(\xi_{B_0}) = \sin(\xi_{B_0})$, $P_s(\varphi_{B_0}) = 1$, and $P_s(J) = P(J)$ are calculated.
2) N_{MC} Monte-Carlo samples are generated by randomly picking the values of r, ξ, φ, α, β, γ, and J from the corresponding sampling distributions. This yields N_{MC} sets of integration variables $(r_i, \xi_i, \varphi_i, \alpha_i, \beta_i, \gamma_i, \xi_{B_0,i}, \varphi_{B_0,i}, J_i)$, where $i = 1, 2, \ldots, N_{MC}$. N_{MC} determines the precision of the numerical integration and has to be set as high as possible. In this manual, N_{MC} was 10^5-10^6.
3) The values of ν_{dd} and λ are calculated for each Monte-Carlo sample above, yielding $\nu_{dd,i}$ and λ_i ($i = 1, 2, \ldots, N_{MC}$). The equations for ν_{dd} and the dependence of ν_{dd} on the integration variables can be found in Chapter 1.6.1. Note that the values of ν_{dd} depend on the g-factors of the electron spin centers. Therefore, the user must specify the principal g-factors for each of the electron spin centers. The equations for λ can be found in Chapter

1.6.2. The dependence of λ on the integration variables is hidden in the resonance angular frequencies ω_A and ω_B that appear in Equations 1.10-1.14. The calculation of ω_A and ω_B requires knowledge of the EPR parameters of spins A and B. Therefore, in addition to the principal *g*-factors, the user must specify all other EPR parameters relevant to the simulation of the EPR spectra of spins A and B (e.g., the number and type of coupled nuclear spins, the hyperfine coupling constants, the inhomogeneous linewidth, etc.). Furthermore, λ depends on the type of PDS experiment (PELDOR or RIDME) and the experimental settings used to perform that experiment (Chapter 1.6.2). This information must also be provided by the user. In the case of RIDME, the T_1 relaxation times of the electron spin centers must be specified too.

4) Finally, $F(t)$ is calculated as a sum over the Monte-Carlo samples:

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{N}\sum_{i=1}^{N_{MC}} \lambda_i\{1 - cos((\nu_{dd,i} + J_i)t)\} \tag{1.23}$$

To compute $B(t)$, the user must specify the type of function used to approximate $B(t)$ (background model) and the parameter(s) of that function (background parameters). The list of background models and corresponding background parameters can be found in Table 1.3. The value(s) of background parameter(s) can be either specified directly by the user or optimized by PeldorFit within the ranges defined by the user. In the latter case, the optimization goal is to find such a $B(t)$ that will minimize the difference between the experimental PDS time trace $V_{exp}(t)$ and the corresponding simulated PDS time trace $V_{sim}(t)$:

$$\min_{\vec{b}}\left(\sum_{j=1}^{N_t}\{V_{exp}(t_j) - V_{sim}(t_j,\vec{b})\}^2\right) = \min_{\vec{b}}\left(\sum_{j=1}^{L}\{V_{exp}(t_j) - F(t_j)B(t_j,\vec{b})\}^2\right) \tag{1.24}$$

where $\vec{b}$ stands for the vector with the background parameter(s), and the sum over *j* denotes the sum over the time points in the PDS time trace. The optimization in Equation 1.24 is done using the Nelder-Mead algorithm.

Often the predicted value of the total modulation depth parameter Λ (Chapter 1.5) exceeds its experimental value. This can have several reasons such as the imperfection of the pump pulse in the PELDOR experiment or non-ideal sample conditions, e.g., incomplete spin labeling or incomplete metal loading of biomolecules. To reproduce this effect in the simulations, PeldorFit uses an additional parameter η, which is used as a modulation depth scale factor. Ideally, $\eta = 1$, and otherwise $0 < \eta < 1$. In rare cases, η was found to be larger than 1. Similar to the background parameters, the value of η can be either predefined by the user or optimized by PeldorFit within

Table 1.3. Background models and their parameters.

Background model	Expression	Parameters
exponential decay	$B(t) = \exp(-kt)$	k
stretched exponential decay	$B(t) = \exp\left(-kt^{d/3}\right)$	k, d
2nd order polynomial	$B(t) = 1 + c_1 t + c_2 t^2$	c_1, c_2
3rd order polynomial	$B(t) = 1 + c_1 t + c_2 t^2 + c_3 t^3$	c_1, c_2, c_3
4th order polynomial	$B(t) = 1 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4$	c_1, c_2, c_3, c_4
Keller's exponential decay	$B(t) = \exp(k_1 t + k_2 t^2)$	k_1, k_2

the ranges defined by the user. In the latter case, η is optimized simultaneously with the background parameters (if some of them are optimized). For this, Equation 1.24 is replaced by

$$\min_{\vec{b},\eta}\left(\sum_{j=1}^{L}\left\{V_{exp}(t_j) - F(t_j,\eta)B(t_j,\vec{b})\right\}^2\right), \text{where } \frac{F(t_j,\eta)}{F(0,\eta)} = 1 - \eta\left(1 - \frac{F(t_j)}{F(0)}\right) \qquad (1.25)$$

1.8 Fitting of PDS time traces

In addition to the simulation, PeldorFit allows to fit the PDS time traces using the parameters of the user-defined model of a spin system as fitting parameters. The supported models and their parameters can be found in Chapter 1.2. For example, if the simplest two-spin model with the unimodal distributions (Chapter 1.2.1) is chosen, six mean values, $\langle r\rangle$, $\langle\xi\rangle$, $\langle\varphi\rangle$, $\langle\alpha\rangle$, $\langle\beta\rangle$, and $\langle\gamma\rangle$, and six widths, Δr, $\Delta\xi$, $\Delta\varphi$, $\Delta\alpha$, $\Delta\beta$, and $\Delta\gamma$, will be the fitting parameters. If necessary, the mean value $\langle J\rangle$ and the distribution width ΔJ of the exchange coupling constant (Chapters 1.6 and 1.7) can be also used as fitting parameters. Depending on the EPR properties of particular spin system, some of these parameters may not affect the PDS time traces (Chapter 1.4). The user can exclude such parameters from the fitting and set them to constant values. The same is possible if the values of some fitting parameters are known in advance. In addition to the model parameters and exchange coupling constant, the background parameters (Table 1.3) and the modulation depth scale factor η (Chapter 1.7) can be optimized. Since the calculation of PDS backgrounds can be well separated from the calculation of the PDS form factors (Chapter 1.6), the optimization of the background parameters is separated from the optimization of model parameters and performed via the procedure described in Chapter 1.7. For this reason, the background parameters are not considered in the following discussion.

During the fitting, the values of the fitting parameters are optimized within the user-defined ranges until χ^2 deviation between the experimental and simulated PDS time trace(s) is minimized:

$$\min_{\vec{p}}\{\chi^2(\vec{p})\},\ \text{where}\ \chi^2(\vec{p}) = \sum_{k=1}^{N_V}\sum_{j=1}^{N_t}\frac{\left(V_{exp,k}(t_j) - V_{sim,k}(t_j,\vec{p})\right)^2}{\sigma_{N,k}^2}, \tag{1.26}$$

where $\vec{p}$ stands for the vector with the fitting parameter(s), the sums over k and j denote the sum over different PDS time traces and the sum over the time points in k^{th} PDS time trace, respectively, $V_{exp,k}(t_j)$ and $V_{sim,k}(t_j,\vec{p})$ are k^{th} experimental PDS time trace and the corresponding simulated PDS time trace (the fit), respectively, and $\sigma_{N,k}$ is the noise level (its standard deviation) in the k^{th} experimental PDS time trace. The calculation of $V_{sim,k}(t_j,\vec{p})$ is done in the same way as described in Chapter 1.7. σ_N is determined from the imaginary component of the PDS time trace (Chapter 1.5).

The optimization of the fitting parameters is done by means of a genetic algorithm.[71,72] This algorithm was chosen because 1) it is a global optimization algorithm, i.e., it is able to find a global minimum of χ^2 even if χ^2 has several local minima, 2) it does not require initial guesses and calculating any derivatives, and 3) it is well suited to deal with a large number of fitting parameters in terms of computational time. A brief description of the implemented genetic algorithm is given in Chapter 1.8.1. Here, some general features of the algorithm need to be highlighted. First, the genetic algorithm is an iterative algorithm that approaches the minimum of χ^2 in multiple optimization steps (Figure 1.6). Therefore, the maximum number of optimization steps is important for the convergence of the algorithm to the global minimum. In fact, the genetic algorithm has a few more intrinsic parameters that can affect its ability to find the global minimum. This is discussed in more detail in Chapter 1.8.1. Second, the genetic algorithm is a stochastic algorithm, that is, an algorithm that explicitly uses randomness to find

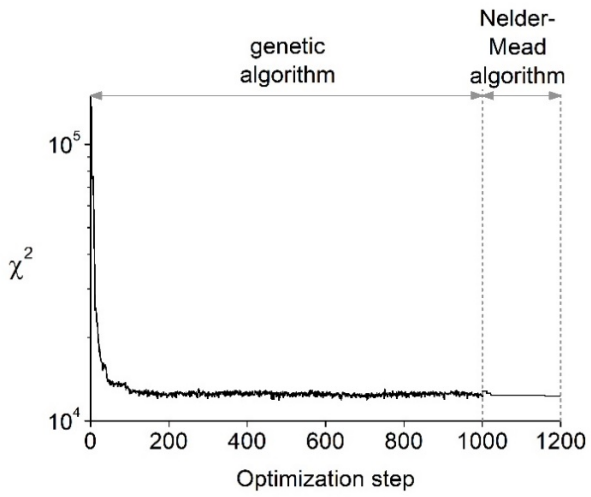

Figure 1.6. χ^2 as a function of optimization step. Optimization steps 1-1000 and 1001-1200 were carried out by the genetic and Nelder-Mead algorithms, respectively.

the minimum of χ^2. Therefore, if the global minimum of χ^2 spans over a certain range of fitting parameters, e.g., due to noise in the experimental PDS data, the genetic algorithm may yield slightly different solutions to the same optimization problem. This does not mean anything other than that the different sets of fitting parameters may yield a comparable χ^2. However, it was sometimes observed that the genetic algorithm can quickly find the solution that appears very close to the global minimum of χ^2, but then, due to stochasticity, requires many more optimization steps to reach the "absolute minimum" of χ^2. To avoid this issue, PeldorFit offers the possibility to apply a deterministic, local optimization algorithm, called Nelder-Mead simplex algorithm, right after the genetic algorithm. Using the initial guess supplied by the genetic algorithm, the Nelder-Mead algorithm quickly converges to the global minimum of χ^2 (Figure 1.6).

1.7.1 Genetic algorithm

The genetic algorithm (GA) is a metaheuristic algorithm that mimics the processes of Darwin's natural evolution. A detailed description and various implementations of the GA can be found elsewhere.[71,72] Here, only a brief description of the implemented GA is given to help the users of PeldorFit to use this algorithm properly.

First, the terminology needs to be explained. In the GA, each fitting parameter is called a "gene". A set of genes determines a complete set of fitting parameters and is called a "chromosome". Thus, the number of genes in a chromosome is determined by the number of fitting parameters. Each chromosome represents one possible solution to the optimization problem. The accuracy of each individual solution is given by the χ^2 value (Equation 1.26) and called the "fitness" property of the chromosome. Chromosomes with the best fitness are chromosomes with the lowest χ^2 values. The procedure that describes the calculation of the fitness also has its own name and is called "scoring". Moreover, the GA always deals with a set of chromosomes called a "generation". The size of a generation, N_c, is specified by the user.

The block diagram of the implemented GA is shown in Figure 1.7. An evolutionary cycle begins with the creation of N_c chromosomes that form a first generation. The genes of the initial chromosomes are randomly selected from the user-defined ranges. Then, all chromosomes in the generation are subjected to scoring, which yields the fitness for each chromosome. According to the evolutionary principle, the fitness determines which chromosomes of the current generation have better chances to produce an offspring and, thus, to contribute to the next generation. To obtain an offspring, the following three procedures are repeated $N_c/2$ times:

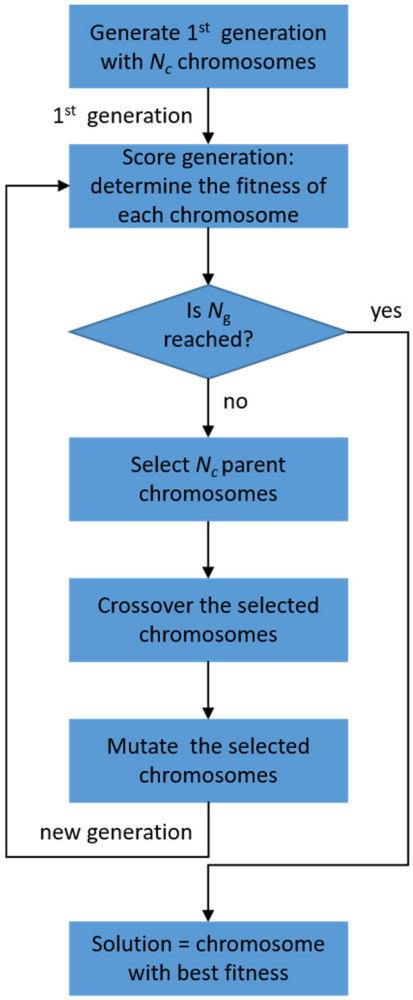

Figure 1.7. Block diagram of the GA implemented in PeldorFit.

1) Two parent chromosomes are determined by a tournament selection. In the tournament selection, two pairs of chromosomes are picked out randomly from the current generation, and the chromosome with the highest fitness is selected from each pair. This procedure ensures that the chromosomes with better fitness have a higher chance of being selected.
2) The genes of the two selected chromosomes undergo crossover with probability p_c. Crossover is responsible for the gene exchange between the two selected chromosomes around one random position in the gene sequence. This procedure is absolutely crucial for the convergence of the GA to the global minimum of χ^2.
3) The genes of the two selected chromosomes undergo mutation with probability p_m. Mutation occurs by randomly changing the genes of chromosomes. It introduces a certain diversity into the generation, preventing rapid convergence of the GA to local minima of χ^2.

Once a new generation with N_c offspring chromosomes has been created, the next evolutionary cycle begins, starting with the scoring and mating described above. The whole process is

repeated many times until the desired maximal number of generations, N_g, is reached. If each generation of the GA is considered as a single optimization step, N_g determines the maximum number of optimization steps of the GA.

On the one hand, the GA has the advantage that it does not require initial estimates for the values of the fitting parameters. On the other hand, the GA has several intrinsic parameters that can affect the ability of GA to find the global minimum of χ^2. Therefore, the values of the intrinsic parameters need to be optimized. Unfortunately, there are no clear rules on how to determine the optimal values of these parameters, and usually they have to be determined empirically. Moreover, the optimal values of the parameters may vary depending on the particular PELDOR data. Our numerous tests showed that reasonable fits to the PELDOR time traces can be obtained by following the empirical rules below:

- Set the number of chromosomes per generation (N_c) to a value which is ~10 times larger than the number of fitting parameters. In general, the performance of the GA improves as N_c increases, but this also causes the optimization time to increase linearly with N_c (if the number of available CPUs is limited).
- Set the maximum number of generations (N_g) to 1000. Usually, this number of generations is sufficient for the GA to converge to the global minimum of χ^2. If the GA does not converge after 1000 generations, as indicated by significant changes in χ^2 over the last few hundred generations, increase the value of N_g. In general, the increase of N_g makes the convergence of the GA more likely, but on the downside, the optimization time increases linearly with N_g (regardless of the number of available CPUs).
- Set the crossover probability (p_c) and the mutation probability (p_m) to 0.5 and 0.01, respectively. Of these two parameters, only the mutation probability was shown to influence the performance of the GA in our tests. As mentioned earlier, mutation is very important for the GA, because it allows avoiding local minima of χ^2. The GA can escape the local minima only if the value of p_m is sufficiently large. However, if p_m is too large, the mutation may destroy chromosomes with good fitness, causing poor convergence. Thus, a compromise between these two effects must be found. Usually $p_m = 0.01$ provides such a compromise. Otherwise, if the convergence of the GA is achieved, but the fit to the PDS time traces does not look good, one can try to increase the value of p_m, e.g., up to 0.05. For such difficult cases, PeldorFit offers the possibility to perform several consequent GA runs with $p_m = p_{m,0} + \Delta p_m \cdot r$, where $p_{m,0}$ is the initial value of p_m, Δp_m is the increment of p_m, and r is the number of the GA run.

