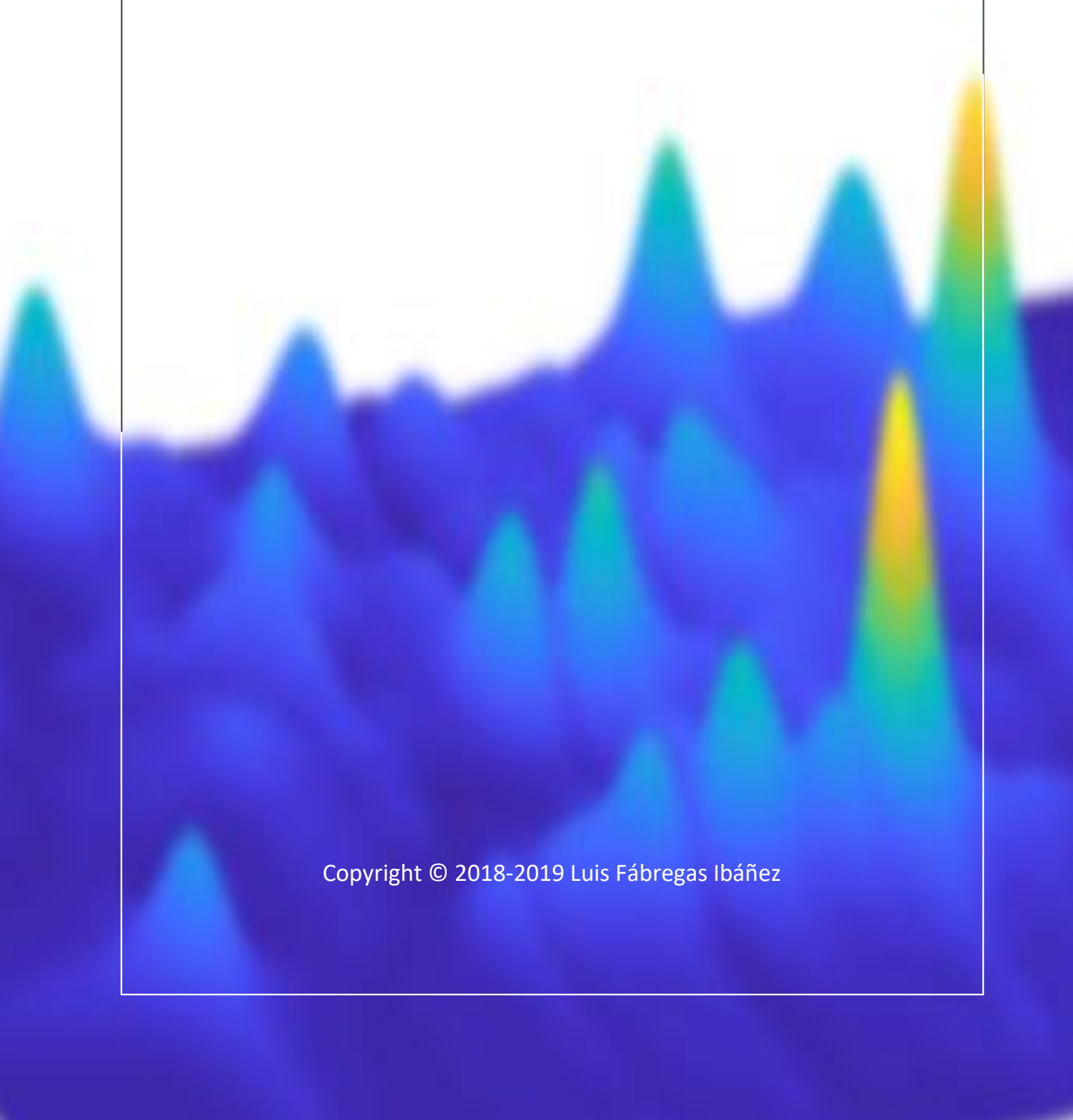
USER GUIDE - MANUAL

HYSCOREAN



Copyright © 2018-2019 Luis Fábregas Ibáñez

Contents

1. Preface

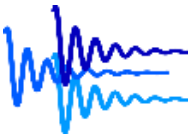
Thank you for downloading Hyscorean (i.e. HYSCORE analysis), a free software for the processing and manipulation of experimental HYSCORE spectra controlled via a graphical user-interface. Hyscorean enables the processing of uniform and non-uniform sampled HYSCORE experiments, background fitting, reconstruction of the signals, signal processing, validation and fitting of the spectra using EasySpin. Therefore, Hyscorean takes care of everything from the raw HYSCORE data to the fitted results ready for interpretation. This user-guide describes all elements, procedures and tricks in Hyscorean so that the user can exploit all the features available.

Processing of NUS HYCORE data is irrelevant if the measurements are impossible with common commercial spectrometers. Therefore, Hyscorean offers the possibility to measure NUS HYSCORE with commercial Bruker spectrometers. This manual also provides a guide to setting up such measurements as well as the source code for the measurement programs.

This program is a free open-source software project: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation 3.0 or any later version. This software has been written for the MATLAB environment and, therefore, is licensed under the GNU Lesser General Public License (LGPL) 3.0.

I want to acknowledge Stefan Stoll and Bradley Worley for their open-source policies on the source code of their software: EasySpin and CAMERA, respectively, which have allowed Hyscorean to reach this state.

