PeldorFit User Manual

Dinar Abdullin
University of Bonn

abdullin@pc.uni-bonn.de

June 27, 2014

Table of contents

1 Introduction

The program PeldorFit performs 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) both spins are considered as single-point objects;

2) spin-spin distances have a normal distribution and all angular parameters have a uniform distribution;

3) 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. The detailed description of the model and its optimization is given 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/PeldorFit. The program is free and can be distributed under GNU General Public License.

This manual is organized in the following order. An installation of the program is described in Chapter 2. Running the program on PC is described in Chapter 3. 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. An example of using the program PeldorFit is given in Chapter 6.

2 Installation

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

3 Running the program

1. Open the Terminal (Linux) or Command Prompt (Windows).
2. Change the directory to the one where the PeldorFit executable file is stored:
   cd path/PeldorFit/Linux (for Linux)
   cd path/PeldorFit/Windows (for Windows).
3. Run the PeldorFit program by the following command:
   sh peldorfit.sh path_to_config/config.cfg (for Linux)
   peldorfit.exe path_to_config/config.cfg (for Windows)
   The configuration file *config.cfg* is a data file which contains all input parameters of the PeldorFit program. The detailed description of the configuration file is given in Chapter 4.

4 Configuration file

The PeldorFit configuration file is a data file which contains all input parameters of the program. Each input parameter is assigned to a certain variable whose name is recognized by the program at execution time. Similar approach is used in Bruker PulseSPEL programs, where the parameters of experiment are stored in configuration files. The PeldorFit configuration files have a *.cfg* extension. They can be created and edited with the usual text editors, e.g. Notepad++. The structure of the PeldorFit configuration file is described in the following.

The first part of the configuration file determines the names of experimental datasets and experimental settings:

Example

```
experimentals = (
  {filename = "Examples/bisnitroxide_XX.dat";   // the name of the data file
  detPiLength = 14;                             // the length of the detection pi-pulse (ns)
  detPiHalfLength = 7;                          // the length of the detection pi/2-pulse (ns)
  pumpPiLength = 14;                            // the 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; }
);
```

An *experimentals* variable joins together the names of PELDOR data files and the corresponding experimental settings. All PELDOR data sets specified in *experimentals* are fitted simultaneously (global fit) by the program. The parameters of each individual experiment are enclosed in the braces. The meaning of each parameter is

filename	A path to a data file which contains a background corrected PELDOR timetrace. The first column of this file should contain the values of time axis in [μs]. The second column should contain the values of a background-subtracted PELDOR signal normalized to 1. Such file can be created by the Bruker Xepr software or by the DeerAnalysis program (a file with the "fit" suffix).
detPiLength	The length of a detection π pulse in [ns]
detPiHalfLength	The length of a detection π/2 pulse in [ns]
pumpPiLength	The length of a pump pulse in [ns]
detFreq	The frequency of detection pulses in [GHz]
pumpFreq	The frequency of a pump pulse in [GHz]
magnField	The value of a magnetic field in [T]

Comments are indicated in the configuration file by # and // symbols. All comments are ignored by the program at execution time.

The second part of the configuration file determines the spectroscopic parameters of spin centers between which the distance is measured:

