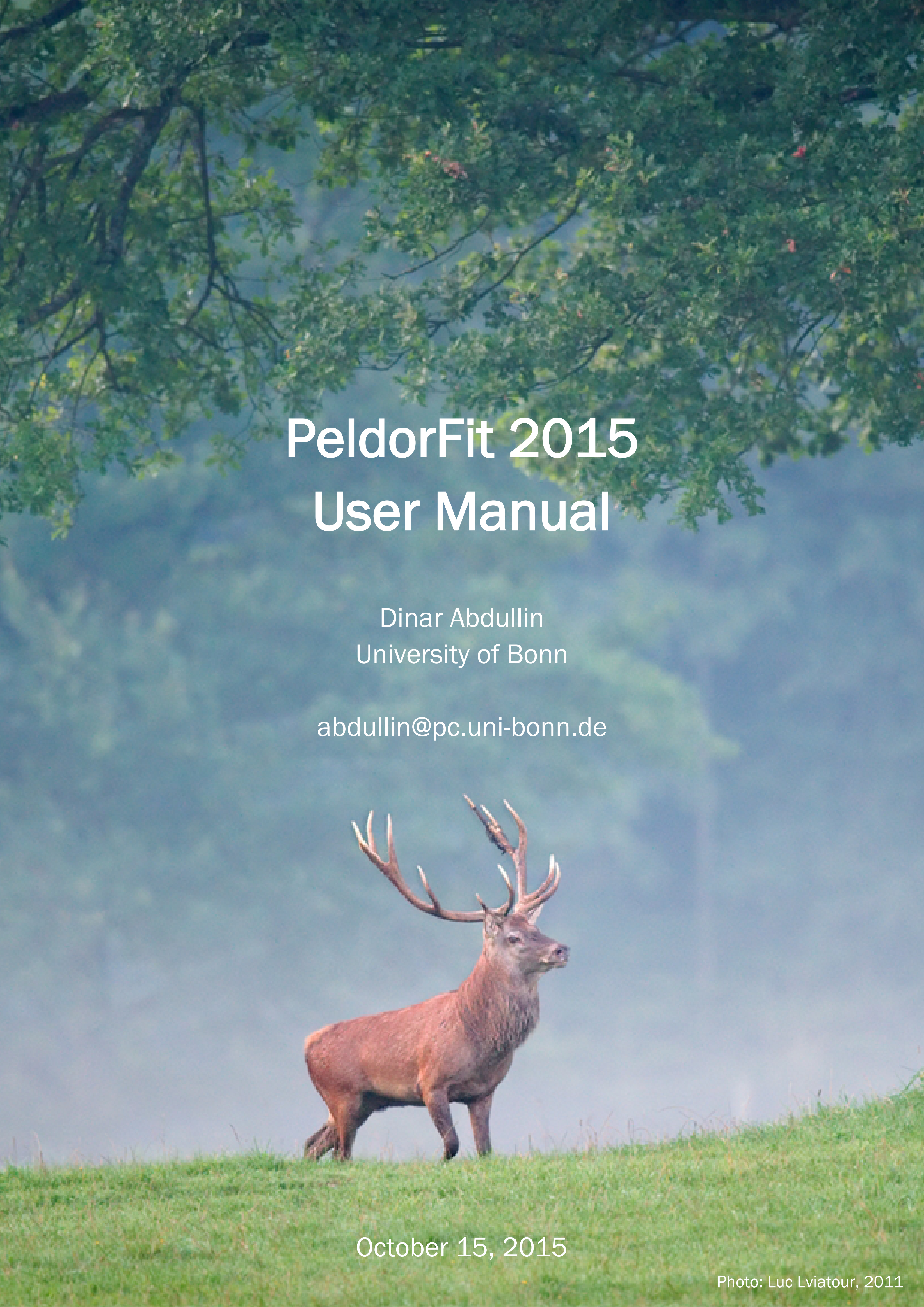

PeldorFit 2015 User Manual

Dinar Abdullin
University of Bonn

abdullin@pc.uni-bonn.de

October 15, 2015

Photo: Luc Lviatour, 2011

Table of contents

1 Introduction

The program PeldorFit performs an analysis of the orientation-selective Pulsed ELectron-electron Double Resonance (PELDOR or DEER) signals. The orientation-selective PELDOR signals contain information about the distance between two spin centers in an ensemble of identical molecules and about the relative orientations of these centers. To extract this information, the program uses a simplified model of a spin pair (Figure 1) for which several assumptions are taken:

1) the spins are considered as single-point objects;

2) geometric parameters of the model have either a normal or uniform distribution;

3) a correlation between individual geometric parameters of the model is neglected.