Enjoy processing with Hyscorean,

Luis Fábregas Ibáñez 

2. Getting started

This section will cover the basic operations to be done by the user to install and prepare Hyscorean to its full use. Installation and basic default definitions will be explained step-by-step.

2.1 Installation

To install Hyscorean and set up all the environment variables required by the software to work the following instructions need to be followed:

I. First all files must be extracted from the ZIP file (downloaded from the EPR@ETH homepage or from the official GitHub server) and saved on a directory/folder of choice.

II. Open MATLAB and change the current path to the Hyscorean installation folder. It is NOT necessary to add all Hyscorean files to the MATLAB path search.

III. Execute the script setup_hyscorean from the MATLAB console and wait until execution finishes. This installation program will take care setting up everything necessary in MATLAB for Hyscorean to work properly. More precisely it takes care of the following:

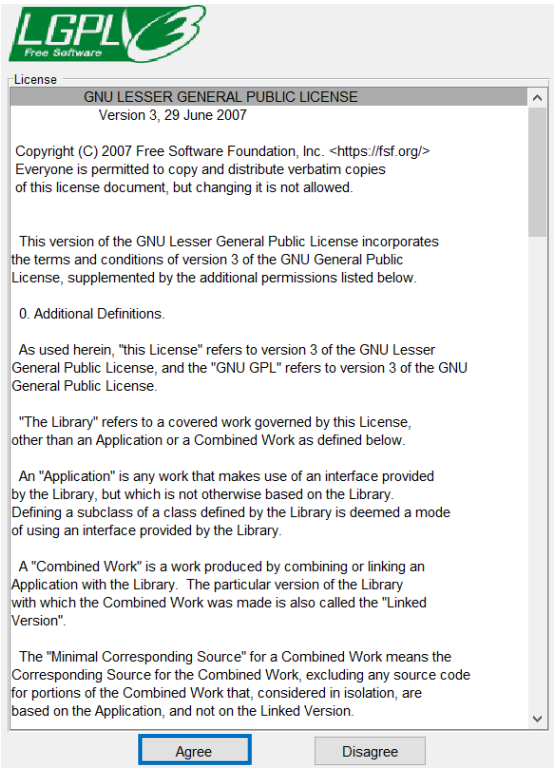
a) All required paths by Hyscorean are saved into the path search of MATLAB for further sessions.

b) The installation and license of required MATLAB & Simulink packages are controlled and the functionality of Hyscorean is adapted to missing packages.

c) The proper installation of EasySpin is controlled and again the functionality of Hyscorean is adapted if missing or not installed.

d) Default user-preferences are set in the MATLAB preferences to allow Hyscorean to keep user-defined defaults between sessions.