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, 6, 96];                        // hyperfine coupling constant A (MHz)
  AStrain = [0, 0, 12];                  // 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, 6, 96];
  AStrain = [0, 0, 12];
  lwpp = 22.4;
};
```

For definiteness, the spins are denoted as spin A and spin B. It is assumed that the resonance frequencies of both spins are determined by the Zeeman interaction, the hyperfine interaction of the electron and nuclear spins, and inhomogeneous line broadening. Unresolved anisotropy of the *g*- and *A*- tensors can also be taken into account via *g*-strain and *A*-strain. The description of each spectroscopic parameter is given below:

$g = [g_{xx}, g_{yy}, g_{zz}]$	*g*-tensor
gStrain = $[\Delta g_{xx}, \Delta g_{yy}, \Delta g_{zz}]$	*g*-strain (the unresolved anisotropy of the *g*-tensor)
n = [n1, n2]	The number of equivalent nuclei. The program allows using two sorts of equivalent nuclei. If the electron spin is coupled to one nucleus only, 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 the electron spin. In 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.
$A = [A1_{xx}, A1_{yy}, A1_{zz}, A2_{xx}, A2_{yy}, A2_{zz}]$	A hyperfine coupling tensor in [MHz]. This variable should consist of 3 or 6 components in case of one sort or two sorts of the 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 to the first sort of the nuclei only.
lwpp	A peak-to-peak linewidth of the inhomogeneous line broadening in [MHz].

The third part of the configuration file specifies optimization (fitting) parameters. The program optimizes the geometric model of a spin pair which should provide a fit to PELDOR time traces. This model is shown in Figure 1. It consists of two coordinate frames associated with the *g*-tensor principle axes of the two spins named here as 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 ***r***. In the reference coordinate system, ***r*** is described by spherical coordinates: length of the vector *r,* polar ξ and

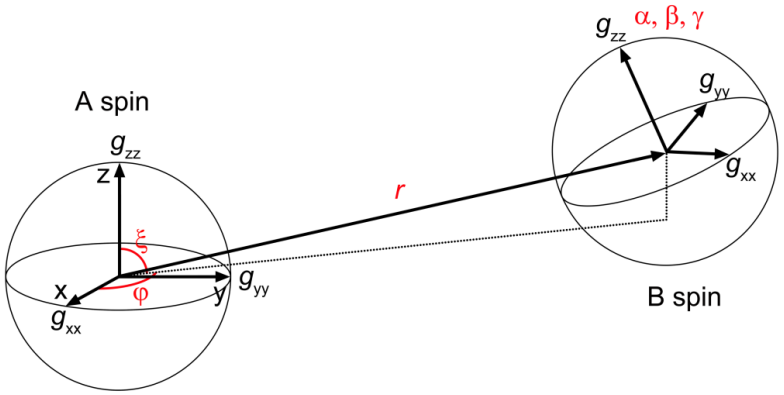

Figure 1. Geometric model of the spin pair. For definiteness, one spin is called A and another one is named B. The *g*-tensor of the A spin is supposed to coincide with the coordinate system of the model. The *g*-tensor of the B spin is defined by the Euler angles (α, β, γ) relative to spin A. The Euler angles are defined in accordance with the *z-x'-z"* convention. The inter-spin vector connecting the two spin centers is represented by the spherical coordinates (r, ξ, φ) relative to spin A. All geometric parameters have a distribution. For simplicity, the distance distribution is approximated by the normal distribution; α, β, γ, ξ and φ angles are supposed to be equally distributed in the intervals $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, $\Delta\xi$ and $\Delta\varphi$, respectively.

azimuthal φ angles. Due to the intrinsic flexibility of each molecule, the described geometric parameters usually have some distribution. The distribution of the inter-spin distances is approximated by the normal distribution, and the α, β, γ, ξ and φ angles are assumed to be uniformly distributed in the intervals $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$, $\Delta\xi$ and $\Delta\varphi$, respectively.

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 trace. If the magnetic tensors of both spins are rhombic, the full set of 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 uses the Gaussian distribution of spin-spin distances. However, this is not always the case e.g. when two or more distinct conformations of spin labels exist. To allow for the 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. As a consequence, the distance

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°]

distribution can be approximated by a sum of two Gaussian functions. The ratio between both conformations can be also varied.

The user may choose which geometric parameters will be optimized in the list named *parameters*:

Example