1.9 Error analysis

Fitting of the PDS time traces by PeldorFit provides the optimized values of the fitting parameters. The precision of these parameters is however unknown. To fill this gap, PeldorFit allows to perform an error analysis. Since the parameter space is quite large and fitting of the PDS time traces is very time consuming (hours to days), common error analyses, such as bootstrap[73] or Bayesian analysis[74], cannot be done in a reasonable time. Therefore, the error analysis is performed in a less general way.

The error analysis starts by recording χ^2 for various subspaces of fitting parameters, typically consisting of one or two fitting parameters. Fitting parameters that do not belong to the selected subspace are set to their optimized values. To estimate the errors of the fitting parameters from these dependencies, some simplifying assumptions have to be made. One possible assumption is that, near the global minimum of χ^2, the contributions to χ^2 from the fitting parameters describing individual distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, $P(\gamma)$, and $P(J)$ are uncorrelated and can therefore be considered separately. Thus, for the two-spin model with the unimodal distributions (Chapter 1.2.1), it is sufficient to record χ^2 for seven two-dimensional subspaces of the fitting parameters, namely $\chi^2(<r>, \Delta r)$, $\chi^2(<\xi>, \Delta\xi)$, $\chi^2(<\varphi>, \Delta\varphi)$, $\chi^2(<\alpha>, \Delta\alpha)$, $\chi^2(<\beta>, \Delta\beta)$, $\chi^2(<\gamma>, \Delta\gamma)$, and $\chi^2(<J>, \Delta J)$ (Figure 1.8a). If some of parameters r, ξ, φ, α, β, γ, and J are not included in the fitting, the corresponding dependencies can be omitted. When distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, $P(\gamma)$, and $P(J)$ are approximated by multimodal distributions (Chapter 1.2.2), the dependence of χ^2 on the parameters of each distribution can be further separated into the dependencies of χ^2 on the parameters of each mode in that distribution. For example, for the bimodal distance distribution, the dependencies $\chi^2(<r_1>, \Delta r_1)$, $\chi^2(<r_2>, \Delta r_2)$ and $\chi^2(w)$ can be considered, where the subscripts "1" and "2" denote the two modes and w is the relative weight of the first mode.

In the next step, each two-dimensional dependence, e.g. $\chi^2(\langle r\rangle, \Delta r)$ (Figure 1.8a), is transformed into two one-dimensional dependencies, e.g., $\chi^2(\langle r\rangle)$ and $\chi^2(\Delta r)$ (Figure 1.8b), where the values of χ^2 are minimized with respect to Δr and $\langle r\rangle$, respectively. The obtained one-dimensional dependencies are then used to determine parameter ranges in which the deviation of the χ^2 values from the global minimum χ^2_{min} is smaller than a threshold $\Delta\chi^2$ (Figure 1.8b). $\Delta\chi^2$ is composed of two contributions. The first contribution, $\Delta\chi_{ci}^2$, accounts for errors associated with noise in the experimental PDS time traces and possible discrepancies between the actual spin

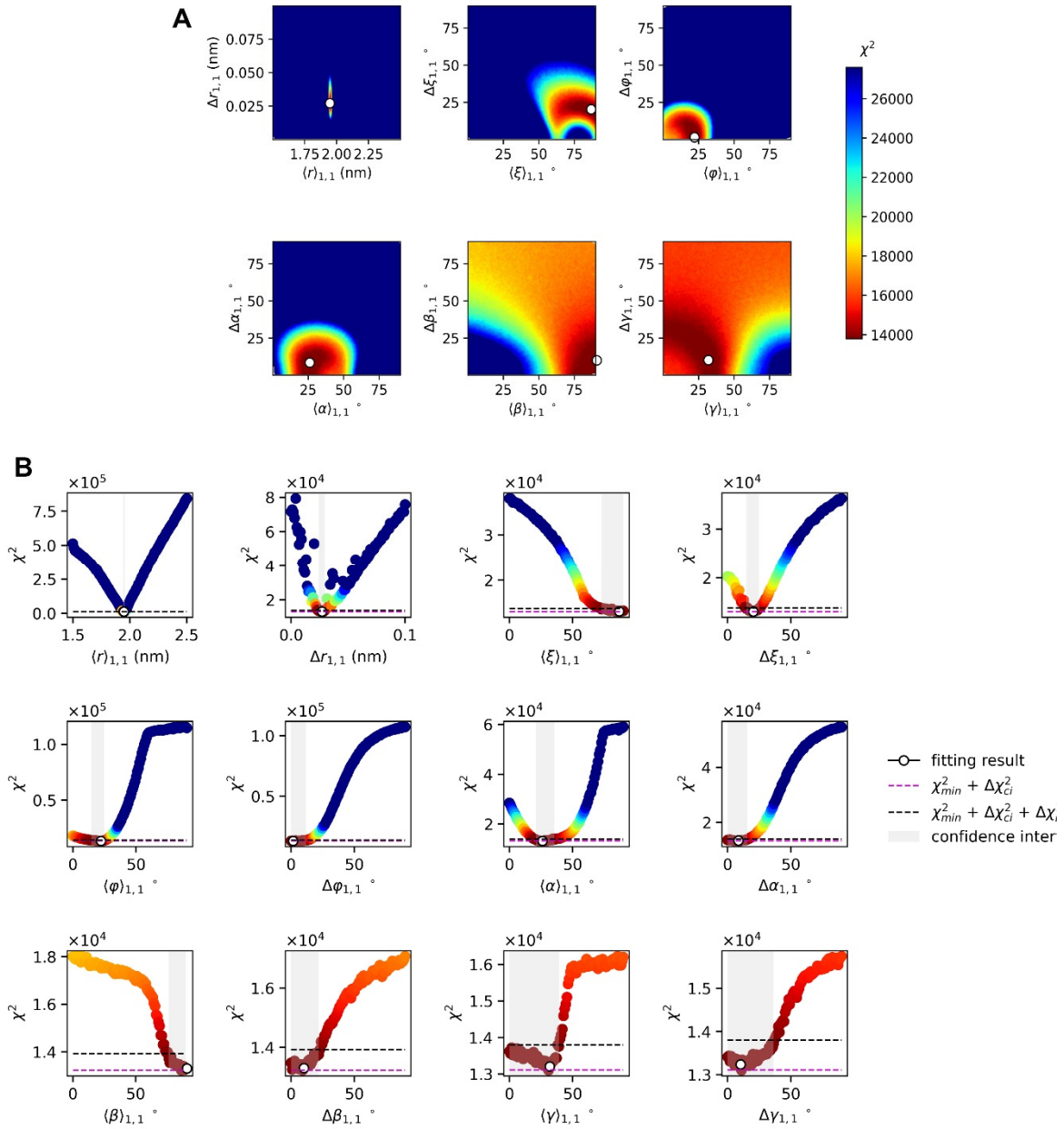

Figure 1.8. Error analysis for the optimized fitting parameters of PeldorFit. A) χ^2 in dependence of the fitting parameters describing the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$. The optimized values of the fitting parameters are depicted by a circle. B) χ^2 in dependence of individual fitting parameters. The χ^2 threshold is depicted by black dashed lines. The uncertainty ranges are shown as gray intervals. Reproduced from Figure 6.4.

system and its model. In PeldorFit, $\Delta\chi_{ci}^2$ is set in accordance to the 3σ confidence interval. Assuming that the measurement errors are distributed normally, the 3σ confidence interval corresponds to $\Delta\chi_{ci}^2 = 3^2 = 9$.[73] The second contribution to $\Delta\chi^2$, $\Delta\chi_{ne}^2$, accounts for the numerical error, which is mainly determined by the accuracy of the Monte-Carlo integration. The value of $\Delta\chi_{ne}^2$ is estimated by calculating χ^2 for the 10^4 identical sets of optimized fitting parameters and then finding the difference between the maximum and minimum values of χ^2.

Lastly, the obtained uncertainty intervals are converted into the errors of the fitting parameters. Our test showed that the uncertainty intervals are often asymmetric with respect to the optimized value. Therefore, the errors of the fitting parameters are calculated as asymmetric errors. An asymmetric error consists of two values: the first is given by the difference between the lower bound of the uncertainty interval and the optimized value of a fitting parameter, and the second is given by the difference between the upper bound of the uncertainty interval and the optimized value of a fitting parameter. For example, the optimized values and asymmetric errors of parameters $<r>$, Δr, $<\xi>$, $\Delta\xi$, $<\varphi>$, $\Delta\varphi$, $<\alpha>$, $\Delta\alpha$, $<\beta>$, $\Delta\beta$, $<\gamma>$, and $\Delta\gamma$ in Figure 1.8 are $1.948^{+0.004}_{-0.001}$ nm, $0.027^{+0.002}_{-0.003}$ nm, $87^{+3}_{-14}{}^{\circ}$, $20^{+5}_{-6}{}^{\circ}$, $22^{+3}_{-8}{}^{\circ}$, $2^{+10}_{-1}{}^{\circ}$, $26^{+10}_{-6}{}^{\circ}$, $9^{+7}_{-9}{}^{\circ}$, $90^{+0}_{-14}{}^{\circ}$, $10^{+12}_{-10}{}^{\circ}$, $32^{+7}_{-32}{}^{\circ}$, and $10^{+26}_{-10}{}^{\circ}$, respectively.

1.10 Technical information

PeldorFit 2022 is a console application written in Python 3.8.6 using the libraries numpy 1.19.3, scipy 1.5.4, matplotlib 3.3.3, and libconf 2.0.1. The source code of the program can be found at

https://github.com/dinarabdullin/PeldorFit2022.

It is free of charge and can be distributed under GNU General Public License v3.0.

PeldorFit is also available as an executable program, which can be used without installing a Python interpreter and its libraries. Currently, such an executable program is available for two operation systems, Windows 10 and Linux CentOS 7. The PeldorFit executable for Windows 10 can be downloaded from:

https://github.com/dinarabdullin/PeldorFit2022/releases/tag/windows10.

The PeldorFit executable for PeldorFit for Linux CentOS 7 can be downloaded from:

https://github.com/dinarabdullin/PeldorFit2022/releases/tag/centos7.

The PeldorFit executable for other operating systems can be prepared on request (contact abdullin@pc.uni-bonn.de).

The calculations performed by means of PeldorFit usually require considerable computing power. Therefore, it is advisable to run the program on a server that has a significant number of CPUs and RAM. When PeldorFit is used in the fitting mode, the optimal number of CPUs is given by the number of chromosomes per generation, N_c (Chapter 1.7.1). All examples in this manual were performed using the "bonna" cluster (University of Bonn). One to six nodes were used, each with 32 CPUs and 192 GB RAM. The computation time was in the range of several hours to several days.

2 Installation

2.1 Using the PeldorFit executable

Currently PeldorFit executables are available for Windows 10 and Linux CentOS 7 operating systems (Chapter 1.10). To install the PeldorFit executable, download the zip archive with PeldorFit from https://github.com/dinarabdullin/PeldorFit2022/releases and unzip it into the user directory. That's all!

2.2 Using the PeldorFit source code

1) Make sure that Python 3.8 is installed. If not, install it from https://www.python.org.
2) Make sure that the default Python version is Python 3.8.X (X may differ). To check the default Python version, open Terminal (Linux, macOS) or Command Prompt (Windows) and type:

   `python3 --version` (Linux, macOS)

   `python --version` (Windows)

   To change the default Python version on a particular OS, see this link.
3) Download the zip archive with PeldorFit source code from https://github.com/dinarabdullin/PeldorFit2022 and unzip it into the user directory. Rename the folder with the PeldorFit source code (usually called PeldorFit2022-master) to PeldorFit.
4) Open new Terminal or new Command Prompt and navigate to the folder with the PeldorFit source code:

   ```
   cd [user directory]/PeldorFit
   ```
5) Now, the python libraries required for PeldorFit need to be installed. The best practice to do this is to use a virtual environment. To create a virtual environment, type in the Terminal or Command Prompt:

   `python3 -m venv peldorfit_env` (Linux, macOS)

   `python -m venv peldorfit_env` (Windows)

   Here, peldorfit_env is the name of environment. Next, activate the virtual environment:

   ```
   source peldorfit_env/bin/activate
   ```

   and install the required python libraries:

   ```
   pip install -U pip
   pip install numpy==1.19.3
   pip install scipy==1.5.4
   pip install matplotlib==3.3.3
   ```

```
pip install libconf==2.0.1
```

6) To convert the PeldorFit source code into the executable, a python library called pyinstaller is used. First, install this library into the peldorfit_env virtual environment:

```
pip install pyinstaller
```

Second, run the following command:

```
pyinstaller --onefile PeldorFit.py
```

This will generate the PeldorFit.spec file in the current folder. Open this file with any text editor and replace the empty brackets in exe = EXE(…) with [('W ignore', None, 'OPTION')]. After this, save and close the PeldorFit.spec file and run the following command:

```
pyinstaller --onefile PeldorFit.spec
```

7) Copy the PeldorFit executable PeldorFit (Linux, macOS) or PeldorFit.exe (Windows) from [user directory]/PeldorFit/dist into [user directory]/PeldorFit.

3 Running the program

3.1 Linux and macOS

To run the program on Linux and macOS, do the following steps:

1) Open Terminal and navigate to the folder with the PeldorFit executable:

```
cd [user directory]/PeldorFit
```

2) Set the permission properties:

```
chmod 755 PeldorFit
chmod 755 PeldorFit.sh
```

3) Prepare a configuration file (see Chapter 4).

4) In the Terminal, navigate to the folder with the configuration file:

```
cd [configuration file directory]
```

5) Run the program:

```
sh [user directory]/PeldorFit/PeldorFit.sh [configuration file name].cfg
```

If the configuration file does not contain any syntax errors, PeldorFit will start the calculations and show progress messages in Terminal or Command Prompt.

3.2 Windows

To run the program on Windows, do the following steps:

1) Open Command Prompt and navigate to the folder with the PeldorFit executable:

```
cd [user directory]/PeldorFit
```

2) Prepare the configuration file (see Chapter 4).

3) In the Command Prompt, navigate to the folder with the configuration file:

```
cd [configuration file directory]
```

4) Run the program:

```
[user directory]/PeldorFit/PeldorFit.exe [configuration file name].cfg
```

If the configuration file does not contain any syntax errors, PeldorFit will start the calculations and show progress messages in Terminal or Command Prompt.

4 Configuration file

The PeldorFit configuration file is an ASCII file with a *.cfg* extension that contains all input parameters. This file can be created and edited using common text editors, such as Notepad++. Since the number of input parameters in the configuration file is usually quite large, it is strongly recommended to use the configuration files from the "examples" folder as a template for creating new configuration files. This saves time and helps to avoid syntax errors. The configuration file has a defined structure consisting of 10 information blocks:

1) Operation mode (Chapter 4.1)
2) Experimental parameters (Chapter 4.2)
3) EPR parameters of a spin system (Chapter 4.3)
4) Simulation parameters (Chapter 4.4)
5) Fitting parameters (Chapter 4.5)
6) Fitting settings (Chapter 4.6)
7) Error analysis parameters (Chapter 4.7)
8) Error analysis settings (Chapter 4.8)
9) Calculation settings (Chapter 4.9)
10) Output settings (Chapter 4.10)

Information blocks 1-3 and 9-10 are mandatory. The information blocks 4-8 are optional and need only be specified if a certain operation mode was selected. Comment lines in the configuration file are marked by the symbol # at the beginning of the line.