The geometry of the model is optimized via genetic algorithm provide the best match between experimental PELDOR signals and the corresponding PELDOR signals simulated for the model. Further description of the model and its optimization can be found in:

D. Abdullin, G. Hagelueken, R. I. Hunter, G. M. Smith, O. Schiemann, *Geometric model-based fitting algorithm for orientation-selective PELDOR data*, Mol. Phys. 2015, 113, 544-560.

PeldorFit is a console application. The source code of the program is written using C++. It was compiled and tested in the Linux and Windows operating systems. Both, the source code and the executables, can be found at https://github.com/dinarabdullin/PeldorFit2015. The program is free and can be distributed under GNU General Public License.

This manual is organized as follows. Installation of the program is described in Chapter 2. Chapter 3 focuses on running the program on PC. Chapter 4 explains how to create and edit configuration files of the program. The description of the output data of the program is given in Chapter 5. Finally, an example of using the program is given in Chapter 6.

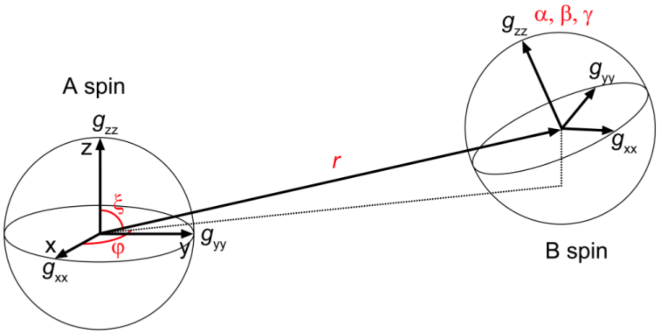

Figure 1. Geometric model of the spin pair used in the program PeldorFit.

2 Installation

Download the zip-archive “PeldorFit2015.zip” from the web-page
https://github.com/dinarabdullin/PeldorFit2015
and unzip it to a directory where the program will be stored. The archive contains two folders with the compiled executable files for the Linux and Windows operating systems. (If you want to compile the program yourself, the source code of the program can be found in https://github.com/dinarabdullin/PeldorFit2015)

3 Running the program

1. Open Terminal (Linux) or Command Prompt (Windows).
2. Change the directory to the one where the PeldorFit executable file is stored:
   cd .../PeldorFit2015/Linux (for Linux)
   cd .../PeldorFitV2015/Windows (for Windows).
3. Prepare the configuration file (see Chapter 4) and run the program by the following command:
   sh PeldorFit.sh .../config.cfg (for Linux)
   PeldorFit.exe .../config.cfg (for Windows)
   An example of the configuration file can be found in the *Examples* folder of the program.

4 Configuration file

The configuration file is an ASCII file which contains all input parameters of the program PeldorFit. The configuration file is read by the program at the beginning of execution. The similar approach is used by the PulseSPEL programs (Bruker), where all parameters of a pulsed ESR experiment are stored in the configuration file. Note that the PeldorFit configuration files have a *.cfg* extension. They can be created and edited with common text editors such as Notepad++. The syntax of the configuration file is relatively simple and intuitive. Each input parameter is specified in this file by a certain name and the corresponding value. The designations and meanings are described in details below. As a remark, the number of the parameters in the configuration file is quite large. Therefore it is suggested to use the configuration file from the installation folder *Examples* (*bisnitroxide_config.cfg*) as a template to build your own configuration file. This will save your time and reduce the probability of errors!

The PeldorFit configuration file contains five main blocks of information:

1) parameters of PELDOR experiments;

2) spectroscopic characteristics of spins involved PELDOR experiments;

3) fitting parameters;

4) settings of an optimization (genetic) algorithm;

5) the content of the program's output.

The first part of the configuration file specifies the file names of the PELDOR data-set and the corresponding experimental settings. All these data is joined together in a list called *experimentals:*

Example

```
experimentals = (
  {filename = "Examples/bisnitroxide_XX.dat";    // name of the data file
  detPiLength = 14;                              // length of the detection pi-pulse (ns)
  detPiHalfLength = 7;                           // length of the detection pi/2-pulse (ns)
  pumpPiLength = 14;                             // length of the pump pi-pulse (ns)
  detFreq =   93.999600;                         // detection frequency (GHz)
  pumpFreq =  93.930000;                         // pump frequency (GHz)
  magnField =  3.3415; },                        // magnetic field (T)
  {filename = "Examples/bisnitroxide_YY.dat";
  detPiLength =   14;
  detPiHalfLength = 7;
  pumpPiLength =   14;
  detFreq =  93.999600;
  pumpFreq = 93.930000;
  magnField =  3.3443; }
);
```

In this list, information about each separate PELDOR experiment is enclosed in braces. Within the braces one has to define the following parameters:

filename A path to a data-file which contains the background-subtracted PELDOR time trace. The first column of this file should contain the values of the time axis in microseconds. The second column should contain the values of the background-subtracted PELDOR signal normalized to 1. Such file can be created by the Bruker Xepr software or by the program DeerAnalysis (the file with the suffix “fit”).

detPiLength The length of the detection π pulse in [ns]

detPiHalfLength The length of the detection $\pi/2$ pulse in [ns]

pumpPiLength The length of the pump pulse in [ns]

detFreq The frequency of the detection pulses in [GHz]

pumpFreq The frequency of the pump pulse in [GHz]

magnField The value of the magnetic field in [T]

Comments are indicated in the configuration file by # or // symbols. Note also that the commas and semicolons appearing in the file are important for the interpretation of the configuration file at execution time.