Before these actions are taken the user must agree to the terms of the GNU LGPL 3.0 license agreement which appears at the start of the setup.



By pressing the button Agree, the license is admitted, and the setup proceeds as described. Otherwise, Hyscorean cannot be used until the license agreement is accepted.

IV. Once the setup program finishes the setup a message will be printed on the console to indicate the outcome of the installation.

A successful installation is indicated by the message:

```
Hyscorean was successfully and fully installed and all functionalities
are operational.
```

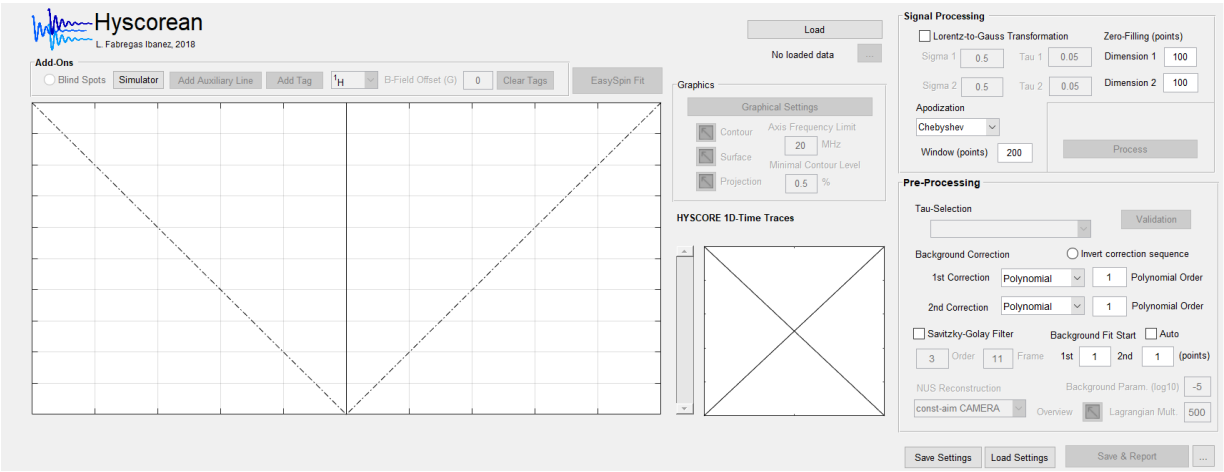
Indicating that the installation proceeded smoothly without any problems and that Hyscorean is ready to be used.
In case some of the external packages is not available these will be individually notified during the installation and once finishes will be indicated by the message:

```
Hyscorean was successfully but incompletely installed.
```

NOTES:
Once the user defines its own defaults, these are stored in the Hyscorean preferences, re-running setup_hyscorean will NOT overwrite any of these by the defaults set during the first installation. Missing packages which are re-installed properly will not be identified until setup_hyscorean is executed again.

2.2 Hyscorean's GUI

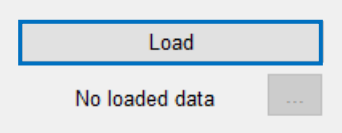
Once installation is finished, Hyscorean can be called from any path in MATLAB by executing the command Hyscorean from the console. This will prompt the Hyscorean graphical user-interface (GUI) to appear:



At this point most of the user-interface controls will be either deactivated or not visible. As a general feature of Hyscorean, the program will activate and deactivate buttons on the GUI according to what the user can and cannot do at that moment. This prevents the user from doing any action which would lead to a crash or error.