4.1 Operation mode

As described in Chapter 1.5, PeldorFit has two main operation modes: "simulation" and "fitting". In the simulation mode, the PDS time traces are simulated using a user-defined model of the spin system and user-defined parameters of the model (Chapter 1.7). In the fitting mode, the parameters of the user-defined model are optimized until the simulated PDS time traces provide the best fit to the experimental simulated PDS time traces (Chapter 1.8). Precision of the optimized parameters is determined via the error analysis (Chapter 1.9). The error analysis can be done either directly after the fitting or at any time later. For the latter case, PeldorFit has an additional operation mode called "error analysis". PeldorFit can run only one of these three modes at a time.

The operation mode of PeldorFit is specified through the following line:

```
mode = 0;
```

Source: PeldorFit/examples/example_s1/config_example_s1.cfg

All options for mode are given below:

mode = 0	Turns on the simulation mode.
mode = 1	Turns on the fitting mode.
mode = 2	Turns on the error analysis mode.

4.2 Experimental parameters

Experimental PDS time traces and corresponding experimental settings are specified under keyword experiments:

```
experiments = (
   {
       name = "offset XX";
       filename = "offsetXX.txt";
       technique = "4pELDOR-rect";
       magnetic_field = 3.3415;
       detection_frequency = 93.999600;
       detection_pulse_lengths = [7, 14];
       pump_frequency = 93.930000;
       pump_pulse_lengths = [14];
       noise_std = 0.0016;
   },
   …
);
```

Source: PeldorFit/examples/example_s1/config_example_s1.cfg

Each individual PDS time trace has to be enclosed in curly brackets. Within the curly brackets, the following data is provided:

name	The name used to denote the PDS time trace in PeldorFit.
filename	The path to an ASCII file which contains the PDS time trace. The first column of this file must contain time points in nanoseconds. The second and third columns must contain the in-phase and quadrature components of the PDS signal respectively.
technique	The PDS technique used to record the PDS time trace. The list of supported PDS techniques can be found in Table 4.1.
zero_point (optional)	The zero point of the PDS time trace. It needs to be provided only when the PDS time trace is very noisy and, therefore, the automatic determination of the zero point (Chapter 1.5) fails.
noise_std (optional)	The standard deviation of noise in the PDS time trace. It needs to be provided only when the quadrature component of the PDS signal is not available (Chapter 1.5).

Depending on technique, different experimental parameters must be specified within the curly brackets. The list of parameters for each of the supported PDS techniques can be found in Table 4.1.

Table 4.1. Supported PDS techniques and the corresponding parameters.

Technique		**Parameters**	
Keyword	**Description**	**Keyword**	**Description**
"4pELDOR-rect"	Four-pulse PELDOR with a rectangular pump pulse	magnetic_field	The value of the magnetic field $\vec{B}_0$ in [T].
		detection_frequency	The frequency of detection pulses in [GHz].
		detection_pulse_lengths = [$t_{\pi/2}$, t_{π}]	The lengths of detection π/2- and π-pulses in [ns].
		pump_frequency	The frequency of a pump pulse in [GHz].
		pump_pulse_lengths = [t_{pump}]	The length of a pump pulse in [ns].
"4pELDOR-chirp"	Four-pulse PELDOR with a chirp pump pulse	magnetic_field	Same as above
		detection_frequency	Same as above
		detection_pulse_lengths	Same as above
		pump_frequency	Same as above
		pump_pulse_lengths	Same as above
		pump_frequency_width	The frequency sweep width of a chirp pump pulse in [GHz].
		pump_pulse_rise_times = [t_{rise}]	The rise time of a chirp pump pulse in [ns].
		critical_adiabaticity	The critical adiabaticity of a chirp pump pulse.
"5pRIDME-rect"	Five-pulse RIDME with rectangular pulses	magnetic_field	Same as above
		detection_frequency	Same as above
		detection_pulse_lengths	Same as above
		mixing_time	The length of a mixing inter-pulse interval in [μs].
		temperature	The temperature of the PDS experiment.

4.3 EPR parameters of a spin system

The EPR parameters of a spin system are specified under keyword spins:

```
spins = (
   {
       g = [2.0104, 2.0073, 2.0033];
       gStrain = [0.0004, 0.0003, 0.0001];
       n = (1);
       I = (1.0);
       Abund = (1.0);
       A = ([8.0, 6.0, 96.0]);
       AStrain = ([0.0, 0.0, 12.0]);
       lwpp = 22.4;
       T1 = 0;
       g_anisotropy_in_dipolar_coupling = 0;
   },
   {
       g = [2.0104, 2.0073, 2.0033];
       gStrain = [0.0004, 0.0003, 0.0001];
       n = (1);
       I = (1.0);
       Abund = (1.0);
       A = ([8.0, 6.0, 96.0]);
       AStrain = ([0.0, 0.0, 12.0]);
       lwpp = 22.4;
       T1 = 0;
       g_anisotropy_in_dipolar_coupling = 0;
   }
);
```

Source: PeldorFit/examples/example_s1/config_example_s1.cfg

The EPR parameters of spins A, B, C, *etc.* (Figure 1.2) are given one after the other separated by curly brackets. Even if some of the spins have identical EPR parameters, the corresponding EPR parameters must be given for each spin separately. In the above example, the EPR parameters of two spins, namely spin A and spin B, are specified. Both spins correspond to the nitroxide radical.

To calculate the resonance frequencies of the spins (ω_A and ω_B in Chapter 1.6.2), PeldorFit uses the assumption that the EPR spectrum of each electron spin is determined by the Zeeman interaction with an applied static magnetic field $\vec{B}_0$, the hyperfine interaction between the electron spin and the surrounding nuclear spins, and the inhomogeneous line broadening. The first two interactions are described by *g*- and *A*-tensors, whose three principal axes are assumed to be collinear. Depending on the symmetry of the electron spin center, these tensors can be either isotropic, axial, or orthorhombic. The unresolved anisotropy of the *g*- and *A*-tensors can be taken into account through *g*- and *A*-strains, respectively. *A*-tensor can be specified for several nuclear spins, including several identical nuclear spins. The abundance of different nuclear isotopes can also be taken into account. Inhomogeneous line broadening is approximated by a Gaussian with a certain peak-to-peak linewidth.

The list of EPR parameters required for PeldorFit is given below:

g — *g*-factor. g must contain three values, g = [g_{xx}, g_{yy}, g_{zz}].

gStrain — *g*-strain (the unresolved anisotropy of the *g*-factor). If nonzero, gStrain must contain three values, gStrain = [Δg_{xx}, Δg_{yy}, Δg_{zz}]. Otherwise, the notation gStrain = [] must be used.

n — The number of equivalent nuclear spins coupled to the electron spin. Depending on the number of different types of equivalent nuclear spins, the following notations must be used:

n = [] when no nuclear spins are coupled to the electron spin;

n = [n_1] when n_1 identical nuclear spins coupled to the electron spin;

n = [n_1, n_2] when two groups of identical nuclear spins, consisting of n_1 and n_2 spins, respectively, are coupled to the electron spin;

etc.

I — The nuclear quantum number(s) of the nucleus (nuclei) coupled to the electron spin. I is related to n as follows:

I = [] if n = [];

I = [I_1] if n = [n_1];

I = [I_1, I_2] if n = [n_1, n_2];

etc.

Abund — The natural abundance of the nuclear isotope(s) coupled to the electron spin. Abund is related to n as follows:

Abund = [] if n = [];

Abund = [$Abund_1$] if n = [n_1];

Abund = [$Abund_1$, $Abund_2$] if n = [n_1, n_2];

etc.

A — The hyperfine coupling constant (*A*-constant) in [MHz]. Even if an A-tensor is axial or isotropic, all three principal components of this tensor must be provided. A is related to n as follows:

A = [] if n = [];

A = [A_{1xx}, A_{1yy}, A_{1zz}] if n = [n_1];

A = [A_{1xx}, A_{1yy}, A_{1zz}, A_{2xx}, A_{2yy}, A_{2zz}] if n = [n_1, n_2];

etc.

AStrain — *A*-strain in [MHz] (the unresolved anisotropy of the *A*-tensor). If nonzero, AStrain is related to n as follows:

AStrain = [] if n = [];

AStrain = [ΔA_{1xx}, ΔA_{1yy}, ΔA_{1zz}] if n = [n_1];

AStrain = [ΔA_{1xx}, ΔA_{1yy}, ΔA_{1zz}, ΔA_{2xx}, ΔA_{2yy}, ΔA_{2zz}] if n = [n_1, n_2];

etc.

When the A-strain is zero, the notation AStrain = [] must be used.

lwpp Peak-to-peak linewidth in [MHz].

T1 Longitudinal (T_1) relaxation time in [µs]. This parameter is required for the analysis of RIDME time traces only. Otherwise, it can be set to 0.

g_anisotropy_in_dipolar_coupling

If set to 0, the anisotropy of the g-factor is neglected when calculating the dipolar frequencies (Equation 1.1). If set to 1, the anisotropy of the g-factor is taken into account when calculating the dipolar frequencies (Equations 1.9 or 1.10).

4.4 Simulation parameters

If the simulation mode is selected (Chapter 4.1), simulation parameters need to be specified under keyword simulation_parameters:

```
simulation_parameters:
{
    r_mean      = 1.94;
    r_width     = 0.03;
    xi_mean     = 90;
    xi_width    = 10;
    phi_mean    = 180;
    phi_width   = 5;
    alpha_mean  = 180;
    alpha_width = 20;
    beta_mean   = 45;
    beta_width  = 20;
    gamma_mean  = 0;
    gamma_width = 20;
    rel_prob    = 1;
    j_mean      = 0;
    j_width     = 0;
};
```

Source: PeldorFit/examples/example_s1/config_example_s1.cfg

In accordance with their definition, simulation parameters are used to simulate experimental PDS time traces specified in experiments. As described from Chapter 1.7, these parameters consist of model parameters that in turn include the geometric parameters of a spin system and the parameters of an exchange coupling distribution. For convenience, all simulation parameters are divided into 15 categories. The definitions, units, and ranges of the parameters of each category are given in Table 4.2. Depending on the model of the spin system, one or more simulation parameters may correspond to the same category. In the above example,

Table 4.2. 15 categories of the simulation/fitting parameters.

Keyword	Definition	Units	Ranges
r_mean	$<r>$ (Chapter 1.2.1)	nm	[1.5 nm, 16 nm]
r_width	Δr (Chapter 1.2.1)	nm	[0 nm, 16 nm]
xi_mean	$<\xi>$ (Chapter 1.2.1)	degree	[0°, 90°]
xi_width	$\Delta\xi$ (Chapter 1.2.1)	degree	[0°, 90°]
phi_mean	$<\varphi>$ (Chapter 1.2.1)	degree	[0°, 90°]
phi_width	$\Delta\varphi$ (Chapter 1.2.1)	degree	[0°, 90°]
alpha_mean	$<\alpha>$ (Chapter 1.2.1)	degree	[0°, 90°]
alpha_width	$\Delta\alpha$ (Chapter 1.2.1)	degree	[0°, 90°]
beta_mean	$<\beta>$ (Chapter 1.2.1)	degree	[0°, 90°]
beta_width	$\Delta\beta$ (Chapter 1.2.1)	degree	[0°, 90°]
gamma_mean	$<\gamma>$ (Chapter 1.2.1)	degree	[0°, 90°]
gamma_width	$\Delta\gamma$ (Chapter 1.2.1)	degree	[0°, 90°]
rel_prob	w (Chapter 1.2.2)	-	[0, 1]
j_mean	$<J>$ (Chapter 1.7)	MHz	[0, 15] MHz
j_width	ΔJ (Chapter 1.7)	MHz	[0, 15] MHz

simulation parameters corresponds to the two-spin model with the unimodal distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$ and $P(\gamma)$ (Chapter 1.2.1). This model is described by a set of parameters $<r>$, Δr, $<\xi>$, $\Delta\xi$, $<\varphi>$, $\Delta\varphi$, $<\alpha>$, $\Delta\alpha$, $<\beta>$, $\Delta\beta$, $<\gamma>$, and $\Delta\gamma$, each corresponding to a separate category with the keyword r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width, respectively. The parameter w (Chapter 1.2.2) and the corresponding keyword rel_prob are irrelevant for the model considered. Instead of removing rel_prob from simulation_parameters, it must be set to 1. The last two keywords, j_mean and j_width, contain the mean value and the distribution width of the exchange coupling constant (Chapter 1.7). In the absence of exchange interaction between the spin centers, both keywords are set to 0.

The alternative model of a two-spin system is described by the n-modal ($n \geq 2$) distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ (Chapter 1.2.2). The corresponding model parameters consist of n sets of parameters $<r>$, Δr, $<\xi>$, $\Delta\xi$, $<\varphi>$, $\Delta\varphi$, $<\alpha>$, $\Delta\alpha$, $<\beta>$, $\Delta\beta$, $<\gamma>$, and $\Delta\gamma$, and $(n - 1)$ values of w. According to Table 4.2, all $<r>$ values correspond to the category r_mean, all Δr values correspond to the category r_width, *etc*. Thus, the keywords r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width each take n values, while the keyword rel_prob takes $(n - 1)$

values. The multiple values of each category must be separated by commas and enclosed in square brackets. In the case of $n = 2$, simulation_parameters looks like this:

```
simulation_parameters:
{
    r_mean      = [2.15, 2.80];
    r_width     = [0.03, 0.11];
    xi_mean     = [  30,   65];
    xi_width    = [   7,   13];
    phi_mean    = [   0,    0];
    phi_width   = [   0,    0];
    alpha_mean  = [   0,    0];
    alpha_width = [   0,    0];
    beta_mean   = [  20,    5];
    beta_width  = [   5,    1];
    gamma_mean  = [   4,    6];
    gamma_width = [   0,    0];
    rel_prob    = 0.52;
    j_mean      = 0.0;
    j_width     = 0.0;
};
```

Source: PeldorFit/examples/example_s2/config_example_s2.cfg

Here, the first and second values in r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width describe the first and second modes in the corresponding distributions, respectively. The relative weights of the modes are determined by rel_prob and equal to 52 % and 48 %.

In the case of the m-spin ($m \geq 3$) model, the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ have to be specified for (m - 1) spin pairs (Chapter 1.2.3). When all distributions are approximated by Gaussian or uniform distributions, the model is described by (m - 1) sets of parameters $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$. In this case, the keywords r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width each take (m - 1) values. In contrast to the previous example, the multiple values of the same category are enclosed in round brackets. In the case of $m = 3$, simulation_parameters looks like this:

```
simulation_parameters:
{
    r_mean      = (1.58, 3.10);
    r_width     = (0.02, 0.08);
    xi_mean     = (  29,   50);
    xi_width    = (  10,   13);
    phi_mean    = (   0,   21);
    phi_width   = (   0,    4);
    alpha_mean  = (  61,   73);
    alpha_width = (  19,    1);
    beta_mean   = (  61,   90);
    beta_width  = (   0,    2);
    gamma_mean  = (   0,    0);
    gamma_width = (   0,    0);
    rel_prob    = ( 1.0,  1.0);
    j_mean      = 0;
```

```
    j_width     = 0;
};
```

Source: PeldorFit/examples/example_s3/config_example_s3.cfg

Here, the first and second values in r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width correspond to the spin pairs A-B and A-C, respectively. Similar to the first example, the parameter w is irrelevant, and the corresponding keyword rel_prob is set to 1 for each of (m - 1) spin pairs.

Note that exchange coupling constant is not supported neither for the two-spin model with n-modal distributions nor for the m-spin model (Chapter 1.6). Therefore, the values of j_mean and j_width are irrelevant and set to 0.

To distinguish different $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$ provided in r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width, respectively, PeldorFit assigns them two indices. The first index corresponds to the number of the spin pair (1 = A-B, 2 = A-C, etc.), while the second index corresponds to the number of the conformational ensemble (mode). For example, the notations r_mean = 1.94, r_mean = [2.15, 2.68], and r_mean = (1.58, 3.10) correspond to r_mean = $\langle r_{1,1} \rangle$, r_mean = [$\langle r_{1,1} \rangle$, $\langle r_{1,2} \rangle$], and r_mean = ($\langle r_{1,1} \rangle$, $\langle r_{2,1} \rangle$), respectively. The numbering of other simulation parameters is done analogously.