```
parameters = (
   {opt = 1;   range = [1.500, 2.500]},      // μ1 (nm)
   {opt = 1;   range = [0.000, 0.100]},      // σ1 (nm)
   {opt = 1;   range = [0.0, 90.0] },        // ξ1 (°)
   {opt = 1;   range = [0.0, 90.0] },        // Δξ1 (°)
   {opt = 1;   range = [0.0, 180.0] },       // φ1 (°)
   {opt = 1;   range = [0.0, 180.0] },       // Δφ1 (°)
   {opt = 1;   range = [0.0, 180.0]},        // α1 (°)
   {opt = 1;   range = [0.0, 180.0] },       // Δα1 (°)
   {opt = 1;   range = [0.0, 90.0] },        // β1 (°)
   {opt = 1;   range = [0.0, 90.0]},         // Δβ1 (°)
   {opt = 1;   range = [0.0, 180.0]},        // γ1 (°)
   {opt = 1;   range = [0.0, 180.0]},        // Δγ1 (°)
   {opt = 0;   range = [1.500, 2.500]},      // μ2 (nm)
   {opt = 0;   range = [0.000, 0.100]},      // σ2 (nm)
   {opt = 0;   range = [0.0, 90.0]},         // ξ2 (°)
   {opt = 0;   range = [0.0, 90.0]},         // Δξ2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // φ2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // Δφ2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // α2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // Δα2 (°)
   {opt = 0;   range = [0.0, 90.0]},         // β2 (°)
   {opt = 0;   range = [0.0, 90.0]},         // Δβ2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // γ2 (°)
   {opt = 0;   range = [0.0, 180.0]},        // Δγ2 (°)
   {opt = 0;   range = [0.000, 1.000]},      // ratio between 1 and 2
   {opt = 0;   range = [0.000, 0.001]},      // J (MHz)
   {opt = 0;   range = [0.000, 0.001]},      // ΔJ (MHz)
   {opt = 1;   range = [0.850, 0.950]}       // damping coefficient for the modulation depth parameter
);
```

In this list, all geometric parameters are given in the predefined order (see Appendix A). In order optimize a certain parameter, the corresponding switch called *opt* should be set to 1, and to 0 otherwise. Moreover, the ranges of optimized parameters should be specified in the *range* arrays.

Additionally to the geometric parameters, the *parameters* list contains few other possible fitting variables. One of them is an exchange coupling constant, which is given in a form of a mean value J and its distribution width ΔJ, both in [MHz]. Another parameter is a phenomenological coefficient η ($0 < \eta \leq 1$) which takes into account factors affecting the modulation depth parameter: non-ideal pulses, incomplete spin labelling, etc. If this coefficient is optimized, the modulation depths of all fitted PELDOR time traces are multiplied by the value of η. In some cases, η can also differ for the PELDOR time traces fitted simultaneously. In order to fit this coefficient separately for each individual time trace, the line in the *parameters* list corresponding to the η parameter should be duplicated as many times as the number of time traces. For example, if two time traces are fitted together with the different η coefficients, the *parameters* list should be extended to:

Example

```
parameters = (
   …
   {opt = 0;   range = [0.000, 0.001]},        // ΔJ (MHz)
   {opt = 1;   range = [0.850, 0.950]},        // damping coefficient for the modulation depth parameter #1
   {opt = 1;   range = [0.850, 0.950]}         //  damping coefficient for the modulation depth parameter #2
);
```

Next, a merit function characterizing a goodness-of-fit has to be set. The program offers three merit functions: root-mean square deviation (RMSD), Pearson product moment correlation coefficient (PCC), and their combination RMSD/PCC. Each merit function is assigned to a certain value of the *scoreIndex* variable: *scoreIndex = 0* for RMSD, *scoreIndex = 1* for RMSD/PCC, and *scoreIndex = 2* for PCC. The user may choose one of them:

Example

```
scoreIndex = 0;                    // RMSD is used as a  merit function
```

The optimization problem is solved by a genetic algorithm. This algorithm has several intrinsic parameters, which determine its ability to find the global minimum of the optimization problem. In order to allow the user to change these parameters, if required, they are specified in the configuration file:

Example

```
genetic:
{
   nAvg = [100000L, 1000000L];     // Number of Monte-Carlo averages
   genSize = 128;                  // Number of chromosomes in one generation
   genMax = [499, 1];              // Max number of generations
   pCros = 0.3                     // Crossover rate
   pExch = 0.5                     // Uniform Crossover rate
   pMut = 0.05;                    // Mutation rate
   ratioCros = 0.5;                // Uniform Crossover to Single Point Crossover ratio
   ratioMut = 0.5;                 // Small Creep Mutation to Random Mutation ratio
   elitism = FALSE;                // Switch for saving of the best chromosome (elitism)
}
```

The description of each parameter is given below:

genetic A variable which joins together all parameters of the genetic algorithm.

nAvg The number of Monte-Carlo averages (the number of conformers of a spin pair) used to simulate each PELDOR time trace.

genSize The size of one generation (the number sets of fitting parameters, for which the goodness-of-fit is calculated in each optimization step).

genMax The maximal number of generations (optimization steps). This parameter is employed as a stopping criterion for the genetic algorithm.

pCros Crossover rate: it determines a probability that the genes of two chosen chromosomes will be mixed.

pExch Exchange rate: it determines a probability that an individual gene will be interchanged during the uniform crossover.

pMut Mutation rate: it determines a probability that the genes of a chosen chromosome will be mutated.

ratioCros The ratio between the uniform crossover and the single point crossover.

ratioMut The ratio between the small-creep mutation and the random mutation.

elitism The switch for the elitism.

The *nAvg* variable can be varied during the fitting. The number of generations (optimization steps) corresponding to each value of *nAvg* can be set in the *genMax* variable. In the example above, 10^5 Monte-Carlo averages are used for the calculations in first 499 generations, whereas the following 1 generation uses 10^6 Monte-Carlo averages.

This procedure allows to reduce the Monte-Carlo noise in the output data exported by the program.

The last part of the configuration files determines some settings of an output data:

Example