2.3 Loading data into Hyscorean

Files containing HYSCORE data can be loaded into the program via the Load button:



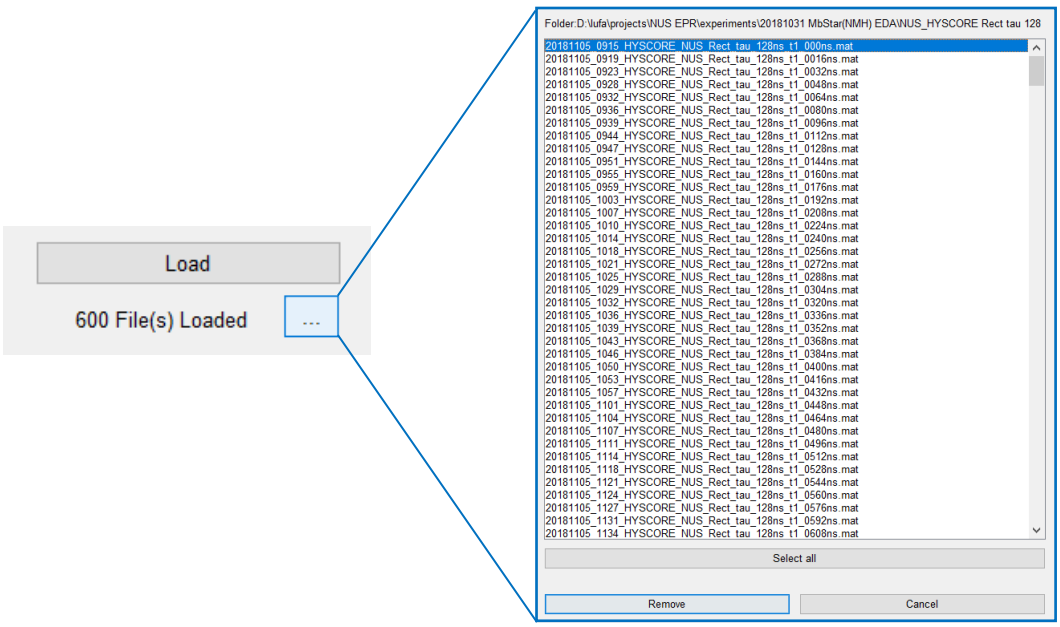
By pushing the button an OS window will appear requesting the user to select the files to be loaded. Multiple files can be loaded by selecting all of them in the window.

Hyscorean is compatible with the following file extensions:

.DSC	Bruker BES3T data file format
.DTA	Bruker BES3T data file format
.mat	MATLAB output file format
.txt	Text files in ASCII format

NOTES:
Loading new data will reset Hyscorean deleting any data stored into the variables as well as all displays. Loading multiple files is required by measurement done by the EPR@ETH AWG spectrometer. For BES3T format files a single file must be loaded.

Once the file(s) have been loaded, the indicator below the Load button will display how may files were loaded. The user can control at any moment which files are currently loaded into the program by pressing the (...) button next to the display. This will open a window with a list of all loaded files where the user can remove loaded files selectively:



2.4 Mounting of HYSCORE data

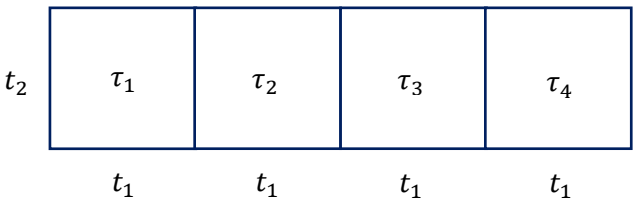
Once the files are loaded the program automatically will start to mount the data into MATLAB variables which the program will manipulate from that point on. The protocol for mounting the data is different for all the file extensions. In the following a short description of the different mounting protocols will be presented. For a description of the mounting of NUS data files see section 4.1.

NOTE:
Once the data has been mounted (for all file formats), the program will automatically set the zero-filling and apodization window lengths in the Processing panel to the corresponding dimensions of the mounted signal.