The second part of the configuration file describes the spectroscopic parameters of two spin centers. For definiteness, the spins are denoted as spin A and spin B:

Example

```
spinA:
{
  g = [2.0104, 2.0073, 2.0033];           // g-factor
  gStrain = [0.0004, 0.0003, 0.0001];     // g-strain
  n = [1];                                // number of nuclei
  I = [1.0];                              // nuclear spin
  A = [8.0, 6.0, 96.0];                   // hyperfine coupling constant A (MHz)
  AStrain = [0.0, 0.0, 12.0];             // A-strain (MHz)
  lwpp = 22.4;                            // peak-to-peak linewidth (MHz)
};

spinB:
{
  g = [2.0104, 2.0073, 2.0033];
  gStrain = [0.0004, 0.0003, 0.0001];
  n = [1];
  I = [1.0];
  A = [8.0, 6.0, 96.0];
  AStrain = [0.0, 0.0, 12.0];
  lwpp = 22.4;
};
```

It is assumed that resonance frequencies of both spins are determined by the Zeeman interaction, the hyperfine interaction of electron spins and nuclear spins, and an inhomogeneous line broadening. The unresolved anisotropy of *g*- and *A*- tensors can also be taken into account via *g*-strain and *A*-strain. The naming of spectroscopic parameters was done in consistency with the program EasySpin:

$g = [g_{xx}, g_{yy}, g_{zz}]$ *g*-tensor: It should consist of three components even in the case of isotropic or axial g-tensors.

$gStrain = [\Delta g_{xx}, \Delta g_{yy}, \Delta g_{zz}]$ *g*-strain (the unresolved anisotropy of the *g*-tensor): If not needed, the notation *gStrain = []* should be used.

n = [n1, n2] The number of equivalent nuclei: The program allows using two sorts of equivalent nuclei. If an electron spin is coupled to only one nucleus, the notation *n = [1]* should be used. In the absence of the hyperfine interaction, the notation *n = []* should be used.

I = [I1, I2] The nuclear spin(s) of a nucleus (nuclei) coupled to an electron spin: In the case of one sort of equivalent nuclei, the notation *I = [I1]* should be used. In the absence of the hyperfine interaction, the notation *I = []* should be used.

$A = [A1_{xx}, A1_{yy}, A1_{zz}, A2_{xx}, A2_{yy}, A2_{zz}]$ The hyperfine coupling tensor in MHz: It should consist of 0, 3 or 6 components in case of 0, 1 or 2 two sorts of equivalent nuclei, correspondingly.

$AStrain = [\Delta A1_{xx}, \Delta A1_{yy}, \Delta A1_{zz}]$ A-strain in MHz (the unresolved anisotropy of the *A*-tensor): It applies only to the first specified sort of the nuclei. If not needed, the notation *AStrain = []* should be used.

lwpp The peak-to-peak linewidth in MHz.

The third part of the configuration file specifies fitting parameters. To provide a fit to the experimental PELDOR time traces, the program optimizes a geometric model of the spin pair (Figure 1). The model consists of two coordinate frames associated with the *g*-tensor principle axes of the two spins, spin A and spin B. Both spins are considered as point objects, i.e. spin delocalization is neglected. The reference coordinate system of the model is set to be coincident with the *g*-tensor principle axes of the spin A. Then, the orientation of the *g*-tensor principle axes corresponding to the spin B is represented by three Euler angles (α, β, γ). The Euler angles are defined in accordance with the z-x'-z'' convention. The vector connecting the origins of these two frames determines the inter-spin vector $\boldsymbol{r}$. In the reference coordinate system, $\boldsymbol{r}$ is described by spherical coordinates: length of the vector r, polar ξ and azimuthal φ

angles. Due to the flexibility of a molecule carrying the spins, the described geometric parameters may have some distributions. The program allows user to approximate their distributions by either a uniform distribution or a normal distribution. Thus, the model is described by the mean values (μ, α, β, γ, ξ, φ) and the distribution widths/standard deviations (σ, $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, $\Delta\xi$, $\Delta\varphi$) of six geometric parameters.