```
# Output data
#----------------------------
OutputDirectory = "Examples/Results/";   // Path to the disk space where the fitting results will be saved

Spec = TRUE;                             // Record the EPR spectrum of the spin pair

SymSol = TRUE;                           // Create symmetry-related solutions

ModDepth = TRUE;                         // Record the angular dependence of the modulation depth
                                         // parameter
ErrorProfile = TRUE;                     // Record the RMSD surface for the chosen fitting parameter
ErrorProfileVar = (                      // Default: the RMSD surface is recorded for μ1 and σ1
   {var = 2;   range = [0.0, 90.0] },    // ξ1
   {var = 3;   range = [0.0, 90.0] }     // Δξ1
);
```

These variables determine:

OutputDirectory The path to a disk space to which the fitting results will be saved.

Spec If set to *TRUE*, the EPR spectrum of the specified spin pair will be simulated after first generation (optimization cycle). The abscise of this spectrum is given in frequency units [GHz], while the magnetic field is set to a constant value specified for the first experiment in the *experimentals* list. In addition to the spectrum, the excitation profiles of the pump and detection pulses are calculated for the first experiment in the *experimentals* list.

SymSol If set to *TRUE*, the symmetry-related sets of fitting parameters will be generated, after the fitting procedure has been finished.

ModDepth If set to *TRUE*, the angular dependence of the modulation depth parameter will be simulated for each time trace based on the optimized geometric model.

ErrorProfile If it is set to *TRUE*, the RMSD between the experimental time traces and their fits will be recorded as a function of chosen fitting parameter(s). The variables of the RMSD plot are specified in the *ErrorProfileVar* variable. While the RMSD plot is recorded for a number of chosen fitting parameters, other fitting parameters are set to their optimized values.

ErrorProfileVar (continuation for *ErrorProfileVar*)

This variable determines which fitting parameters will be used as variables of the RMSD plot. If no parameters are specified in this variable, i.e. *ErrorProfileVar = ()*, the RMSD will be recorded by default for the spin-spin distance μ_1 and its standard deviation σ_1. These two parameters will be varied in the ranges specified for them in the *parameters* list. To set the variables of the RMSD plot to other fitting parameters, the numbers of fitting parameters (*var*), as pre-defined in the parameters list (Appendix A), and the corresponding variation ranges (*range*) should be provided. For example, the expression