4.5 Fitting parameters

If the fitting mode or the error the analysis mode is selected (Chapter 4.1), fitting parameters need to be specified under keyword fitting_parameters:

```
fitting_parameters:
{
    r_mean      : {optimize = ([1]); range = ([1.50, 2.50]); value = ()    };
    r_width     : {optimize = ([1]); range = ([0.00, 0.10]); value = ()    };
    xi_mean     : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    xi_width    : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    phi_mean    : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    phi_width   : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    alpha_mean  : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    alpha_width : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    beta_mean   : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    beta_width  : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    gamma_mean  : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    gamma_width : {optimize = ([1]); range = ([ 0.0, 90.0]); value = ()    };
    rel_prob    : {optimize = ([0]); range = ();             value = (1.0)};
    j_mean      : {optimize = ([0]); range = ();             value = (0.0)};
    j_width     : {optimize = ([0]); range = ();             value = (0.0)};
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

In accordance with their definition, fitting parameters are variables whose values are optimized during the fitting of experimental PDS time traces specified in experiments. Similar to the simulation parameters (Chapter 4.4), fitting parameters consist of model parameters that in turn include the geometric parameters of a spin system and the parameters of an exchange coupling distribution (Chapter 1.8). Moreover, in complete analogy to the simulation parameters, all fitting parameters are divided into 15 categories. The definitions, units, and ranges of the parameters of each category are given in Table 4.2. Each line in fitting_parameters corresponds to one category and contains three keywords called optimize, range, and value. The optimize keyword allows to specify which model parameters will be the fitting parameters, i.e., the parameters that will be optimized by PeldorFit. The ranges in which the fitting parameters will be optimized are specified under the range keyword. The value keyword allows to specify constant values for the model parameters that were excluded from the fitting parameters. The number of values in optimize, range, and value depends on the model selected by the user. In the above example, fitting_parameters is given for the model of a two-spin system with the unimodal distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$ and $P(\gamma)$ (Chapter 1.2.1). This model is described by a set of parameters $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$, each corresponding to a separate category with the keyword r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width, respectively. Thus, there is one model parameter per each category. In this case, optimize can be defined in one of the following ways:

optimize = ([1])	The corresponding model parameter is included in the fitting parameters and will be optimized in the ranges specified in range.
optimize = ([0])	The corresponding model parameter is excluded from the fitting parameters and will be set to a constant value specified in value.

In the example, optimize is set to 1 for r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width. Therefore, the corresponding model parameters $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$ will be the fitting parameters. In contrast, optimize is set to 0 for rel_prob, because the corresponding parameter w (Chapter 1.2.2) is irrelevant for the model considered. Moreover, optimize is set to 0 for j_mean and j_width, which contain the mean value $\langle J \rangle$ and the distribution width ΔJ of the exchange coupling constant, respectively (Chapter 1.7).

Having set the fitting parameters, range is specified as follows:

range = ([*lb, ub*]) If optimize = ([1]), the corresponding model parameter will be optimized in the range [*lb, ub*]. The units of *lb* and *ub* depend on the parameter category (Table 4.2).

range = () If optimize = ([0]), no ranges are required.

Lastly, value is specified as follows:

value = (*v*) If optimize = ([0]), the corresponding model parameter will be set to a constant value, *v*. The units of *v* depend on the parameter's category (Table 4.2).

value = () If optimize = ([1]), no value is required.

Next, more sophisticated models are considered. One of them is the two-spin model with the *n*-modal ($n \geq 2$) distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ (Chapter 1.2.2). An example of fitting_parameters for such a model with $n = 2$ is given below:

```
fitting_parameters:
{
    r_mean      : {optimize = ([1,1]); range = ([2.00, 2.50], [2.50, 3.00]); value = ()        };
    r_width     : {optimize = ([1,1]); range = ([0.00, 0.20], [0.00, 0.30]); value = ()        };
    xi_mean     : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    xi_width    : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    phi_mean    : {optimize = ([0,0]); range = ();                           value = (0.0, 0.0)};
    phi_width   : {optimize = ([0,0]); range = ();                           value = (0.0, 0.0)};
    alpha_mean  : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    alpha_width : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    beta_mean   : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    beta_width  : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    gamma_mean  : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    gamma_width : {optimize = ([1,1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    rel_prob    : {optimize = ([1]);   range = ([ 0.0,  1.0]);               value = ()        };
    j_mean      : {optimize = ([0]);   range = ();                           value = (0.0)     };
    j_width     : {optimize = ([0]);   range = ();                           value = (0.0)     };
};
```

Source: PeldorFit/examples/example_f2/config_example_f2.cfg

In the example, all six distributions are bimodal. The model parameters consist of two sets of parameters $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$, plus one value of w. According to Table 4.2, both $\langle r \rangle$ values correspond to the category r_mean, both Δr values correspond to the category r_width, *etc*. Consequently, optimize takes two values for r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, gamma_width, and one value for rel_prob. Multiple values of optimize that correspond to the same category but different modes must be separated by a comma and enclosed first in square brackets and then in round brackets. In turn, each individual value of optimize can be either 1, which means that the corresponding parameter is included in the fitting parameters, or 0, which means that the corresponding parameter is excluded from the fitting parameters. In the above example, all model parameters except $\langle \varphi \rangle$ and $\Delta \varphi$ are set to be the fitting parameters. The reason to exclude $\langle \varphi \rangle$ and $\Delta \varphi$ from the fitting parameters is the axial symmetry of the *g*- and *A*-tensors of spin A (Table 1.1). $\langle J \rangle$ and ΔJ are also excluded from the

fitting, because exchange coupling constant is not supported for the two-spin model with *n*-modal distributions (Chapter 1.6). The values of range and value are specified in accordance with the optimize values. When optimize is set to 1 for a particular model parameter, the ranges in which this parameter will be optimized must be set under range. In contrast, when optimize is set to 0, a constant value of the model parameter must be set under value. When optimize contains two values, as e.g. in the case of r_mean, four different options for range and value exist. All options are considered below using the example of r_mean:

1) r_mean : {optimize = ([1,1]); range = ([2.00, 2.50], [2.50, 3.00]); value = () };
   Both <*r*>s are the fitting parameters. The ranges, in which each of <*r*> will be optimized, are provided under range. No values need to be provided under value.
2) r_mean : {optimize = ([1,0]); range = ([2.50, 3.00]); value = (2.68)};
   The first <*r*> is the fitting parameter, while the second <*r*> is not. The ranges, in which the first <*r*> will be optimized, are provided one after the other under range. The constant value of the second <*r*> is provided under value.
3) r_mean : {optimize = ([0,1]); range = ([2.00, 2.50]), value = (2.68)};
   The second <*r*> is the fitting parameter, while the first <*r*> is not. The ranges, in which the second <*r*> will be optimized, are provided under range. The constant value of the first <*r*> is provided under value.
4) r_mean : {optimize = ([0,0]); range = (); value = (2.15, 2.68)};
   Both <*r*> are not the fitting parameters. The constant values of both <*r*> are provided under value. No values need to be provided under range.

Another model that requires separate consideration is the *m*-spin ($m \geq 3$) model (Chapter 1.2.3). An example of fitting_parameters for such a model with $m = 3$ is given below:

```
fitting_parameters:
{
    r_mean      : {optimize = ([1],[1]); range = ([1.00, 2.00], [2.50, 3.50]); value = ()        };
    r_width     : {optimize = ([1],[1]); range = ([0.00, 0.10], [0.00, 0.20]); value = ()        };
    xi_mean     : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    xi_width    : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    phi_mean    : {optimize = ([0],[1]); range = ([ 0.0, 90.0]);               value = (0.0)     };
    phi_width   : {optimize = ([0],[1]); range = ([ 0.0, 90.0]);               value = (0.0)     };
    alpha_mean  : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    alpha_width : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    beta_mean   : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    beta_width  : {optimize = ([1],[1]); range = ([ 0.0, 90.0], [ 0.0, 90.0]); value = ()        };
    gamma_mean  : {optimize = ([0],[0]); range = ();                           value = (0.0, 0.0)};
    gamma_width : {optimize = ([0],[0]); range = ();                           value = (0.0, 0.0)};
    rel_prob    : {optimize = ([0],[0]); range = ();                           value = (1.0, 1.0)};
    j_mean      : {optimize = ([0]);     range = ();                           value = (0.0)     };
    j_width     : {optimize = ([0]);     range = ();                           value = (0.0)     };
};
```

Source: PeldorFit/examples/example_f3/config_example_f3.cfg

For the three-spin system, the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ have to be specified for two spin pairs, namely the spin pairs A-B and A-C. When all distributions are

approximated by Gaussian or uniform distributions, the model is described by two sets of parameters $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$, each describing one of the spin pairs. According to Table 4.2, both $\langle r \rangle$ values correspond to the category r_mean, both Δr values correspond to the category r_width, *etc*. Consequently, optimize takes two values for r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width. Multiple values of optimize that correspond to the same category but different spin pairs must be separated by a comma and enclosed first in square brackets and then in round brackets. As above, each individual value of optimize can be either 1, which means that the corresponding parameter is included in the fitting parameters, or 0, which means that the corresponding parameter is excluded from the fitting parameters. In the above example, both values of $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, and $\Delta \beta$, as well as the values of $\langle \varphi \rangle$ and $\Delta \varphi$ for the A-C spin pair are set to be the fitting parameters. In contrast, the parameters $\langle \gamma \rangle$ and $\Delta \gamma$ of both spin pairs and the parameters $\langle \varphi \rangle$ and $\Delta \varphi$ of the A-B spin pair are excluded from the fitting parameters. The parameters $\langle \gamma \rangle$ and $\Delta \gamma$ are excluded because of the axial symmetry of the *g*- and *A*-tensors of spins B and C (Table 1.1). Similarly, the parameters $\langle \varphi \rangle$ and $\Delta \varphi$ are excluded because of the axial symmetry of the *g*- and *A*-tensors of spin A (Table 1.1). Note, however, that the last two parameters are excluded only for the A-B spin pair, but not for the A-C spin pair. This is due to the following reason: to compute the distance between the spins B and spin C, PeldorFit uses angle $\varphi_{AC} - \varphi_{AB}$, where φ_{AC} and φ_{AB} are the φ angles of the A-B and A-C spin pairs, respectively. Since one can rotate the coordinate system of the model such that $\varphi_{AB} = 0$, angle $\varphi_{AC} - \varphi_{AB}$ is solely determined by φ_{AC}. Consequently, $\langle \varphi \rangle$ and $\Delta \varphi$ of the A-B spin pair can be set to 0, whereas $\langle \varphi \rangle$ and $\Delta \varphi$ of the A-C spin pair need to be optimized. The parameters *w*, $\langle J \rangle$, and ΔJ are irrelevant for the considered model and, therefore, excluded from the fitting parameters too. Finally, the values of range and value are specified using the same guidelines as mentioned in the first and second examples of this Chapter.

To distinguish different $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, and $\Delta \gamma$ provided in r_mean, r_width, xi_mean, xi_width, phi_mean, phi_width, alpha_mean, alpha_width, beta_mean, beta_width, gamma_mean, and gamma_width, respectively, PeldorFit assigns them two indices. The first index corresponds to the number of the spin pair (1 = A-B, 2 = A-C, etc.), while the second index corresponds to the number of the conformational ensemble (mode). Different notations are considered below using the example of r_mean:

1) r_mean : {optimize = ([1]); range = ([1.50, 2.50]); value = ()};

   This line defines the fitting parameter $\langle r_{1,1} \rangle$.

2) r_mean : {optimize = ([1,1]); range = ([2.00, 2.50], [2.50, 3.00]); value = () };

This line defines the fitting parameters $\langle r_{1,1} \rangle$ and $\langle r_{1,2} \rangle$.

3) r_mean : {optimize = ([1],[1]); range = ([1.00, 2.00], [2.50, 3.50]); value = ()};

This line defines the fitting parameters $\langle r_{1,1} \rangle$ and $\langle r_{2,1} \rangle$.

The numbering of other fitting parameters is done analogously.

4.6 Fitting settings

Fitting setting need to be specified only if the fitting mode or the error the analysis mode is selected (Chapter 4.1). These settings are provided under keyword fitting_settings:

```
fitting_settings:
{
    optimization_method = "ga_local";
    parameters : {number_of_runs = 5,
                  number_of_generations = 1000,
                  generation_size = 128,
                  crossover_probability = 0.5,
                  mutation_probability = 0.01,
                  crossover_probability_increment = 0,
                  mutation_probability_increment = 0.01,
                  parent_selection = "tournament",
                  nelder_mead_maxiter = 200};
    display_graphics = 0;
    goodness_of_fit = "chi2";
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

They include:

optimization_method	Optimization method used for the fitting. Available optimization methods can be found in Table 4.3.
parameters	Intrinsic parameters of the optimization method. The list of the intrinsic parameters can be found in Table 4.3.
display_graphics	If set to 1, the picture of a fit (Chapter 5.2) will be displayed after each iteration of the optimization algorithm. If set to 0, the above picture will be displayed only at the end of the optimization.
goodness_of_fit	Scoring parameter used to determine the goodness of fit. Available options are given below: "chi2" - χ^2 (Equation 1.26) "reduced_chi2" - reduced χ^2, i.e., χ^2 normalized by $(N - P)$, here N is the total number of data points in the PDS time traces, and P is the total number of fitting and background parameters. "chi2_noise_std_1" - χ^2 with $\sigma_{N,k} = 1$ (Equation 1.26)

Table 4.3. Optimization methods and their intrinsic parameters.

Optimization method		Parameters	
Keyword	**Description**	**Keyword**	**Description**
"ga"	GA (Chapter 1.7.1)	number_of_runs	Number of GA runs
		number_of_generations	Maximum number of generations (N_g)
		generation_size	Number of chromosomes per generation (N_c)
		crossover_probability	crossover probability (p_c)
		mutation_probability	mutation probability (p_m)
		crossover_probability_increment	The increment of the crossover probability (Δp_c)
		mutation_probability_increment	The increment of the mutation probability (Δp_m)
		parent_selection	The method of parent selection. Currently, only the tournament selection (keyword "tournament") is supported.
"ga_local"	GA (Chapter 1.7.1) and Nelder-Mead algorithm (scipy)	number_of_runs	Same as above
		number_of_generations	Same as above
		generation_size	Same as above
		crossover_probability	Same as above
		mutation_probability	Same as above
		crossover_probability_increment	Same as above
		mutation_probability_increment	Same as above
		parent_selection	Same as above
		nelder_mead_maxiter	The maximum number of iterations in the Nelder-Mead algorithm

The discussion of the main optimization method, the genetic algorithm (GA), can be found in Chapter 1.7.1. Chapter 1.7.1 also provides some empirical rules that may help to choose the values of the GA's intrinsic parameters optimally. χ^2 should be used as a scoring parameter except in cases when there are special reasons for using another parameter.