Depending on the symmetry of the *g*- and *A*-tensors of spins A and B, all 12 geometric parameters as defined above or only a subset of them are needed to simulate the PELDOR time traces. If the magnetic tensors of both spins are rhombic, the full set of geometric parameters is required for the calculations. In contrast, 8 parameters are sufficient when the magnetic tensors of both spins are axial. Moreover, due to the inversion symmetry of the magnetic tensors (spin Hamiltonian) there are a number of relative orientations of the spin centers, which provide identical PELDOR time traces. If each spin is considered as a point object, the 180° rotation of one of the *g*-tensor principle axes doesn't influence the shape of the PELDOR time trace. All these considerations are summarized in Table 1, which specifies the number of required angular parameters and their variation ranges for all possible combinations of isotropic, axial, and rhombic spin systems A and B.

The model described above has a unimodal distribution of spin-spin distances. However, this is not always the case, for example, when two or more distinct conformations of a molecule exist. To allow for a bimodal distribution of spin-spin distances, the number of geometric parameters of the model can be extended up to 24 by adding the second set of the r, α, β, γ, ξ, and φ parameters and their distributions. The ratio between both conformations can be also optimized as another fitting parameter.

Additionally to the geometric parameters, the fitting parameters may include an exchange coupling constant J, its distribution width ΔJ, and inversion efficiency of the pump pulse η ($0 < \eta \leq 1$). The latter parameter is sometimes very useful for the fitting since it allows taking into account imperfections of the pump pulse, incomplete spin labelling, etc.

Table 1. Number and ranges of Euler angles (α, β, γ) and polar angles (ξ, φ) required to simulate the PELDOR time traces in the cases of isotropic, axial, and rhombic magnetic tensors of spins A and B.

Spin A	Spin B	α	β	γ	ξ	φ
axial	isotropic	-	-	-	[0°, 90°]	-
axial	axial	[0°, 180°]	[0°, 90°]	-	[0°, 90°]	-
rhombic	isotropic	-	-	-	[0°, 90°]	[0°, 180°]
rhombic	axial	[0°, 180°]	[0°, 90°]	-	[0°, 90°]	[0°, 180°]
rhombic	rhombic	[0°, 180°]	[0°, 90°]	[0°, 180°]	[0°, 90°]	[0°, 180°]

The configuration file stores all possible fitting parameters in the list called *parameters*. Note that the order, with which the fitting parameters appear in this list, is pre-defined and should not be changed (see Example and Appendix A).

Example

```
parameters = (
   {opt = 1;   mode = 1;   range = [1.50, 2.50]},          // Parameter 1:  distance mean 1
   {opt = 1;   mode = 1;   range = [0.00, 0.10]},          // Parameter 2:  distance width 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },          // Parameter 3:  xi mean 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },          // Parameter 4:  xi width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 5:  phi mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 6:  phi width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 7:  alpha mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 8:  alpha width 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },          // Parameter 9:  betta mean 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },          // Parameter 10: betta width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 11: gamma mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},          // Parameter 12: gamma width 1
   {opt = 0;   mode = 1;   range = [1.50, 2.50]},          // Parameter 13: distance mean
   {opt = 0;   mode = 1;   range = [0.00, 0.10]},          // Parameter 14: distance width 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },          // Parameter 15: xi mean 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },          // Parameter 16: xi width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 17: phi mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 18: phi width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 19: alpha mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 20: alpha width 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },          // Parameter 21: betta mean 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },          // Parameter 22: betta width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 23: gamma mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},          // Parameter 24: gamma width 2
   {opt = 0;   mode = 0;   range = [0.00, 1.00]},          // Parameter 25: ratio between 1 and 2
   {opt = 0;   mode = 0;   range = [0.00, 5.00]},          // Parameter 26: j coupling mean
   {opt = 0;   mode = 0;   range = [0.00, 5.00]},          // Parameter 27: j coupling width
   {opt = 1;   mode = 0;   range = [0.85, 0.95]}           // Parameter 28: pump pulse efficiency
);
```

User may choose which parameters will be optimized by PeldorFit via setting the values of the setting *opt* to 1. Otherwise the value of *opt* should be set to 0. When *opt* is set to 1, another two settings called *mode* and *range* have to be specified. The *mode* setting determines which type of distribution will be used for r, α, β, γ, ξ, φ, and J: when set to 0 a uniform distribution will be used, and when set to 1 a normal distribution will be used. Note that the *mode* setting has another meaning for the parameter η (parameter 28): when set to 0 a single value of η will be optimized for all PELDOR time traces, and when set to 1 the value of η will be optimized for each PELDOR time trace separately. The *range* setting defines the lower and the upper bounds for each fitting parameter. The bounds of the angular parameters are defined in the Table 1 and usually do not need to be changed. In contrast, the bounds of the distance parameters (μ and σ), the coupling constant (J and ΔJ) and the inversion efficiency of the pump pulse (η) have to be carefully adjusted for each PELDOR data-set.