```
ErrorProfileVar = (
   {var = 2;   range = [0.0, 90.0] },
   {var = 3;   range = [0.0, 90.0] }
);
```

means that the RMSD will be recorded for the ξ_1 and $\Delta\xi_1$ parameters, varying them in the range [0°, 90°].

Note that the *Spec, SymSol, ModDepth, ErrorProfile* variables should be set to one of two possible values, *TRUE* or *FALSE*.

5 Output data

The following output files are created by the PeldorFit program:

fitN_M — These files contain the fits of the PELDOR time traces specified in the *experimentals* list. *N* denotes to the number of generation (optimization step) and *M* is the number of a time trace as it appears in the *experimentals* list (the numbering starts from 0). To save the computational time, these files are created for each 50-th generation only. The first column of the file contains the values of a time axis in [µs], the second and third columns contain to the values of the PELDOR signal and its fit, correspondingly.

score — This file contains best values of goodness-of-fit parameter found for each generation (optimization step).

param — This file contains the best values of the fitting parameters found for each generation (optimization step). The parameters of *n-th* generation are listed in the *n*-th row of the file. The order and units of the fitting parameters are the same as in the *parameters* list.

spectrum	This file contains the EPR spectrum of a spin pair in the frequency domain. The spectrum is simulated for a constant magnetic field which equals to the value of the *magnField* parameter of the first experiment specified in the *experimentals* list. The first column of the file contains the values of a frequency axis in [GHz]. The second column contains the values of the spectrum normalized to 1. The third and fourth columns contain the excitation profiles of the pump and detection pulses, correspondingly. These profiles are simulated for the first experiment specified in the *experimentals* list.
symmetricSolutions	This file contains the symmetry-related sets of the fitting parameters and their RMSD. In analogy to the *param.dat* file, each set of the parameters is listed in an individual row. The last number of each row corresponds to the RMSD value of the current solution.
lambdaM	These files contain the angular dependences of the modulation depth parameter simulated for each PELDOR time trace. *M* denotes the number of a time trace as it appears in the *experimentals* list (the numbering starts from 0). The first column of the file contains the values of the dipolar angle in [°]. The second column contains the values of the modulation depth parameter normalized to its integral.
errorprofile_p1_p2	This file contains *RMSD* values corresponding to the different values of fitting parameters specified in the *ErrorProfileVar* variable. *p1* and *p2* denote the indices of the chosen variables (in case of two variables). Each line of this file contains the values of the varied fitting parameters and the corresponding RMSD value.

6 Example

This chapter shows an example of using PeldorFit for a PELDOR data-set stemming from a nitroxide biradical **1** (Figure 2). This data-set was acquired at W-band frequencies and consists of 6 time traces corresponding to the different combinations of the microwave frequencies/magnetic fields. Before being used in PeldorFit, all PEDOR time traces were background corrected by means of the DeerAnalysis program.

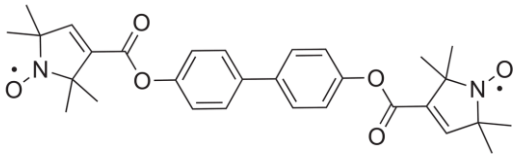

Figure 2. Structure of the nitroxide biradical **1**.

The fitting of PELDOR data by the PeldorFit program can be done in following steps:

1. Open the Terminal (Linux) or Command Prompt (Windows).
2. Change the directory to the one where the PeldorFit executable file is stored:
   cd path/PeldorFit/Linux (for Linux)
   cd path/PeldorFit/Windows (for Windows)
3. Run the program by the following command:
   sh peldorfit.sh "*Examples/bisnitroxide_config.cfg*" (for Linux)
   peldorfit.exe "*Examples/bisnitroxide_config.cfg*" (for Windows)
   The *bisnitroxide_config.cfg* is a configuration file. Its content is given in Appendix B.
4. While the PeldorFit program is running, it outputs the number of a current optimization step (generation) and the corresponding value of the merit function into the Terminal/Command Prompt line:

   Example

   Number of PELDOR time traces: 6
   **** Optimization via genetic algorithm ****
   Generation no. 1 Score = 0.125099
   ...
   Generation no. 100 Score = 0.0268819

   The fitting procedure stops when the number of the optimization steps (generations) reaches the *genMax* value. During/after the fitting, the output data can be found in the *Examples/Results* directory.

The representation and interpretation of the fitting results for the biradical **1** is given in:

- D. Abdullin, G. Hagelueken, R.I. Hunter, G.M. Smith, O. Schiemann, *Molecular Physics* **2014**, DOI: 10.1080/00268976.2014.960494.

Appendix A. Numbering of fitting parameters

No.	Parameters	Units
0	μ (1st conformation)	nm
1	σ (1st conformation)	nm
2	ξ (1st conformation)	degrees
3	$\Delta\xi$ (1st conformation)	degrees
4	φ (1st conformation)	degrees
5	$\Delta\varphi$ (1st conformation)	degrees
6	α (1st conformation)	degrees
7	$\Delta\alpha$ (1st conformation)	degrees
8	β (1st conformation)	degrees
9	$\Delta\beta$ (1st conformation)	degrees
10	γ (1st conformation)	degrees
11	$\Delta\gamma$ (1st conformation)	degrees
12	μ (2nd conformation)	nm
13	σ (2nd conformation)	nm
14	ξ (2nd conformation)	degrees
15	$\Delta\xi$ (2nd conformation)	degrees
16	φ (2nd conformation)	degrees
17	$\Delta\varphi$ (2nd conformation)	degrees
18	α (2nd conformation)	degrees
19	$\Delta\alpha$ (2nd conformation)	degrees
20	β (2nd conformation)	degrees
21	$\Delta\beta$ (2nd conformation)	degrees
22	γ (2nd conformation)	degrees
23	$\Delta\gamma$ (2nd conformation)	degrees
24	ratio between 1st and 2nd conformations	-
25	J	MHz
26	ΔJ	MHz
27 and higher	η	-

Appendix B. Example of the configuration file

The content of the *bisnitroxide_config.cfg* configuration file is listed.

Example