2.4.1 Mounting Bruker BES3T data

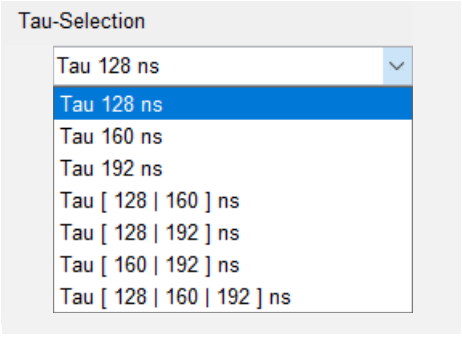
Data obtained from commercial Bruker spectrometers is mounted by loading the .DSC or .DTA files. All experimental parameters and sweep axes are extracted from the descriptors on those files.

Due to the convenience of saving all τ-values employed in the HYSCORE experiments into one file, Hyscorean expects the an $NxNM$ data grid where N is the dimension of the HYSCORE signal and M is the number of τ-values measured during the experiment. Therefore, the .DSC and .DTA files are expected to have the following structure:



The τ-values employed during the experiments are then extracted from the PulseSPEL program stored into the descriptors. It is important that the τ-values defined in the PulseSPEL program are stored in the delay d1 (the default delay variable used by XEPR). Hyscorean also automatically detects the number of folded experiments (i.e. number of τ-values employed) without any further input from the user.

Once the data has been mounted, all different τ-values and their possible combinations will be updated for the user to choose from the Tau-selection list box:



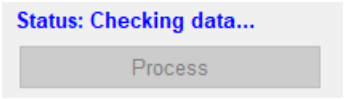

```
begin exp [INTG QUAD]
for k=1 to n
totscans (n)
scansdone (k)
dy=0
d1=144
for y=1 to 512
        sweep x =1 to sx
                shot i = 1 to h
                        p1[ph0]
                        d1
                        p1[ph0]
                        d1
                        dy
                        p0 [ph1]
                        d1
                        dx
                        p1 [ph2]
                        d1
                        d0
                        acq [sg1]
                next i
                dx = dx +d30
                next x
        dx=0
        dy=dy+d31
       next y
next k
end exp
```

NOTE:
The τ -values combination determines which of the measured HYSCORE signals are added (in time-domain) together prior signal processing. Thus, choosing a different combination will restart the processing from the beginning (see later).

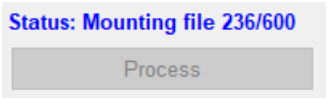
At this point the different parameters for processing of the HYSCORE signal can be selected and the processing started (see 3. Basic HYSCORE processing).

2.4.2 Mounting EPR@ETH AWG data

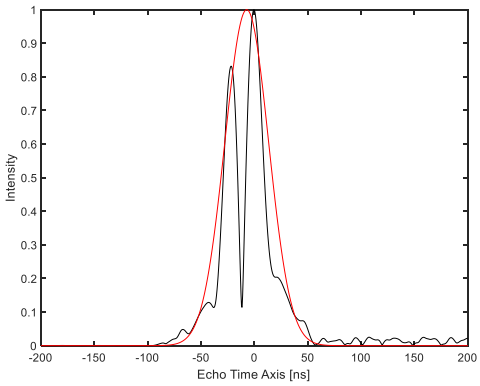
Data obtained from this home-built spectrometer is stored in multiple files, each of them containing a single sweep along one dimension. Therefore, all files must be loaded to mount all of them into a single variable. In contrast to commercial Bruker spectrometers, the data contains the raw echoes and not the integral. First the echoes are extracted from the output files and mounted. This starts with a control protocol where the consistency between all loaded files of the size of the echo arrays is checked. This procedure is reported via the status display:



The echoes are then mounted. This procedure can be rather long due to the need to pre-process each echo prior to mounting. The user can monitor the progress again via the status display:



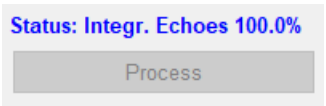
Next, the echoes are integrated according to the following procedure:



I. A Gaussian function if fitted to the absolute intensity of the first echo of the mounted data.
II. The fitted function is employed as a window and applied to each echo (as a matched filter)
III. The filtered echoes are integrated

NOTE:
The Gaussian matched filtering can be changed in the MATLAB function integrateEcho.m to boxcar integration.