Optimization of the chosen parameters is performed by a genetic algorithm. This algorithm has been shown to be very efficient when one deals with a large number of optimization parameters and needs a global optimization (finding the global minimum even when the problem has several local minima). Importantly, the genetic algorithm has its own intrinsic parameters which determine its ability to find the global minimum of the optimization problem. The optimal values of these parameters may vary depending on a particular PELDOR data-set. Therefore the configuration file allows user to set these values manually. Note that the intrinsic parameters given in the *Example* were obtained after extensive tests of the genetic algorithm on several PELDOR data-sets and therefore recommended for the use.

Example

```
genetic:
{
   num_generations_max = 500;         // Maximal number of generations
   size_generation = 192;             // Generation size (number of chromosomes)
   prob_crossover = 0.5;              // Crossover rate
   prob_mutation = 0.01;              // Mutation rate
   merit_function = 1;                // Merit function: 1 - RMSD, 2 - RMSD/Pearson, 3 - Pearson
   num_averages = 100000;             // Number of Monte-Carlo averages
};
```

where

num_generations_max The total number of optimization steps (generations). This parameter determines the convergence of the algorithm to the global minimum and the duration of optimization.

size_generation The number of sets of parameters which are treated by the genetic algorithm simultaneously (number of chromosomes). A good practice is to set this parameter to 10 times number of the fitting parameters.

prob_crossover The crossover rate.

prob_mutation The mutation rate.

merit_function The merit function used to calculate a goodness-of-fit.

merit_function = 1 : root-mean square deviation (RMSD)

merit_function = 3 : Pearson product moment correlation coefficient (PCC)

merit_function = 2 : RMSD/PCC

num_averages The number of Monte-Carlo averages used to simulate PELDOR signals. It determines how accurate the powder averaging will be done. (This value should not be lower than 10^5.) This parameter also influences the duration of optimization.

The last part of the configuration file determines which information will be recorded after PeldorFit has finished the optimization.

Example

```
output:
{
    directory = "Examples/Results/";          // Path to the disk space where the fitting results will be saved
    record_spectrum = 1;                      // Record the EPR spectrum of the spin system
    record_score = 1;                         // Record the goodness-of-fit as a function of the optimization step
    record_parameters = 1;                    // Record the best parameters of the model
    record_fit = 1;                           // Record the fit to the PELDOR signals
    record_form_factor = 1;                   // Record the form-factor for the PELDOR signals
    record_symmetric_solutions = 1;           // Record symmetry-related sets of the model's parameters
    record_error_plot = 1;                    // Record the RMSD as a function of the 'error_plot_variables'
    error_plot_variables = ([1,2]);           // Variables of the RMSD plot
    error_plot_size = 1000;                   // Number of points in the RMSD plot
};
```

where

directory — A path to the disk space where the output data will be stored.

record_spectrum — If set to 1, the ESR spectrum of the spin pair will be simulated. While recording the spectrum, the value of the magnetic field is set to the value of *magnField* of the first experiment in the *experimentals* list.

record_score — If set to 1, the goodness-of-fit will be recorded as a function of the optimization step.

record_parameters — If set to 1, the best values of the fitting parameters will be recorded.

record_fit — If set to 1, the fit to the PELDOR time traces will be recorded.

record_form_factor — If set to 1, the angular dependence of the form-factors will be recorded for the PELDOR signals.

record_symmetric_solutions — If set to 1, 16 symmetry-related sets of the fitting parameters will be recorded. The goodness-of-fit will be also recorded for each set.

record_error_plot — If set to 1, the goodness-of-fit will be recorded as a function of chosen *error_plot_variables*.

error_plot_variables — Variables of the RMSD plot: Each fitting parameter is associated with the index, with which this parameter appears in the *parameters* list (see Appendix A). For example, the index of 1 corresponds to μ_1, and the index of 2 corresponds to σ_1. The variables of the error plot are specified by such indeces. These numbers should be written in square brackets for each separate error plot. For example, the notation `error_plot_variables = ([1], [3], [5])` means that three separate error plots will be successively recorded for the parameters μ_1, ξ_1, and φ_1. Each error plot can have more than one dimension. For example, the notation error_plot_variables = ([1, 2], [3, 4]) means that two error plots will be recorded for the pairs of parameters, μ_1 and σ_1, ξ_1 and $\Delta\xi_1$. While recording the error plot, the chosen fitting parameters are varied within the ranges specified for these parameters in the *parameters* list.

error_plot_size   The number of points in the RMSD plot.

To switch off the recording of particular output data, the corresponding parameter should be set to 0 instead of 1.