4.7 Error analysis parameters

As described in Chapter 1.9, the error analysis is based on recording χ^2 for different subspaces of fitting parameters, usually consisting of one or two fitting parameters. The parameters that form different subspaces are specified under keyword error_analysis_parameters:

```
error_analysis_parameters:
{
    parameters = (["r_mean", "r_width"],
                  ["xi_mean", "xi_width"],
                  ["phi_mean", "phi_width"],
                  ["alpha_mean", "alpha_width"],
                  ["beta_mean", "beta_width"],
                  ["gamma_mean", "gamma_width"]);
    spin_pairs = ([1,1], [1,1], [1,1], [1,1], [1,1], [1,1]);
    components = ([1,1], [1,1], [1,1], [1,1], [1,1], [1,1]);
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

As described at the end of Chapter 4.6, each fitting parameters has three identifiers: 1) the category, 2) the spin pair number, and 3) the conformational ensemble (mode) number. In error_analysis_parameters, these three identifiers are provided under the keywords parameters, spin_pairs, and components, respectively. Identifiers corresponding to different parameter subspaces are separated by commas and enclosed in round brackets. Identifiers corresponding to the same parameter subspace are also separated by commas but enclosed in square brackets. The order in which identifiers of different fitting parameters are provided must be the same for parameters, spin_pairs, and components. In the above example, χ^2 is recorded for six two-dimensional subspaces of the fitting parameters, namely $\chi^2(<r_{1,1}>, \Delta r_{1,1})$, $\chi^2(<\xi_{1,1}>, \Delta\xi_{1,1})$, $\chi^2(<\varphi_{1,1}>, \Delta\varphi_{1,1})$, $\chi^2(<\alpha_{1,1}>, \Delta\alpha_{1,1})$, $\chi^2(<\beta_{1,1}>, \Delta\beta_{1,1})$, and $\chi^2(<\gamma_{1,1}>, \Delta\gamma_{1,1})$. Each of these dependencies is recoded with the number of points specified in error_analysis_settings (under the keyword sample_size) and with the fitting parameters' ranges specified in fitting_parameters (under the keyword range).

Note that when the fitting mode (Chapter 4.1) is selected, the error analysis automatically starts after the fitting. Alternatively, the fitting can be performed first without the error analysis and, if necessary, the error analysis can be performed at any time later using the error analysis mode (Chapter 4.1). To skip the error analysis, error_analysis_parameters should be defined as follows:

```
error_analysis_parameters:
{
    parameters = ();
    spin_pairs = ();
    components = ();
};
```

4.8 Error analysis settings

Error analysis settings need to be specified only if the fitting mode or the error the analysis mode is selected (Chapter 4.1). These settings are provided under keyword error_analysis_settings:

```
error_analysis_settings:
{
    sample_size = 10000;
    confidence_interval = 3;
    filepath_optimized_parameters = "";
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

They include:

sample_size	The number of samples used to record the dependence of χ^2 on each single subset of fitting parameters.
confidence_interval	Confidence level in [σ] used to estimate the error of the optimized fitting parameters (Chapter 1.9). For example, confidence_interval = 1 sets the confidence level to 1σ (68 %), confidence_interval = 2 sets the confidence level to 2σ (95 %), and confidence_interval = 3 sets the confidence level to 3σ (99.7 %). The use of the confidence levels is based on the assumption that the errors of optimized fitting parameters have Gaussian distributions. In practice, the error distributions may deviate from a Gaussian, but their approximation by a Gaussian should still yield reasonable error estimates.
filepath_optimized_parameters	The path to the fitting_parameters.dat file of PeldorFit that contains the optimized fitting parameters (Chapter 5.2). This path need to be provided only when the error analysis mode is selected (Chapter 4.1). Otherwise, it must be initialized with an empty string, as in the above example.

4.9 Calculation settings

Calculation settings provide the user additional control over the simulation of PDS time traces. These settings are specified under keyword calculation_settings:

```
calculation_settings:
{
    integration_method = "monte_carlo"
    mc_sample_size = 100000;
    distributions : {r = "normal",
                     xi = "vonmises",
                     phi = "vonmises",
                     alpha = "vonmises",
                     beta = "vonmises",
                     gamma = "vonmises",
                     j = "normal"};
    excitation_treshold = 0.001;
    euler_angles_convention = "ZXZ";
    background_model = "exp";
    background_parameters = {
       decay_constant : {optimize = 1; range = [0, 1]; value = 0.05};
       scale_factor   : {optimize = 1; range = [0, 1]; value = 1   };
    };
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

They include:

integration_method	Numerical integration method used in the calculation of PDS time traces (Equation 1.23). Currently, only the Monte-Carlo method (Chapter 1.7), which corresponds to the keyword "monte_carlo", is supported.
mc_sample_size	The number of Monte-Carlo samples, N_{MC} (Chapter 1.7). This parameter determines the precision of the numerical integration performed via Equation 1.23. For high precision, N_{MC} must be set as high as possible. On the downside, the calculation time increases linearly with N_{MC}, which sets some limits on the practically realizable values of N_{MC}. Our test showed that $N_{MC} = 10^6$ yields good compromise between the precision and the calculation time. The use of N_{MC} below 10^5 is not recommended, because of the corresponding precision is low.
distributions	Distribution functions used for $P(r)$ (keyword r), $P(\xi)$ (keyword xi), $P(\varphi)$ (keyword phi), $P(\alpha)$ (keyword alpha), $P(\beta)$ (keyword beta), $P(\gamma)$ (keyword gamma), and $P(J)$ (keyword j). The definition of r, ξ, φ, α, β, γ, and J can be found in Chapters 1.2 and 1.6. The distributions $P(r)$ and $P(J)$ can be described by

	either a uniform distribution (keyword "uniform") or a Gaussian distribution (keyword "normal"). All angular distributions, namely $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$, can be described by either a uniform distribution (keyword "uniform") or a von Mises distribution (keyword "vonmises").
excitation_threshold	Threshold for $p_{det}(\omega)$ and $p_{pump}(\omega)$ (Equation 1.11), below which both probabilities are considered to be negligibly small.
euler_angles_convention	Euler angle convention for angles α, β, and γ (Chapter 1.2). By default, the ZXZ (keyword "ZXZ") convention must be used. Alternatively, the ZYZ (keyword "ZYZ") convention or any other can be selected. Note that the use of capital letters in the name of convention is important.
background_model	The background model ($B(t)$ in Chapter 1.6). The list of supported background models can be found in Table 4.4.
background_parameters	The parameters of the background model specified in background_model. The list of parameters for each of the supported background models can be found in Table 4.4. Note that the modulation depth scale factor η appears in Table 4.4 along with the background parameters, although in fact it is not a background parameter. The reasons for this are given below. As mentioned in Chapters 1.7 and 1.8, the background parameters and η can be can be either specified directly by the user or optimized by PeldorFit within the ranges defined by the user. In the latter case, the optimization goal is to find such a $B(t)$ and η that will minimize the difference between the experimental PDS time trace(s) and the corresponding simulated PDS time trace(s). To enable both options, each background parameter is described by three keywords: optimize, range, and value. The meaning of these keywords is almost identical to the meaning of the corresponding keywords used for the fitting parameters (Chapter 4.6). The optimize keyword allows to specify whether the corresponding background parameters will be optimized or not. If optimize = 1, the corresponding background parameter will be optimized for each individual time trace (specified in

experiments) in the range specified under range and using the value specified under value as an initial value ("first guess"). The ranges and initial values suggested for the background parameters are given in Table 4.4. If optimize = 0, the corresponding background parameter will be set to a constant value specified under value. In this case, range should be set to an empty array, i.e., range = [].

Table 4.4. Background models and their parameters.

Background model		**Background parameters**			
Keyword	**Description**[a]	**Keyword**	**Description**[a]	**Range**	**Initial value**
"exp"	exponential decay	decay_constant	k	[0, 1]	0.05
		scale_factor	η	[0, 1]	1
"stretched_exp"	stretched exponential decay	decay_constant	k	[0, 1]	0.05
		dimension	d	[1, 5]	3
		scale_factor	η	[0, 1]	1
"polynom2"	2nd order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"polynom3"	3rd order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		c3	c_3	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"polynom4"	4th order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		c3	c_3	[-2, 2]	0
		c4	c_4	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"keller"	Keller's exponential decay	k1	k_1	[-1, 1]	0
		k2	k_2	[-1, 1]	0
		scale_factor	η	[0, 1]	1

[a] For definitions, see Table 1.3 and Chapter 1.7.

4.10 Output settings

Output settings are specified under keyword output:

```
output:
{
    directory = "";
    save_data = 1;
    save_figures = 1;
};
```

Source: PeldorFit/examples/example_f1/config_example_f1.cfg

They include:

directory	A directory in which the output data will be stored. When initialized by an empty string, as in the above example, the directory of the configuration file will be used.
save_data	If set to 1, all ASCII files generated by PeldorFit will be saved in a separate folder in directory. Otherwise, this flag should be set to 0.
save_figures	If set to 1, all pictures generated by PeldorFit will be saved in a separate folder in directory. Otherwise, this flag should be set to 0.

5 Output data

The output data of PeldorFit is stored in two forms: text files with the *.dat* extension and graphics files with the *.png* extension. The detailed description of all output files for different operation modes is given below.

5.1 Simulation output

Output data generated in the simulation mode (see Chapter 1.7 and Chapter 4) consist of:

5.1.1 Text files

Filename: logfile

Description: This file contains the messages displayed by PeldorFit at execution.

Content: Progress messages.

Filename: time_trace_X.dat

Description: This file contains an experimental PDS time trace and a corresponding simulated PDS time trace. X is the name of a PDS experiment specified in experiments.

Content: Column 1: time points in [µs].
Column 2: the in-phase component of the experimental PDS time trace.
Column 3: the simulated PDS time trace.
Column 4: the quadrature component of the experimental PDS time trace.

Filename: background_X.dat

Description: This file contains an experimental PDS time trace and its simulated background. X is the name of a PDS experiment specified in experiments.

Content: Column 1: time points in [µs].
Column 2: the in-phase component of the experimental PDS time trace.
Column 3: the simulated background.
Column 4: the quadrature component of the experimental PDS time trace.

Filename: background_parameters.dat

Description: This file contains the values of the background parameters. These parameters are specified in background_parameters and used to obtain the backgrounds stored in background_X.dat.

Content: Column 1: the names of the PDS experiments specified in experiments.
Column 2: the values of 1st background parameter specified in background_parameters.
...
Column N+1: the values of N^{th} background parameter specified in background_parameters.
(N is the total number of the background parameters.)

Filename: epr_spectrum_X.dat

Description: This file contains the simulated EPR spectrum of a spin system specified in spins. X is the name of a PDS experiment specified in experiments. The spectrum is calculated in the frequency domain using the magnetic field value specified in magnetic_field.

Content: Column 1: frequencies in [GHz].
Column 2: the simulated EPR spectrum.

Filename: detection_bandwidth_X.dat

Description: This file contains an excitation profile of detection pulses, $p_{det}(\omega)$ (Chapter 1.6.2). X is the name of a PDS experiment specified in experiments.

Content: Column 1: frequencies in [GHz].
Column 2: the relative probabilities of excitation.

Filename: pump_bandwidth_X.dat (only for PELDOR experiments)

Description: This file contains an excitation profile of a pump pulse, $p_{pump}(\omega)$ (Chapter 1.6.2). X is the name of a PDS experiment specified in experiments.

Content: Column 1: frequencies in [GHz].
Column 2: the relative probabilities of excitation.

5.1.2 Graphics files

Filename: time_traces.png

Description: In this picture, the experimental PDS time traces are overlaid with the corresponding simulated PDS time traces.

Filename: bandwidths.png

Description: In this picture, the EPR spectrum of the spin system is overlaid with the excitation profiles of the detection and pump pulses.

Filename: backgrounds.png

Description: In this picture, the experimental PDS time traces are overlaid with corresponding backgrounds.

5.2 Fitting output

Output data generated in the fitting mode and partly in the error analysis mode (Chapter 1.7, Chapter 1.9, and Chapter 4) consist of:

5.2.1 Text files

Filename: logfile

Description: This file contains the messages displayed by PeldorFit at execution.

Content: Progress messages.

Filename: score.dat

Description: This file contains the goodness of fit recorded as a function of optimization step.

Content: Column 1: The numbers of optimization steps. The total number of optimization steps is specified in fitting_settings.
Column 2: The goodness of fit values, e.g., χ^2 values. The goodness of fit parameter is specified in goodness_of_fit.

Filename: fit_X.dat

Description: This file contains an experimental PDS time trace and its fit. X is the name of a PDS experiment specified in experiments.

Content: Column 1: time points in [µs].
Column 2: the in-phase component of the experimental PDS time trace.
Column 3: the best fit obtained for the PDS time trace in Column 2.
Column 4: the quadrature component of the experimental PDS time trace.

Filename: fitting_parameters.dat

Description: This file contains the optimized values of fitting parameters specified in fitting_parameters.

Content: Column 1: the categories of the fitting parameters.

Column 2: the spin pair numbers of the fitting parameters.

Column 3: the conformational ensemble (mode) numbers of the fitting parameters.

Column 4: the optimization flags set for the fitting parameters.

Column 5: the optimized or fixed values of the fitting parameters.

Column 6: the negative asymmetric errors of the optimized fitting parameters.

Column 7: the positive asymmetric errors of the optimized fitting parameters.

Filename: symmetry_related_solutions.dat

Description: This file contains the symmetry-related values of angular parameters (Chapter 1.4), which are the part of fitting parameters contained in fitting_parameters.dat.

Content: Column 1: the categories of the angular parameters.

Column 2: the spin pair numbers of the angular parameters.

Column 3: the conformational ensemble (mode) numbers of the angular parameters.

Columns 4-19: symmetry-related values of the angular parameters. In addition, the last raw of each column contains the goodness of fit value that corresponds to the set of angular parameters provided above it. The goodness of fit parameter is specified in goodness_of_fit.

Filename: background_X.dat

Description: This file contains an experimental PDS time trace and its simulated background. X is the name of a PDS experiment specified in experiments.

Content: Column 1: time points in [μs].

Column 2: the in-phase component of the experimental PDS time trace.

Column 3: the simulated background.

Column 4: the quadrature component of the experimental PDS time trace.

Filename: background_parameters.dat

Description: This file contains the values of the background parameters. These parameters are specified in background_parameters and are used to obtain the backgrounds stored in background_X.dat.

Content: Column 1: the names of the PDS experiments specified in experiments.

Column 2: 1st background parameter specified in background_parameters.

…

Column N+1: N^{th} background parameter specified in background_parameters.

(N is the total number of background parameters.)

Filename: score_vs_parameters_X.dat

Description: This file contains χ^2 recorded for a subspace of fitting parameters, usually consisting of one or two fitting parameters. The index X indicates that the subspace of fittings parameters is given by the X^{th} set of parameters in error_analysis_parameters.

Content: Column 1: the values of the 1st fitting parameter.

…

Column N: The values of the N^{th} fitting parameter.

Column N+1: χ^2 values.

(N denotes the total number of fitting parameters in the subspace.)

Filename: score_vs_parameter_X.dat

Description: This file contains χ^2 recorded as a function of one fitting parameter. The index X indicates that the fitting parameter corresponds to the X^{th} parameter specified in error_analysis_parameters.

Content: Column 1: the values of the fitting parameter.

Column 2: χ^2 values.

5.2.2 Graphics files

Filename: score.png

Description: The goodness of fit (e.g., χ^2) is depicted as a function of optimization step.

Filename: fits.png

Description: In this picture, the experimental PDS time traces are overlaid with the corresponding fits.

Filename: backgrounds.png

Description: In this picture, the experimental PDS time traces are overlaid with the corresponding backgrounds.

Filename: score_vs_parameters.png

Description: χ^2 is depicted as a function of individual fitting parameters or pairs of fitting parameters. The optimized values of the fitting parameters are shown by white dots.

Filename: confidence_intervals.png

Description: χ^2 is depicted as a function of individual fitting parameters. The optimized values of the fitting parameters are shown by white dots. The χ^2 threshold is depicted by the black dashed line. This threshold consists of two contributions, $\Delta\chi_{ci}^2$, which is determined at the user-defined confidence level, and $\Delta\chi_{ne}^2$, which takes into account the numerical error. The uncertainty ranges of the fitting parameters are shown as gray intervals.

6 Examples

This chapter provides examples of how one can use PeldorFit to simulate or to fit the PDS time traces.

6.1 Simulation example

In this example, the simulation of six orientation-selective PELDOR time traces recorded on nitroxide biradical **1** (Figure 6.1) is considered. The PELDOR time traces were recorded at W-band using six different pump and detection positions in the nitroxide spectrum (Figure 6.1b). The details of the PELDOR experiments can be found elsewhere.[48] The simulation is done through the following steps:

- *Preparation / preview of the configuration file.* The input data for the given simulation is contained in configuration file config_example_s1.cfg located in PeldorFit/examples/example_s1. Detailed information on how to create and interpret such a configuration file can be found in Chapter 4. A brief summary of config_example_s1.cfg is given below. This file begins with the selection of the simulation mode (see mode in Chapter 4.1). The simulation of the PELDOR time traces is divided into the simulation of the PELDOR form factors and the simulation of the PELDOR backgrounds, followed by the calculation of the product between the form factors and the corresponding backgrounds (Chapter 1.7). To

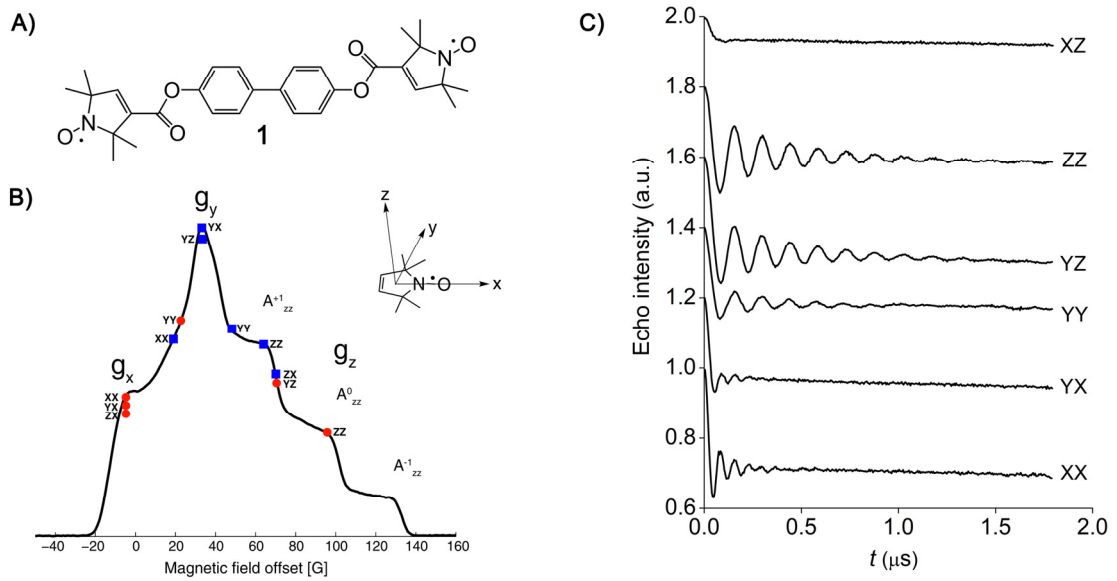

Figure 6.1. A) The chemical structure of nitroxide biradical **1**. B) The echo-detected W-band EPR spectrum of **1**. The detection and pump positions are depicted by red dots and blue squares, respectively. The dots and squares are labelled in accordance with the frequency offsets used in the PELDOR experiments. The inset shows the orientation of nitroxide g-tensor axes with respect to the nitroxide structure. C) The W-band PELDOR time traces of **1**. Adapted from Ref. [48].

Table 6.1. Parameters used to simulate the PELDOR time traces of **1**.

Distribution	Distribution type	Mean value	Width[a]
$P(r)$	Gaussian	1.95 nm	0.03 nm
$P(\xi)$	von Misses	87°	20°
$P(\varphi)$	von Misses	22°	2°
$P(\alpha)$	von Misses	26°	9°
$P(\beta)$	von Misses	90°	10°
$P(\gamma)$	von Misses	32°	10°

[a] The width is defined as a standard deviation.

simulate the form factors, the nitroxide-nitroxide pair of **1** is described by the two-spin model with unimodal distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ (Chapter 1.2.1). $P(r)$ is approximated by the Gaussian distribution, while the five angular distributions are approximated by the von Mises distribution (see distributions in Chapter 4.9). The corresponding mean values and distribution widths are thus used as simulation parameters (see simulation_parameters in Chapter 4.4). The values of all simulation parameters are listed in Table 6.1. In addition to the simulation parameters, the parameters of the PELDOR experiments (see experiments in Chapter 4.2) and the EPR parameters of the two nitroxides (see spins in Chapter 4.3) are used to simulate the form factors. The calculation of the form factors is performed via the Monte-Carlo method with 10^6 samples (see integration_method and mc_sample_size in Chapter 4.9). The backgrounds of the PELDOR time traces are approximated by an exponential decay (see background_model in Chapter 4.9). The corresponding decay constant k and the modulation depth scale factor η are used as background parameters (see background_parameters in Chapter 4.9). Both parameters are optimized to minimize the difference between the experimental and simulated PELDOR time traces (Equation 1.25).

- *Running the program.* To start the simulation using the above configuration file, one has to follow the instructions of Chapter 3. In Linux, the following commands must be entered in Terminal:

`cd [user directory]/PeldorFit` (opens the folder with PeldorFit)

`chmod 755 PeldorFit` (sets the permission properties)

`chmod 755 PeldorFit.sh` (sets the permission properties)

`cd [user directory]/PeldorFit/examples/example_s1` (opens the folder with the configuration file)

`sh [user directory]/PeldorFit/PeldorFit.sh config_example_s1.cfg` (runs PeldorFit)

During the program operation, numerous messages are displayed in Terminal (Linux, macOS) or Command Prompt (Windows). They are also saved in output file logfile. Some of these messages are discussed below:

```
Experiment 'offset XX' was loaded
Phase correction: -0 deg
Zero point: 0.206 us
Noise std: 0.001600
```

These messages correspond to the PELDOR time trace specified in experiments under name 'offset XX'. As described in Chapter 1.5, each PDS time trace is subjected to the preprocessing (block 2 in Figure 1.5), which includes phase correction, the calculation of a zero point, normalization, and the calculation of a noise level. The above messages display the values of phase, zero point, and noise level obtained for the 'offset XX' time trace. Similar messages are also displayed for all other PELDOR time trace specified in experiments.

Next, one should see the following message:

```
Zero is encountered among the standard deviations of noise!
To avoid problems with the scoring, the standard deviation of noise is set to 1 for
all experiments.
```

This message is displayed when the quadrature component of at least one of the PDS time traces specified in experiments is given by zeros (e.g., because it could not be measured). In this case, the noise level cannot be determined from the quadrature component (Chapter 1.5) and, instead, PeldorFit sets the noise level to 1 for all PDS time traces.

Finally, the simulation itself is accompanied by the following messages:

```
Computing the time trace of the experiment 'offset XX'...
Background parameters:
Decay constant:  0.019172190184541672
Scale factor:  1.0314800173488965
Chi2 (noise std = 1): 0.0163
```

These messages display the optimized values of background parameters and the χ^2 value (Equation 1.26) for the PELDOR time trace with the name 'offset XX'. Similar messages are displayed for all other PELDOR time trace specified in experiments.

- *Results overview.* Simulation results are saved into a new folder, which is automatically created in examples/example_s1. They consist of several output files (Chapter 5.1), including three graphics files discussed below. The bandwidths.png file contains an overlay of the simulated EPR spectrum of **1** with the excitation profiles of the detection and pump pulses, $p_{det}(\omega)$ and $p_{pump}(\omega)$, respectively (Figure 6.2a). Such an overlay reveals which spectral components of the nitroxide contribute to the PELDOR time traces recorded with different

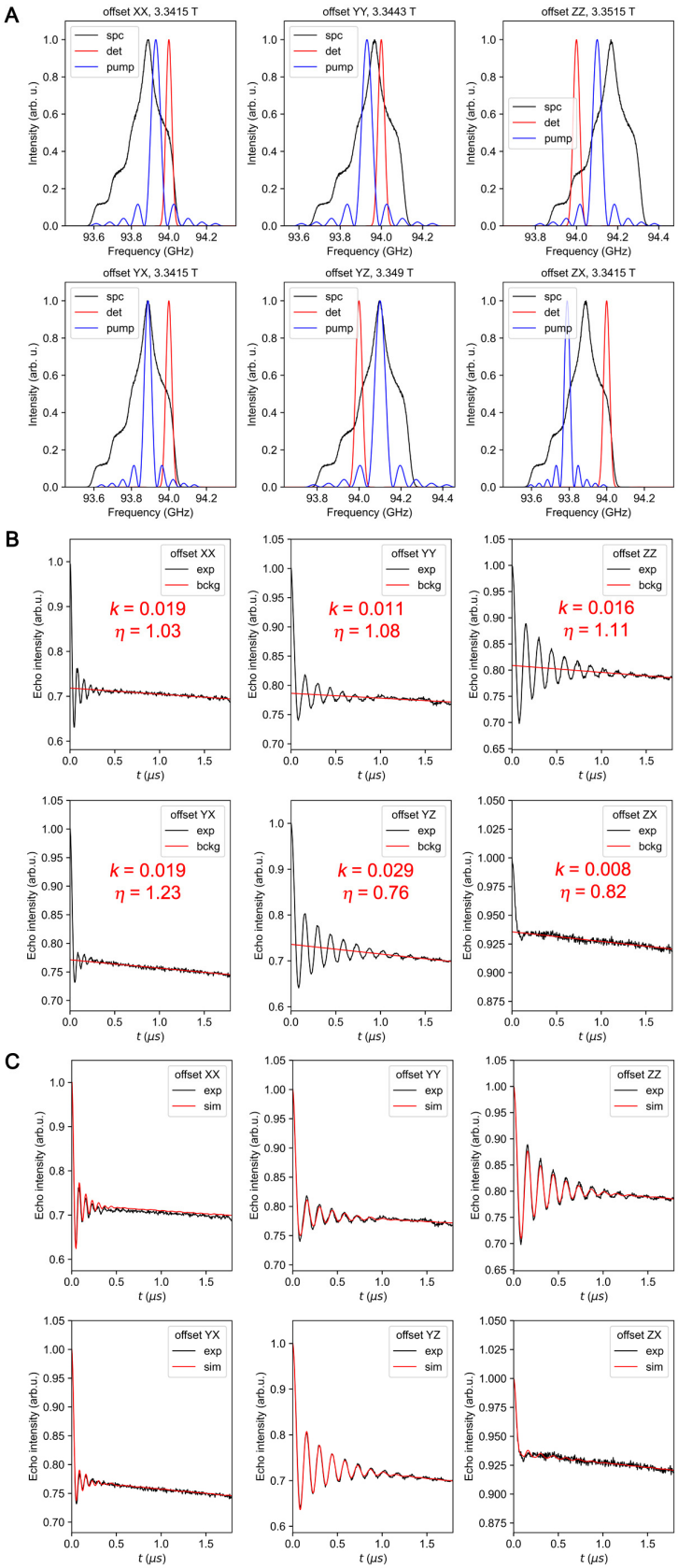

Figure 6.2. A) The simulated EPR spectrum (black) of **1** is overlaid with the excitation profiles of the detection (red) and pump (blue) pulses. B) The experimental PELDOR time traces (black) of **1** are overlaid with the simulated backgrounds (red). The optimized values of background parameters are shown on top of each time trace. C) The experimental PELDOR time traces (black) of **1** are overlaid with the corresponding simulated PELDOR time traces (red).

frequency offsets. In the backgrounds.png file, the experimental PELDOR time traces are overlaid with the simulated backgrounds (Figure 6.2b). As mentioned above, the PELDOR backgrounds are approximated by the exponential decay. The corresponding optimized values of decay constant k and scaling factor η can be found in the background_parameters.dat file (Figure 6.2b). Finally, the time_traces.png file contains an overlay of the experimental and simulated PELDOR time traces (Figure 6.2c). As can be seen from Figure 6.2c, the simulated time traces are almost identical to the experimental ones. Thus, the simulation parameters given in Table 6.1 allow to reproduce the PELDOR data. Based this it is tempting to conclude that these parameters describe well the distance and angular distributions in **1**. Note, however, that the simulation leaves the uncertainty ranges of the simulation parameters unknown. Therefore, some of the simulation parameters and corresponding distributions may have significant errors. To estimate these errors, the fitting of the PELDOR time traces with the subsequent error analysis can be done. This is discussed in the next chapter.

6.2 Fitting examples

6.2.1 Orientation-selective PELDOR on a two-spin system

In this example, the fitting of six orientation-selective PELDOR time traces recorded on nitroxide biradical **1** (Figure 6.1) is considered. The fitting is done through the following steps:

- *Preparation / preview of the configuration file.* The input data for the given fitting is contained in configuration file config_example_f1.cfg located in PeldorFit/examples/example_f1. Detailed information on how to create and interpret such a configuration file can be found in Chapter 4. A brief summary of config_example_f1.cfg is given below. This file begins with the selection of the fitting mode (see mode in Chapter 4.1). To perform the model-based fitting, the nitroxide-nitroxide pair of **1** is described by the two-spin model with unimodal distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ (Chapter 1.2.1). $P(r)$ is approximated by the Gaussian distribution, while the five angular distributions are approximated by the von Mises distribution (see distributions in Chapter 4.9). The corresponding mean values $<r_{1,1}>$, $<\xi_{1,1}>$, $<\varphi_{1,1}>$, $<\alpha_{1,1}>$, $<\beta_{1,1}>$, and $<\gamma_{1,1}>$, as well as distribution widths $\Delta r_{1,1}$, $\Delta\xi_{1,1}$, $\Delta\varphi_{1,1}$, $\Delta\alpha_{1,1}$, $\Delta\beta_{1,1}$, and $\Delta\gamma_{1,1}$ are used as fitting parameters (see fitting_parameters in Chapter 4.4). During the fitting, the values of $<r_{1,1}>$, $\Delta r_{1,1}$, and all angular parameters are optimized in the ranges [1.5, 2.5] nm, [0.0, 0.1] nm, and [0°, 90°], respectively. The fitting is done in two steps: first, 1000 iterations of the GA is performed and, second, the GA solution is refined by 200 iterations of the Nelder-Mead algorithm (see optimization_method and parameters in Section 4.6). At each iteration of the GA and the

Nelder-Mead algorithm, the PELDOR time traces are simulated for the current values of the fitting parameters, after which the goodness of fit is calculated as the χ^2 deviation between the simulated and experimental PELDOR time traces (see goodness_of_fit in Chapter 4.6). The simulation of the PELDOR time traces is divided into the simulation of the PELDOR form factors and the simulation of the PELDOR backgrounds, followed by the calculation of the product between the form factors and the corresponding backgrounds (Chapter 1.7). The form factors are simulated via the Monte-Carlo method with 10^6 samples (see integration_method and mc_sample_size in Chapter 4.9). In addition to the fitting parameters, this simulation uses the parameters of the PELDOR experiments (see experiments in Chapter 4.2) and the EPR parameters of the two nitroxides (see spins in Chapter 4.3). The backgrounds are simulated as exponential decays (see background_model in Chapter 4.9). The corresponding background parameters, the decay constant k and the modulation depth scale factor η (see background_parameters in Chapter 4.9), are optimized for each PELDOR time trace such that the product of the simulated form factor and background yields the smallest χ^2 deviation (Equation 1.36) from the experimental PELDOR time trace. Note that the optimization of the background parameters is done at each iteration of the GA and the Nelder-Mead algorithm and, thus, runs separately from the optimization of the model parameters (fitting parameters). After the fitting, the confidence intervals of the fitting parameters are determined via the error analysis (Chapter 1.9). For this, χ^2 is recorded for six two-dimensional subspaces of the fitting parameters, namely $\chi^2(<r_{1,1}>, \Delta r_{1,1})$, $\chi^2(<\xi_{1,1}>, \Delta\xi_{1,1})$, $\chi^2(<\varphi_{1,1}>, \Delta\varphi_{1,1})$, $\chi^2(<\alpha_{1,1}>, \Delta\alpha_{1,1})$, $\chi^2(<\beta_{1,1}>, \Delta\beta_{1,1})$, and $\chi^2(<\gamma_{1,1}>, \Delta\gamma_{1,1})$ (see error_analysis_parameters in Chapter 4.7). Based on these error surfaces, the 3σ confidence intervals of individual fitting parameters are determined (see confidence_interval in Chapter 4.8).

- *Running the program.* To start the simulation using the above configuration file, one has to follow the instructions of Chapter 3. In Linux, the following commands must be entered in Terminal:

`cd [user directory]/PeldorFit` (opens the folder with PeldorFit)

`chmod 755 PeldorFit` (sets the permission properties)

`chmod 755 PeldorFit.sh` (sets the permission properties)

`cd [user directory]/PeldorFit/examples/example_s1` (opens the folder with the configuration file)

`sh [user directory]/PeldorFit/PeldorFit.sh config_example_f1.cfg` (runs PeldorFit)

During the program operation, numerous messages are displayed in Terminal (Linux, macOS) or Command Prompt (Windows). They are also saved in output file logfile. Some of these messages are discussed below:

```
Experiment 'offset XX' was loaded
Phase correction: -0 deg
Zero point: 0.206 us
Noise std: 0.001600
```

These messages are correspond to the PELDOR time trace specified in experiments under name 'offset XX'. As described in Chapter 1.5, each PDS time trace is subjected to the preprocessing (block 2 in Figure 1.5), which includes phase correction, the calculation of a zero point, normalization, and the calculation of a noise level. The above messages display the values of phase, zero point, and noise level obtained for the 'offset XX' time trace. Similar messages are also displayed for all other PELDOR time trace specified in experiments.

Next, messages showing the fitting progress appear:

```
Starting the optimization via genetic algorithm...
Run 1 / 1, optimization step 1 / 1000: Chi2 = 133750.394071
…
Run 1 / 1, optimization step 1000 / 1000: Chi2 = 13023.074640
The best solution was found in run no. 1 (crossover probability 0.500000, mutation
probability 0.010000)
The optimization is finished. Total duration: 1 day, 2:35:16.885982
Starting the optimization via Nelder-Mead algorithm...
Optimization step 1 / 200: Chi2 = 13220.299597
…
Optimization step 199 / 200: Chi2 = 12985.668696
The optimization is finished. Total duration: 2:50:12.801322
```

In accordance with the configuration file, the fitting consists of 1000 iterations of the GA and 200 iterations of the Nelder-Mead algorithm. After the fitting, a summary of all fitting and background parameters is displayed (not shown). This is followed by the messages showing the progress of the error analysis:

```
Starting the error analysis...
Computing the numerical error...
Numerical error (chi2) = 6.87e+02
Computing the score threshold...
Score threshold (chi2) = 6.96e+02
Computing the score as a function of fitting parameters...
Parameter set 1 / 6
…
```

```
Parameter set 6 / 6
Computing the uncertainty intervals of the optimized fitting parameters...
The error analysis is finished. Total duration: 13:37:43.747142
```

These messages include the estimate of the χ^2 numerical error, $\Delta\chi_{ne}^2$, as well as the χ^2 threshold, $\Delta\chi^2$, which is used to calculate the uncertainty intervals of the fitting parameters (Chapter 1.9). After the error analysis, all fitting parameters are displayed one more time, this time together with the corresponding errors:

```
Optimized fitting parameters:
```

Parameter	No. spin pair	No. component	Optimized	Value	-Error	+Error
r mean (nm)	1	1	yes	1.948	-0.001181	0.004208
r width (nm)	1	1	yes	0.02713	-0.002803	0.001671
xi mean (deg)	1	1	yes	86.77	-13.66	3.163
xi width (deg)	1	1	yes	20.27	-5.784	4.539
phi mean (deg)	1	1	yes	22.08	-7.466	2.728
phi width (deg)	1	1	yes	1.504	-1.302	10.14
alpha mean (deg)	1	1	yes	26.13	-5.509	9.572
alpha width (deg)	1	1	yes	8.535	-8.42	7.162
beta mean (deg)	1	1	yes	90.77	-14.39	0.0
beta width (deg)	1	1	yes	10.05	-9.93	11.87
gamma mean (deg)	1	1	yes	31.91	-31.71	6.959
gamma width (deg)	1	1	yes	10.26	-10.08	26.25
rel_prob	1	1	no	1.0	nan	nan
J mean (MHz)	1	1	no	0.0	nan	nan
J width (MHz)	1	1	no	0.0	nan	nan

If the confidence intervals of the fitting parameters are equal to or exceed the optimization ranges, their errors are set to nan (“not a number” or “not defined”). The error of the fixed parameters are also set to nan.

- *Results overview.* The results of the fitting are saved into a new folder, which is automatically created in examples/example_f1. They consist of several output files (Chapter 5.2), some of which are discussed below.

The fitting is considered successful only if the fitting algorithm converges to the global minimum of χ^2. In practice, it is often difficult to distinguish between the global minimum and local minima. Nevertheless, the global minimum is reached with high probability if the following two criteria are fulfilled: 1) χ^2 reaches a minimum and subsequently does not change significantly in the last few hundred iterations of the fitting algorithm, and 2) the final χ^2 corresponds to a good fit to the PELDOR time traces (as confirmed by eye). To test the first criterion, the evolution of χ^2 over the course of the fitting can be considered. This data is contained in the score.png file. As can be seen in Figure 6.3a, χ^2 drops rapidly during the first 50 iterations of the GA, then gradually decreases until 400 iterations, and finally does not change significantly between 400 and 1000 iterations. The subsequent 200 iterations of the

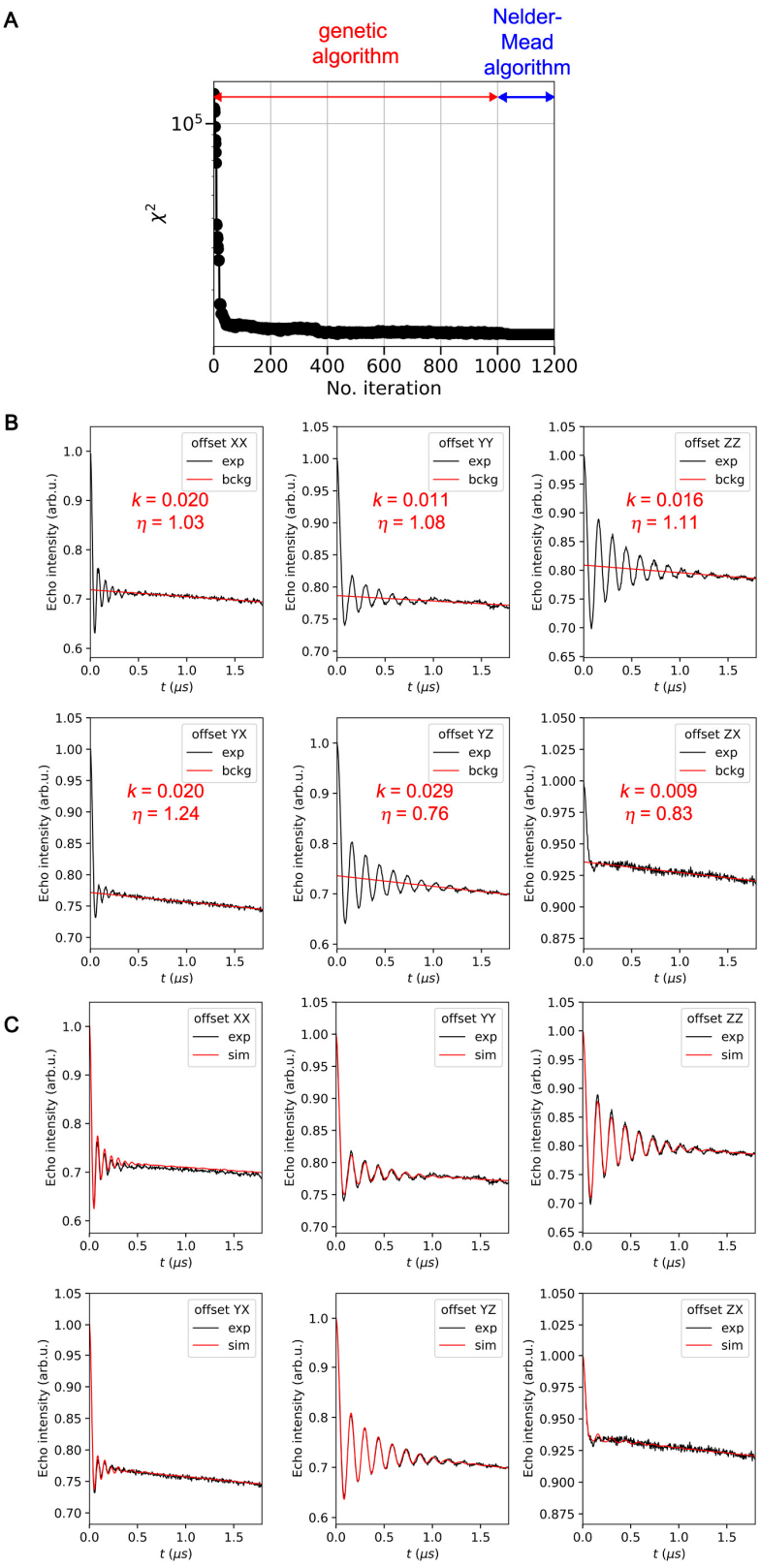

Figure 6.3. Fitting of the PELDOR time traces of **1**. A) χ^2 as a function of optimization step. Optimization steps 1-1000 and 1001-1200 were carried out by the genetic and Nelder-Mead algorithms, respectively. C) The experimental PELDOR time traces (black) of **1** are overlaid with the optimized backgrounds (red). The optimized values of background parameters are shown on top of each time trace. C) The experimental PELDOR time traces (black) of **1** are overlaid with their fits (red).

Table 6.2. Parameters used to fit the PELDOR time traces of **1**.

Distribution	Distribution type	Mean value (notation)	Width[a] (notation)
$P(r)$	Gaussian	$1.948^{+0.004}_{-0.001}$ nm ($<r_{1,1}>$)	$0.027^{+0.002}_{-0.003}$ nm ($\Delta r_{1,1}$)
$P(\xi)$	von Misses	$87°^{+3°}_{-14°}$ ($<\xi_{1,1}>$)	$20°^{+4°}_{-6°}$ ($\Delta\xi_{1,1}$)
$P(\varphi)$	von Misses	$22°^{+3°}_{-7°}$ ($<\varphi_{1,1}>$)	$2°^{+10°}_{-1°}$ ($\Delta\varphi_{1,1}$)
$P(\alpha)$	von Misses	$26°^{+10°}_{-6°}$ ($<\alpha_{1,1}>$)	$9°^{+7°}_{-9°}$ ($\Delta\alpha_{1,1}$)
$P(\beta)$	von Misses	$90°^{+0°}_{-14°}$ ($<\beta_{1,1}>$)	$10°^{+12°}_{-10°}$ ($\Delta\beta_{1,1}$)
$P(\gamma)$	von Misses	$32°^{+7°}_{-32°}$ ($<\gamma_{1,1}>$)	$10°^{+26°}_{-10°}$ ($\Delta\gamma_{1,1}$)

[a] The width is defined as a standard deviation.

Nelder-Mead algorithm also result in a negligible change of χ^2. All above indicates that the fitting has converged to a χ^2 minimum, presumably, the global minimum. The fit to the PELDOR time traces can be found in the fits.png file. As can be seen from Figure 6.3c, the shapes of all experimental PELDOR time traces is well reproduced by the corresponding fit. Thus, the second criterion is fulfilled too and, therefore, the obtained minimum of χ^2 is the global minimum. This shows that the optimized model of the nitroxide-nitroxide spin pair provides the best possible fit to the PELDOR data and that this model, with all its assumptions, is sufficient and adequate for the given molecule.

Next, the parameters of the optimized model are considered. The optimized values of the background (background_parameters.dat) and fitting parameters (fitting_parameters.dat) are listed in Figure 6.3b and Table 6.2, respectively. The simulated backgrounds can be found in the backgrounds.png file and are shown in Figure 6.3b. As can be seen, all backgrounds provide a good fit to the unmodulated part of the PELDOR time traces. To determine the 3σ confidence intervals of the optimized fitting parameters, the two- and one-dimensional error profiles were calculated (see Chapter 1.9 for details). These error profiles can be found in the score_vs_parameters.png and score_vs_parameter.png files, respectively, and are reproduced in Figure 6.4. In this figure, the uncertainty ranges of the fitting parameters are shown in dark red, and the optimized values of the fitting parameters are depicted as white dots. As expected, all the dots appear within the above uncertainty ranges, which means that the optimized values of the fitting parameters are within the corresponding uncertainty ranges. All one-dimensional error profiles have a single, well-defined minimum. Since the shape of this minimum deviates from the normal distribution, the errors of the fitting parameters are determined as asymmetric errors. Each asymmetric error consists of two values: the first is given by the difference between the lower bound of the uncertainty interval and the optimized value of a fitting parameter, and

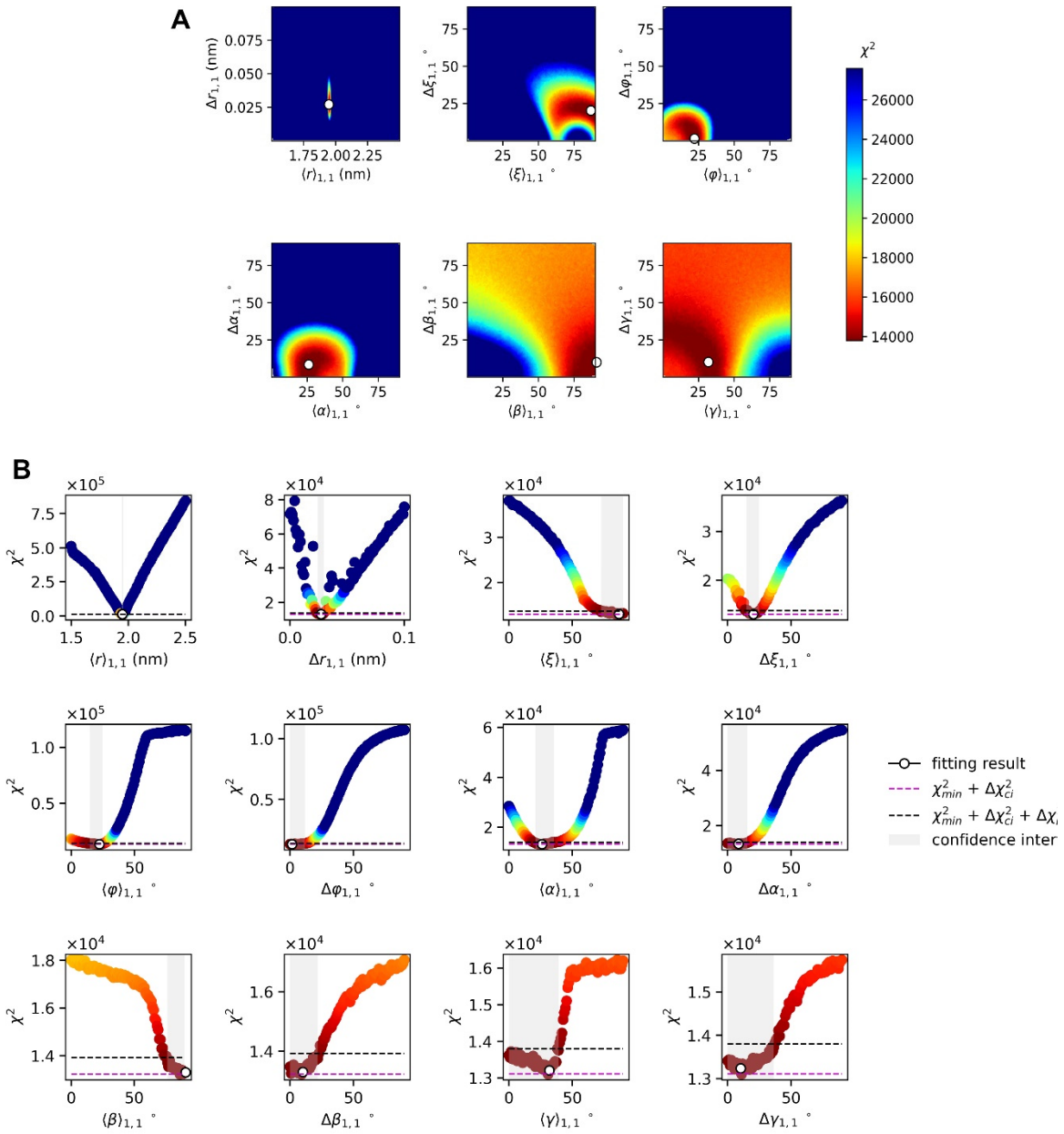

Figure 6.4. Error analysis for the optimized fitting parameters of PeldorFit. A) χ^2 in dependence of the fitting parameters describing the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$. The optimized values of the fitting parameters are depicted by a circle. B) χ^2 in dependence of individual fitting parameters. The χ^2 threshold is depicted by black dashed lines. The uncertainty ranges are shown as gray intervals.

the second by the difference between the upper bound of the uncertainty interval and the optimized value of a fitting parameter. The errors of all fitting parameters are given in Table 6.2. According to this table, the mean value and width of $P(r)$ have an error below 0.01 nm! The mean values and widths of the angular distributions have different errors. The reason for this might be that the different angular parameters have unequal contributions to the orientation selectivity. On average, the parameters of the angular distributions have an error of 10°.

Table 6.3. Symmetry-related sets of angular parameters of **1**.

Transformation[a]	$\langle\xi_{1,1}\rangle$	$\langle\varphi_{1,1}\rangle$	$\langle\alpha_{1,1}\rangle$	$\langle\beta_{1,1}\rangle$	$\langle\gamma_{1,1}\rangle$	χ^2
Fitting result	87°	22°	26°	91°	32°	13478
180° rotation about g_{xx}^B	87°	22°	206°	89°	148°	13418
180° rotation about g_{yy}^B	87°	22°	206°	89°	328°	13437
180° rotation about g_{zz}^B	87°	22°	26°	91°	212°	13517
180° rotation about g_{xx}^A	93°	338°	154°	89°	212°	13442
180° rotation about g_{xx}^A and g_{xx}^B	93°	338°	334°	91°	328°	13145
180° rotation about g_{xx}^A and g_{yy}^B	93°	338°	334°	91°	148°	13285
180° rotation about g_{xx}^A and g_{zz}^B	93°	338°	154°	89°	32°	13340
180° rotation about g_{yy}^A	93°	158°	334°	89°	212°	13134
180° rotation about g_{yy}^A and g_{xx}^B	93°	158°	154°	91°	328°	13310
180° rotation about g_{yy}^A and g_{yy}^B	93°	158°	154°	91°	148°	13355
180° rotation about g_{yy}^A and g_{zz}^B	93°	158°	334°	89°	32°	13561
180° rotation about g_{zz}^A	87°	202°	206°	91°	32°	13367
180° rotation about g_{zz}^A and g_{xx}^B	87°	202°	26°	89°	148°	13264
180° rotation about g_{zz}^A and g_{yy}^B	87°	202°	26°	89°	328°	13328
180° rotation about g_{zz}^A and g_{zz}^B	87°	202°	206°	91°	212°	13408

[a] g_{xx}^A, g_{yy}^A, and g_{zz}^A denote the principal components of the *g*-tensor of spin A; g_{xx}^B, g_{yy}^B, and g_{zz}^B denote the principal components of the *g*-tensor of spin B.

The symmetry-related sets of angular parameters $\langle\xi_{1,1}\rangle$, $\langle\varphi_{1,1}\rangle$, $\langle\alpha_{1,1}\rangle$, $\langle\beta_{1,1}\rangle$, and $\langle\gamma_{1,1}\rangle$ can be found in the symmetry_related_solutions.dat file and are shown in Table 6.3. The first raw of this table contains the values obtained from the fitting (compare to Table 6.2), while the next 15 rows provide their symmetry-based equivalents. As follows from the last column of Table 6.3, all 16 sets yield the same χ^2 within the numerical error ($\Delta\chi_{ne}^2 = 687$).

6.2.2 Orientation-selective PELDOR on a two-spin system with two states

6.2.3 Orientation-selective PELDOR on a three-spin system

6.2.4 Orientation-selective PELDOR on a two-spin system with J

6.2.5 RIDME on a two-spin system with an anisotropic spin center

7 References

[1] A. D. Milov, K. M. Salikhov, M. D. Shchirov, *Fiz. Tverd. Tela* **1981**, *23*, 975.

[2] R. E. Martin, M. Pannier, F. Diederich, V. Gramlich, M. Hubrich, H. W. Spiess, *Angew. Chem. Int. Ed.* **1998**, *37*, 2833.

[3] P. P. Borbat, E. R. Georgieva, J. H. Freed, *J. Phys. Chem. Lett.* **2013**, *4*, 170.

[4] P. E. Spindler, I. Waclawska, B. Endeward, J. Plackmeyer, C. Ziegler, T. F. Prisner, *J. Phys. Chem. Lett.* **2015**, *6*, 4331.

[5] A. Doll, G. Jeschke, *Phys. Chem. Chem. Phys.* **2017**, *19*, 1039.

[6] M. Di Valentin, M. Albertini, M. G. Dal Farra, E. Zurlo, L. Orian, A. Polimeno, M. Gobbo, D. Carbonera, *Chem. Eur. J.* **2016**, *22*, 17204.

[7] L. V. Kulik, S. A. Dzuba, I. A. Grigoryev, Y. Tsvetkov, *Chem. Phys. Lett.* **2001**, *343*, 315.

[8] S. Milikisyants, F. Scarpelli, M. G. Finiguerra, M. Ubbink, M. Huber, *J. Magn. Reson.* **2009**, *201*, 48.

[9] D. Abdullin, M. Suchatzki, O. Schiemann, *Appl. Magn. Reson.* **2022**, *53*, 539.

[10] S. Saxena, J. H. Freed, *Chem. Phys. Lett.* **1996**, *251*, 102.

[11] G. Jeschke, M. Pannier, A. Godt, H. W. Spiess, *Chem. Phys. Lett.* **2000**, *331*, 243.

[12] P. Schöps, P. E. Spindler, A. Marko, T. F. Prisner, *J. Magn. Reson.* **2015**, *250*, 55.

[13] O. Schiemann, T. F. Prisner, *Q. Rev. Biophys.* **2007**, *40*, 1.

[14] G. Jeschke, *eMagRes* **2016**, *5*, 1459.

[15] P. P. Borbat, J. H. Freed, *eMagRes* **2017**, 465.

[16] C. R. Timmel, J. R. Harmer (Eds.) *Structure and Bonding*, *Vol. 152*, Springer Berlin Heidelberg, **2013**.

[17] G. Jeschke, V. Chechik, P. Ionita, A. Godt, H. Zimmermann, J. Banham, C. R. Timmel, D. Hilger, H. Jung, *Appl. Magn. Reson.* **2006**, *30*, 473.

[18] S. G. Worswick, J. A. Spencer, G. Jeschke, I. Kuprov, *Sci. Adv.* **2018**, *4*, eaat5218.

[19] L. Fábregas Ibáñez, G. Jeschke, S. Stoll, *Magn. Reson.* **2020**, *1*, 209.

[20] R. A. Stein, A. H. Beth, E. J. Hustedt, *Methods Enzymol.* **2015**, *563*, 531.

[21] C. J. López, Z. Yang, C. Altenbach, W. L. Hubbell, *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110*, E4306-15.

[22] O. Schiemann, C. A. Heubach, D. Abdullin, K. Ackermann, M. Azarkh, E. G. Bagryanskaya, M. Drescher, B. Endeward, J. H. Freed, L. Galazzo et al., *J. Am. Chem. Soc.* **2021**, *143*, 17875.

[23] O. Schiemann, P. Cekan, D. Margraf, T. F. Prisner, S. T. Sigurdsson, *Angew. Chem. Int. Ed.* **2009**, *48*, 3292.

[24] B. Endeward, J. A. Butterwick, R. MacKinnon, T. F. Prisner, *J. Am. Chem. Soc.* **2009**, *131*, 15246.

[25] C. Abé, D. Klose, F. Dietrich, W. H. Ziegler, Y. Polyhach, G. Jeschke, H.-J. Steinhoff, *J. Magn. Reson.* **2012**, *216*, 53.

[26] I. Tkach, S. Pornsuwan, C. Höbartner, F. Wachowius, S. T. Sigurdsson, T. Y. Baranova, U. Diederichsen, G. Sicoli, M. Bennati, *Phys. Chem. Chem. Phys.* **2013**, *15*, 3433.

[27] M. Kerzhner, D. Abdullin, J. Więcek, H. Matsuoka, G. Hagelueken, O. Schiemann, M. Famulok, *Chem. Eur. J.* **2016**, *22*, 12113.

[28] V. P. Denysenkov, T. F. Prisner, J. Stubbe, M. Bennati, *Proc. Natl. Acad. Sci. U.S.A.* **2006**, *103*, 13386.

[29] V. P. Denysenkov, D. Biglino, W. Lubitz, T. F. Prisner, M. Bennati, *Angew. Chem. Int. Ed.* **2008**, *47*, 1224.

[30] I. M. C. van Amsterdam, M. Ubbink, G. W. Canters, M. Huber, *Angew. Chem. Int. Ed.* **2003**, *42*, 62.

[31] J. E. Lovett, A. M. Bowen, C. R. Timmel, M. W. Jones, J. R. Dilworth, D. Caprotti, S. G. Bell, L. L. Wong, J. Harmer, *Phys. Chem. Chem. Phys.* **2009**, *11*, 6840.

[32] M. M. Roessler, M. S. King, A. J. Robinson, F. A. Armstrong, J. Harmer, J. Hirst, *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107*, 1930.

[33] Z. Yang, M. R. Kurpiewski, M. Ji, J. E. Townsend, P. Mehta, L. Jen-Jacobson, S. Saxena, *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, E993-1000.

[34] D. Abdullin, N. Florin, G. Hagelueken, O. Schiemann, *Angew. Chem. Int. Ed.* **2015**, *54*, 1827.

[35] A. M. Bowen, E. O. D. Johnson, F. Mercuri, N. J. Hoskins, R. Qiao, J. S. O. McCullagh, J. E. Lovett, S. G. Bell, W. Zhou, C. R. Timmel et al., *J. Am. Chem. Soc.* **2018**, *140*, 2514.

[36] D. Margraf, B. E. Bode, A. Marko, O. Schiemann, T. F. Prisner, *Mol. Phys.* **2007**, *105*, 2153.

[37] B. E. Bode, J. Plackmeyer, T. F. Prisner, O. Schiemann, *J. Phys. Chem. A* **2008**, *112*, 5064.

[38] B. E. Bode, J. Plackmeyer, M. Bolte, T. F. Prisner, O. Schiemann, *J. Organomet. Chem.* **2009**, *694*, 1172.

[39] Z. Yang, D. Kise, S. Saxena, *J. Phys. Chem. B* **2010**, *114*, 6165.

[40] D. Abdullin, G. Hagelueken, R. I. Hunter, G. M. Smith, O. Schiemann, *Mol. Phys.* **2015**, *113*, 544.

[41] A. M. Bowen, M. W. Jones, J. E. Lovett, T. G. Gaule, M. J. McPherson, J. R. Dilworth, C. R. Timmel, J. R. Harmer, *Phys. Chem. Chem. Phys.* **2016**, *18*, 5981.

[42] A. Potapov, *J. Magn. Reson.* **2020**, *316*, 106769.

[43] A. V. Astashkin, B. O. Elmore, W. Fan, J. G. Guillemette, C. Feng, *J. Am. Chem. Soc.* **2010**, *132*, 12059.

[44] D. Abdullin, H. Matsuoka, M. Yulikov, N. Fleck, C. Klein, S. Spicher, G. Hagelueken, S. Grimme, A. Lützen, O. Schiemann, *Chem. Eur. J.* **2019**, *25*, 8820.

[45] D. Abdullin, P. Brehm, N. Fleck, S. Spicher, S. Grimme, O. Schiemann, *Chem. Eur. J.* **2019**, *25*, 14388.

[46] D. Abdullin, *Appl. Magn. Reson.* **2020**, *51*, 725.

[47] A. Weber, O. Schiemann, B. Bode, T. F. Prisner, *J. Magn. Reson.* **2002**, *157*, 277.

[48] G. W. Reginsson, R. I. Hunter, P. A. S. Cruickshank, D. R. Bolton, S. T. Sigurdsson, G. M. Smith, O. Schiemann, *J. Magn. Reson.* **2012**, *216*, 175.

[49] N. C. Kunjir, G. W. Reginsson, O. Schiemann, S. T. Sigurdsson, *Phys. Chem. Chem. Phys.* **2013**, *15*, 19673.

[50] A. Meyer, J. J. Jassoy, S. Spicher, A. Berndhäuser, O. Schiemann, *Phys. Chem. Chem. Phys.* **2018**, *20*, 13858.

[51] G. Jeschke, *Macromol. Rapid Commun.* **2002**, *23*, 227.

[52] B. E. Bode, D. Margraf, J. Plackmeyer, G. Dürner, T. F. Prisner, O. Schiemann, *J. Am. Chem. Soc.* **2007**, *129*, 6736.

[53] G. Hagelueken, W. J. Ingledew, H. Huang, B. Petrovic-Stojanovska, C. Whitfield, H. ElMkami, O. Schiemann, J. H. Naismith, *Angew. Chem. Int. Ed.* **2009**, *48*, 2904.

[54] G. Jeschke, M. Sajid, M. Schulte, A. Godt, *Phys. Chem. Chem. Phys.* **2009**, *11*, 6580.

[55] C. Pliotas, R. Ward, E. Branigan, A. Rasmussen, G. Hagelueken, H. Huang, S. S. Black, I. R. Booth, O. Schiemann, J. H. Naismith, *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, E2675-82.

[56] T. von Hagens, Y. Polyhach, M. Sajid, A. Godt, G. Jeschke, *Phys. Chem. Chem. Phys.* **2013**, *15*, 5854.

[57] R. Ward, C. Pliotas, E. Branigan, C. Hacker, A. Rasmussen, G. Hagelueken, I. R. Booth, S. Miller, J. Lucocq, J. H. Naismith et al., *Biophys. J.* **2014**, *106*, 834.

[58] J. J. Jassoy, A. Meyer, S. Spicher, C. Wuebben, O. Schiemann, *Molecules* **2018**, *23*, 682.

[59] D. Abdullin, G. Hagelueken, O. Schiemann, *Phys. Chem. Chem. Phys.* **2016**, *18*, 10428.

[60] E. J. Hustedt, F. Marinelli, R. A. Stein, J. D. Faraldo-Gómez, H. S. Mchaourab, *Biophys. J.* **2018**, *115*, 1200.

[61] A. D. Milov, A. G. Maryasov, Y. D. Tsvetkov, *Appl. Magn. Reson.* **1998**, *15*, 107.

[62] A. D. Milov, A. B. Ponomarev, Y. Tsvetkov, *Chem. Phys. Lett.* **1984**, *110*, 67.

[63] A. D. Milov, Y. D. Tsvetkov, *Appl. Magn. Reson.* **1997**, *12*, 495.

[64] K. Keller, M. Qi, C. Gmeiner, I. Ritsch, A. Godt, G. Jeschke, A. Savitsky, M. Yulikov, *Phys. Chem. Chem. Phys.* **2019**, *21*, 8228.

[65] D. Abdullin, F. Duthie, A. Meyer, E. S. Müller, G. Hagelueken, O. Schiemann, *J. Phys. Chem. B* **2015**, *119*, 13534.

[66] A. F. Bedilo, A. G. Maryasov, *J. Magn. Reson.* **1995**, *116*, 87.

[67] D. Abdullin, O. Schiemann, *ChemPlusChem* **2020**, *85*, 353.

[68] A. Marko, D. Margraf, H. Yu, Y. Mu, G. Stock, T. Prisner, *J. Chem. Phys.* **2009**, *130*, 64102.

[69] A. Doll, G. Jeschke, *J. Magn. Reson.* **2017**, *280*, 46.

[70] A. V. Astashkin, *Methods Enzymol.* **2015**, *563*, 251.

[71] S. Katoch, S. S. Chauhan, V. Kumar, *Multimed. Tools Appl.* **2021**, *80*, 8091.

[72] A. Vie, A. M. Kleinnijenhuis, D. J. Farmer, "Qualities, challenges and future of genetic algorithms: a literature review", can be found under https://arxiv.org/pdf/2011.05277, **2020**.

[73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes in C. The art of scientific computing*, Cambridge University Press, Cambridge, **1992**.

[74] T. H. Edwards, S. Stoll, *J. Magn. Reson.* **2016**, *270*, 87.