This procedure can also be monitored via the status bar:



The τ-values are extracted from the experimental data structure and all their combination again displayed in the Tau-selection list box for the user to choose.

2.4.3 Mounting ASCII formatted data

If the user has HYSCORE data files to process via Hyscorean, which do not pertain to any of the above described formats, these can still be loaded via ASCII formatted files. The program will consider all lines starting with % as comments and ignore them.

The files must contain five columns and a row per measured point. For each point the corresponding t1 and t2 times must be given in the first and second column in nanoseconds, respectively. The real and imaginary part of the data point are then given in the third and fourth column. Should the data contain only real values, the fourth column must be set to zeroes. The τ-value employed for the experiment to which the data point belongs must be then given as the fifth column in nanoseconds. Summarizing, the ASCII file should have the following format:

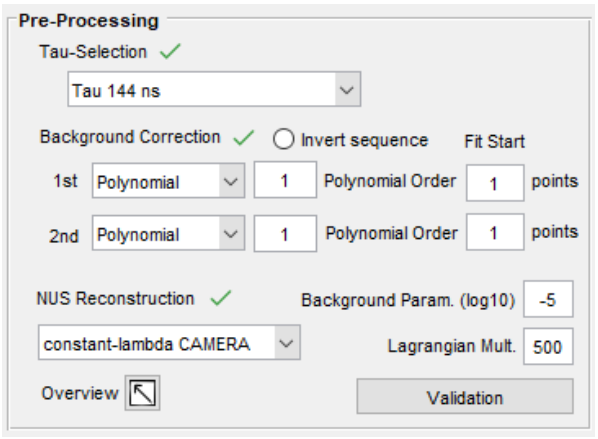
```
%-------------------------------------------------------------------------------
% t1-Timings[ns] t2-Timings[ns]   Real Signal    Imag Signal  Tau-Values[ns]
%-------------------------------------------------------------------------------
       3200           2782          -27784         -11212         128
       3200           2798          -31112         -18688         128
       3200           2814          -20220         -14740         128
       3200           2830          -8132          -2292          128
       3200           2846          -1072           3768          128
       3200           2862           2624           10380         128
       3200           2878           3300           11832         128
       3200           2894          -692            10684         128
       3200           2911          -10764          5592          128
       3200           2927          -30208         -5752          128
       3200           2943          -46836         -12508         128
       3200           2959          -54712         -14696         128
       3200           2975          -49192         -7840          128
       3200           2991          -40796          752           128
       3200           3007          -35592          10012         128
       3200           3023          -24184          14036         128
       3200           3039          -13828          14624         128
       ...            ...             ...            ...           ...
```

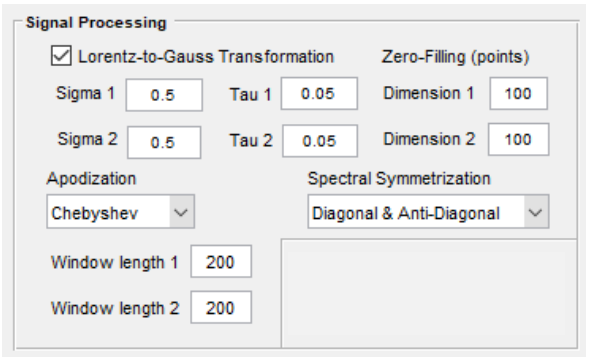
Using this ASCII format, all data corresponding to HYSCORE experiments measured at different tau-values can be given as input in a single compact file. As in the previous sections, the program automatically mounts the HYSCORE signals by their corresponding τ-values and their possible combinations are updated for the user to choose from the Tau-selection list box.