Additionally, the program PeldorFit has an option called “error plot only”. This option can be useful in the cases, when the fitting of the PELDOR time traces has been already performed, but user wants to record an additional error plot(s) for some of the fitting parameters. Using this opion, user does not need to run the whole optimization again (which is time consuming!); he only needs to activate the *error_plot_only* mode in the configuration file:

Example

```
error_plot_only:
{
   enable = 0;                                                  // Switch to record only the error plot for the
                                                                // previously optimized fitting parameters
   error_plot_variables = ([1,2]);                              // Variables of the RMSD plot
   error_plot_size = 1000;                                      // Number of points in the RMSD plot
   input_directory = “Examples/Results/parameters.dat";         // Path to the file with the optimized fitting
                                                                // parameters
   output_directory = "Examples/Results/";                      // Path to the disk space where the error plot will
                                                                // be saved
};
```

where

enable   Enable the “error plot only” mode .

error_plot_variables   Same setting as *error_plot_variables* in the *output* block.

error_plot_size   Same setting as *error_plot_size* in the *output* block.

input_directory   A path to the disk space where the file with the optimized values of fitting parameters (*parameters.dat*) is stored.

output_directory   Path to the disk space where the output data will be stored.

5 Output data

Depending on the output settings of the configuration file the following output files are generated by the program PeldorFit:

1. *parameters.dat*

   The first column of the file contains the names of the fitting parameters, and the second column contains the optimized values of these parameters.

2. *fit.dat*

   For each PELDOR time trace specified in the configuration file the program writes 3 columns into the "fit" file. The first column corresponds to the time values in microseconds. Other two columns contain the values of the experimental signal and its fit, respectively.

3. *score.dat*

   The first column of the file consists of the numbers of optimization steps, and the second column is filled in with the corresponding values of a goodness-of-fit (e.g. RMSD values).

4. *spectrum.dat*

   The first column of the file contains the x-axis of the ESR spectrum given in gigahertz. The second column contains the values of the spectrum normalized to 1. The spectrum is calculated for the constant value of the magnetic field which is set to the value of *magnField* of the first experiment in the *experimentals* list of the configuration file.

5. *symmetric_parameters.dat*

   The file has a similar structure to the "parameters" file, but instead of only one set of the optimized fitting parameters, it contains 16 symmetry-related sets. Moreover, a goodness-of-fit value is outputted for each set of parameters in the last row of this file.

6. *form_factor.dat*

   The first column of the file contains the values of the dipolar angle ranging from 0° to 90°. In the next columns, the form-factors calculated for the PELDOR time traces are outputted in the same order as the time traces appear in the *experimentals* list of the configuration file.

7. *errorplot_X.dat or errorplot_X_Y.dat where X, Y = 1 - 28*

   Depending on the dimension *D* of the error plot, the first *D* columns of the file contain the values of the fitting parameters chosen for the error plot. The last column of the file contains the corresponding values of a goodness-of-fit.

6 Example

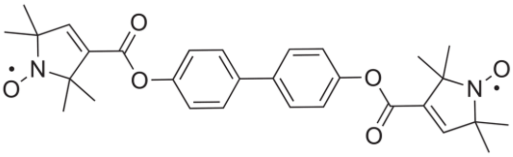

Figure 2. Structure of the bis-nitroxide system.

This chapter provides an example of using the program PeldorFit for the analysis of the PELDOR data-set acquired on the bis-nitroxide model system (Figure 2). The data-set consists of 6 PELDOR time traces which were measured using various values of the pump/detection frequencies and the magnetic field. Before using the time traces in PeldorFit, the background was subtracted from each time trace by means of the program DeerAnalysis.

As a prerequisite to the PeldorFit analysis, a configuration file has to be created. The configuration file for the bis-nitroxide is called *bisnitroxide_config.cfg* and located in the folder *Examples* of the program's directory. The content of this configuration file is reproduced in Appendix B and described in details in Chapter 4. As soon as the configuration file is prepared, PeldorFit can be started by the command:

```
sh PeldorFit.sh /Examples/bisnitroxide_config.cfg (for Linux)
PeldorFit.exe /Examples/bisnitroxide_config.cfg (for Windows)
```

While PeldorFit is running, it displays the current number of optimization step into the Terminal (Linux) or Command Prompt (Windows):

```
Loading input data from the config file...
Number of PELDOR signals is 4
Input data is loaded!
Optimization of the spin geometry via genetic algorithm... Please be patient!
Recording the spectrum... Done!
Optimization step 1/1000
Optimization step 2/1000
…
Optimization step 1000/1000
```

Optimization runs until the number of the optimization steps reaches the value of num_generations_max defined in the configuration file. After this, the output data that is specified in the configuration file is recorded into the folder /*Examples*/*Results*:

```
Recording the fitness vs the number of optimization steps... Done!
Recording the fit to the PELDOR signals... Done!
Recording the best values of fitting parameters... Done!
Recording the form-factors of PELDOR signals... Done!
Recording the symmetry-related sets of fitting parameters... Done!
Recording the error plot... Done!
Finished! The optimization took 6 hours.
```

This data can be then plotted and analysed by means of programs like Origin or Matlab. Such analysis has been done for the present bis-nitroxide molecule in

D. Abdullin, G. Hagelueken, R. I. Hunter, G. M. Smith, O. Schiemann, *Geometric model-based fitting algorithm for orientation-selective PELDOR data*, Mol. Phys. 2015, 113, 544-560.