```
#  PeldorFit Configuration File

# Experimental parameters
#----------------------------
experimentals = (
  {filename = "Examples/bisnitroxide_XX.dat";      // the name of the data file
  detPiLength = 14;                               // the length of the detection pi-pulse (ns)
  detPiHalfLength = 7;                            // the length of the detection pi/2-pulse (ns)
  pumpPiLength = 14;                              // the 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:
{
  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, 6, 96];                        // hyperfine coupling constant A (MHz)
  AStrain = [0, 0, 12];                   // 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, 6, 96];
  AStrain = [0, 0, 12];
  lwpp = 22.4;
}

# Fitting parameters
#----------------------------
 parameters = (
   {opt = 1;   range = [1.500, 2.500]},     // distance mean 1
   {opt = 1;   range = [0.000, 0.100]},     //  distance width 1
   {opt = 1;   range = [0.0, 90.0] },       // ksi mean 1
   {opt = 1;   range = [0.0, 90.0] },       // ksi width 1
   {opt = 1;   range = [0.0, 180.0] },      // phi mean 1
   {opt = 1;   range = [0.0, 180.0] },      // phi width 1
   {opt = 1;   range = [0.0, 180.0]},       // alpha mean 1
   {opt = 1;   range = [0.0, 180.0] },      // alpha width 1
   {opt = 1;   range = [0.0, 90.0] },       // betta mean 1
   {opt = 1;   range = [0.0, 90.0]},        // betta width 1
   {opt = 1;   range = [0.0, 180.0]},       // gamma mean 1
   {opt = 1;   range = [0.0, 180.0]},       // gamma width 1
   {opt = 0;   range = [1.500, 2.500]},     // distance mean 2
   {opt = 0;   range = [0.000, 0.100]},     // distance width 2
   {opt = 0;   range = [0.0, 90.0]},        // ksi mean 2
   {opt = 0;   range = [0.0, 90.0]},        // ksi width 2
   {opt = 0;   range = [0.0, 180.0]},       // phi mean 2
   {opt = 0;   range = [0.0, 180.0]},       // phi width 2
   {opt = 0;   range = [0.0, 180.0]},       // alpha mean 2
   {opt = 0;   range = [0.0, 180.0]},       // alpha width 2
   {opt = 0;   range = [0.0, 90.0]},        // betta mean 2
   {opt = 0;   range = [0.0, 90.0]},        // betta width 2
   {opt = 0;   range = [0.0, 180.0]},       // gamma mean 2
   {opt = 0;   range = [0.0, 180.0]},       // gamma width 2
   {opt = 0;   range = [0.000, 1.000]},     // ratio between 1 and 2
   {opt = 0;   range = [0.000, 0.001]},     // exchange coupling mean
```

```
   {opt = 0;   range = [0.000, 0.001]},      // exchange coupling width
   {opt = 1;   range = [0.850, 0.950]}       // damping coefficient for the modulation depth parameter
);

# Merit function
# 0 - RMSD; 1 - RMSD/PCC; 2 - PCC
scoreIndex = 0;                              // RMSD is used as a  merit function

# Genetic algorithm
#----------------------------
genetic:
{
   nAvg = [100000L, 1000000L ];              // Number of Monte-Carlo averages
   genSize = 128;                            // Number of chromosomes in one generation
   genMax = [499, 1];                        // Max number of generations
   pCros = 0.3                               // Crossover rate
   pExch = 0.5                               // Uniform Crossover rate
   pMut = 0.05;                              // Mutation rate
   ratioCros = 0.5;                          // Uniform Crossover to Single Point Crossover ratio
   ratioMut = 0.5;                           // Small Creep Mutation to Random Mutation ratio
   elitism = FALSE;                          // Switch for saving of the best chromosome (elitism)
}

# Output data
#----------------------------
OutputDirectory = "Examples/Results/";       // Path to the disk space where the fitting results will be saved

Spec = TRUE;                                 // Record the EPR spectrum of the spin pair

SymSol = TRUE;                               // Create symmetry-related solutions

ModDepth = TRUE;                             // Record the angular dependence of the modulation depth
                                             // parameter
ErrorProfile = TRUE;                         // Record the RMSD surface for the chosen fitting parameter
ErrorProfileVar = ();                        // Default: the RMSD surface is recorded for distance mean 1 and
                                             // distance width 1

### 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.