To make the error plots, one can use the Matlab function *PeldorFitErrorPlot.m*, which is located in the main directory of the program. It can be also downloaded from the web-page: https://github.com/dinarabdullin/PeldorFit2015

Appendix A. Fitting parameters

No.	Parameter	Unit	Range	Mode
1	μ_1	nm	[0, ∞]	
2	σ_1	nm	[0, ∞]	
3	ξ_1	degrees	[0, 90]	
4	$\Delta\xi_1$	degrees	[0, 90]	
5	φ_1	degrees	[0, 180]	
6	$\Delta\varphi_1$	degrees	[0, 180]	
7	α_1	degrees	[0, 180]	
8	$\Delta\alpha_1$	degrees	[0, 180]	
9	β_1	degrees	[0, 90]	
10	$\Delta\beta_1$	degrees	[0, 90]	
11	γ_1	degrees	[0, 180]	
12	$\Delta\gamma_1$	degrees	[0, 180]	
13	μ_2	nm	[0, ∞]	
14	σ_2	nm	[0, ∞]	Mode 0: Uniform distribution
15	ξ_2	degrees	[0, 90]	Mode 1: Normal distribution
16	$\Delta\xi_2$	degrees	[0, 90]	
17	φ_2	degrees	[0, 180]	
18	$\Delta\varphi_2$	degrees	[0, 180]	
19	α_2	degrees	[0, 180]	
20	$\Delta\alpha_2$	degrees	[0, 180]	
21	β_2	degrees	[0, 90]	
22	$\Delta\beta_2$	degrees	[0, 90]	
23	γ_2	degrees	[0, 180]	
24	$\Delta\gamma_2$	degrees	[0, 180]	
25	ratio between 1st and 2nd conformations	-	[0, 1]	
26	J	MHz	[-∞, ∞]	
27	ΔJ	MHz	[-∞, ∞]	
28	η	-	[0, 1]	Mode 0: η is common for all PELDOR signals Mode 1: η is individual for each PELDOR signal

Appendix B. Example of the configuration file

```
#------------------------------------------------ PeldorFit   Configuration File ---------------------------------------------------

# Experimental parameters
#------------------------------------------------
experimentals = (
  {filename = "Examples/bisnitroxide_XX.dat";      // name of the data file
  detPiLength = 14;                                // length of the detection pi-pulse (ns)
  detPiHalfLength = 7;                             // length of the detection pi/2-pulse (ns)
  pumpPiLength = 14;                               // length of the pump pi-pulse (ns)
  detFreq = 93.999600;                             // detection frequency (GHz)
  pumpFreq = 93.930000;                            // pump frequency (GHz)
  magnField = 3.3415; },                           // magnetic field (T)
  {filename = "Examples/bisnitroxide_YY.dat";
  detPiLength = 14;
  detPiHalfLength = 7;
  pumpPiLength = 14;
  detFreq = 93.999600;
  pumpFreq = 93.930000;
  magnField = 3.3443; },
  {filename = "Examples/bisnitroxide_ZZ.dat";
  detPiLength = 14;
  detPiHalfLength = 7;
  pumpPiLength = 16;
  detFreq = 93.999600;
  pumpFreq = 94.100400;
  magnField = 3.3515; },
  {filename = "Examples/bisnitroxide_YX.dat";
  detPiLength = 14;
  detPiHalfLength = 7;
  pumpPiLength = 18;
  detFreq = 93.999600;
  pumpFreq = 93.888000;
  magnField = 3.3415; },
  {filename = "Examples/bisnitroxide_YZ.dat";
  detPiLength = 14;
  detPiHalfLength = 7;
  pumpPiLength = 14;
  detFreq = 93.999600;
  pumpFreq = 94.100400;
  magnField = 3.3490; },
  {filename = "Examples/bisnitroxide_ZX.dat";
  detPiLength = 14;
  detPiHalfLength = 7;
  pumpPiLength = 23;
  detFreq = 93.999600;
  pumpFreq = 93.789600;
  magnField = 3.3415; }
);
```

```
# Spectroscopic parameters
#------------------------------------------------
spinA:                                      // nitroxide spin 1
{
  g = [2.0104, 2.0073, 2.0033];             // g-factor
  gStrain = [0.0004, 0.0003, 0.0001];       // g-strain
  n = [1];                                  // number of nuclei
  I = [1.0];                                // nuclear spin
  A = [8.0, 6.0, 96.0];                     // hyperfine coupling constant A (MHz)
  AStrain = [0.0, 0.0, 12.0];               // A-strain (MHz)
  lwpp = 22.4;                              // peak-to-peak linewidth (MHz)
};

spinB:                                      // nitroxide spin 2
{
  g = [2.0104, 2.0073, 2.0033];
  gStrain = [0.0004, 0.0003, 0.0001];
  n = [1];
  I = [1.0];
  A = [8.0, 6.0, 96.0];
  AStrain = [0.0, 0.0, 12.0];
  lwpp = 22.4;
};


# Fitting parameters
#------------------------------------------------
parameters = (
   {opt = 1;   mode = 1;   range = [1.50, 2.50]},        // Parameter 1:  distance mean 1
   {opt = 1;   mode = 1;   range = [0.00, 0.10]},        // Parameter 2:  distance width 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },        // Parameter 3:  xi mean 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },        // Parameter 4:  xi width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 5:  phi mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 6:  phi width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 7:  alpha mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 8:  alpha width 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },        // Parameter 9:  betta mean 1
   {opt = 1;   mode = 0;   range = [0.0, 90.0] },        // Parameter 10: betta width 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 11: gamma mean 1
   {opt = 1;   mode = 0;   range = [0.0, 180.0]},        // Parameter 12: gamma width 1
   {opt = 0;   mode = 1;   range = [1.50, 2.50]},        // Parameter 13: distance mean
   {opt = 0;   mode = 1;   range = [0.00, 0.10]},        // Parameter 14: distance width 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },        // Parameter 15: xi mean 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },        // Parameter 16: xi width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 17: phi mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 18: phi width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 19: alpha mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 20: alpha width 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },        // Parameter 21: betta mean 2
   {opt = 0;   mode = 0;   range = [0.0, 90.0] },        // Parameter 22: betta width 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 23: gamma mean 2
   {opt = 0;   mode = 0;   range = [0.0, 180.0]},        // Parameter 24: gamma width 2
   {opt = 0;   mode = 0;   range = [0.00, 1.00]},        // Parameter 25: ratio between 1 and 2
   {opt = 0;   mode = 0;   range = [0.00, 5.00]},        // Parameter 26: j coupling mean
   {opt = 0;   mode = 0;   range = [0.00, 5.00]},        // Parameter 27: j coupling width
   {opt = 1;   mode = 0;   range = [0.85, 0.95]}         // Parameter 28: pump pulse efficiency
);
```

```
# Genetic algorithm
#------------------------------------------------
genetic:
{
   num_generations_max = 500;            // Maximal number of generations
   size_generation = 128;                // Generation size (number of chromosomes)
   prob_crossover = 0.5;                 // Crossover rate
   prob_mutation = 0.01;                 // Mutation rate
   merit_function = 1;                   // Merit function: 1 - RMSD, 2 - RMSD/Pearson, 3 - Pearson
   num_averages = 100000;                // Number of Monte-Carlo averages
};


# Output data
#------------------------------------------------
output:
{
   directory = "Examples/Results/";          // Path to the disk space where the fitting results will be saved
   record_spectrum = 1;                      // Record the EPR spectrum of the spin system
   record_score = 1;                         // Record the goodness-of-fit as a function of the optimization step
   record_parameters = 1;                    // Record the best parameters of the model
   record_fit = 1;                           // Record the fit to the PELDOR signals
   record_form_factor = 1;                   // Record the form-factor for the PELDOR signals
   record_symmetric_solutions = 1;           // Record symmetry-related sets of the model's parameters
   record_error_plot = 1;                    // Record the RMSD as a function of the 'error_plot_variables'
   error_plot_variables = ([1,2]);           // Variables of the RMSD plot
   error_plot_size = 1000;                   // Number of points in the RMSD plot
};

error_plot_only:
{
   enable = 0;                                                // Switch to record only the error plot for the
                                                              // previously optimized fitting parameters
   error_plot_variables = ([1,2]);                            // Variables of the RMSD plot
   error_plot_size = 1000;                                    // Number of points in the RMSD plot
   input_directory = “Examples/Results/parameters.dat";       // Path to the file with the optimized fitting
                                                              // parameters
   output_directory = "Examples/Results/";                    // Path to the disk space where the error plot will
                                                              // be saved
};


### eof
```

Afterword

If you have any question about the performance or usage of the program, I am opened for discussion: abdullin@pc.uni-bonn.de

Any suggestions how to improve the code or the algorithm of the program are highly appreciated.

I thank Prof. Dr. Olav Schiemann and Dr. Gregor Hagelueken for the helpful discussions. I also acknowledge the community of www.stackoverflow.com.