The mounting protocol also automatically checks the uniformity of the input data to check whether the experiment was performed under non-uniform sampling conditions. For cases where the data is non-uniform refer to section 4.1. for further details.

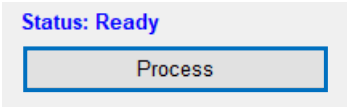
3. Basic HYSCORE Processing

This section presents the basics of Hyscorean to get from the mounted HYSCORE signal to a processed spectrum. The processing is divided into two clear blocks: the pre-processing and the processing part. The parameters to be set by the user are separated into two different panels (which are activated once the mounting finished) in the Hyscorean GUI:

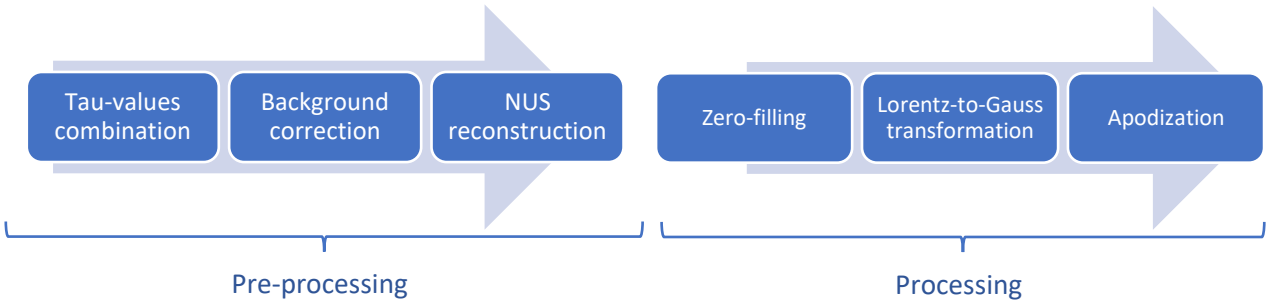




Hyscorean's processing protocol is started by pressing the Process button

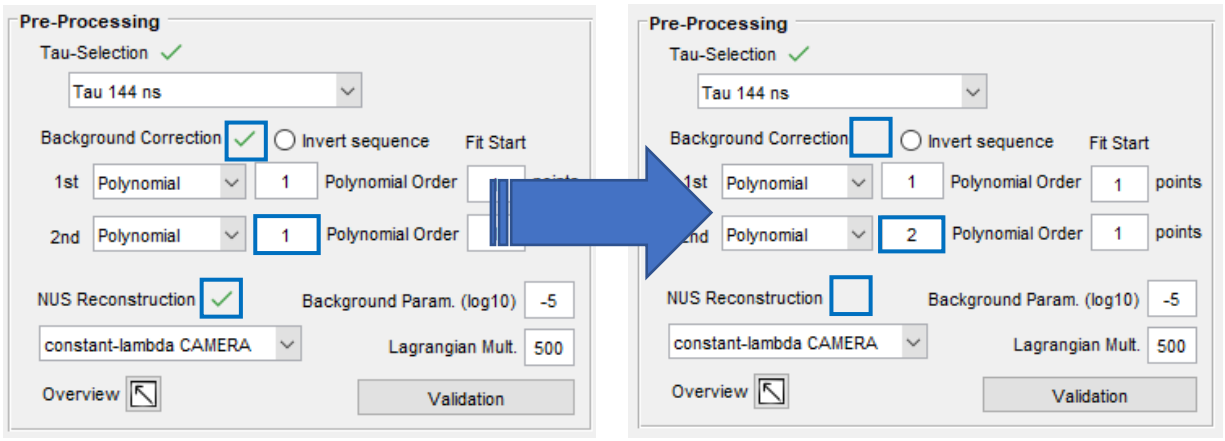


and then follows this structure:



To enhance the processing times of Hyscorean a series of switches are employed at each step of the pre-processing. Once the Process button is pressed and the processing finishes, all switches are activated and pressing the Process button again will result in the pre-processing being skipped. This allows to avoid performing computationally costly procedures such as background correction or NUS reconstruction (which is always skipped for non-NUS signals) when there are only changes in the processing part. If any of the parameters in the Pre-processing panel is modified the corresponding switches will be deactivated and the processing will start at that point of the pre-processing.

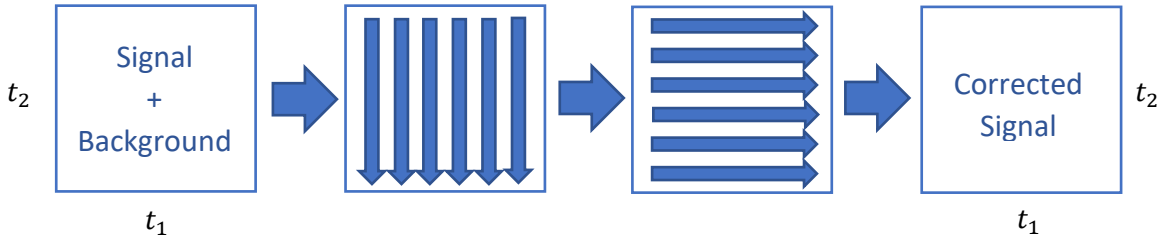
The steps of the pre-processing which have been finished and will be skipped are denoted by the ✓ marker next to the panel components.



In the following sections the different procedures in the whole processing protocol will be presented and described. The NUS reconstruction will be discussed in section 4.

3.1 Background correction

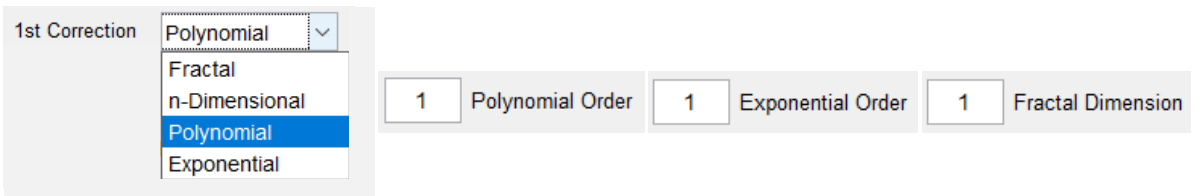
The background or baseline correction follows the combination of the different HYSCORE signals at different τ-values. The correction of the 2D-background is performed sequentially. First the baselines of the single 1D-traces are corrected along one dimension and then repeated along the other.



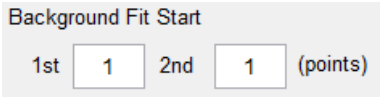
To each trace along a first dimension, a model is fitted and then subtracted to each of them. Next, another model is fitted to the traces along the second dimension and again subtracted. The order in which the background is corrected can be inverted by enabling the Invert correction sequence option. The background models available in Hyscorean are:

- Polynomial of n-th order $B(t) = \sum_{i=0}^{n} c_i t^i$
- Exponential (polynomial fitted to the logarithm) $\log(B(t)) = \sum_{i=0}^{n} c_i t^i$
- Stretched exponential $B(t) = \exp(-ct^{n/3})$
- Fractal $B(t) = \exp(-ct^{n/3})$

The model to fit along each dimension can be selected under the Background correction section. For each model the dimensionality parameter n can be given in the Polynomial Order, Exponential Order or Fractal Dimension edit box next to the model list:



The background if fitted to the whole trace starting from the first point. However, the point at which the model starts to be fitted can be adjusted manually from the Background start edit boxes:



For complex-valued signals the background correction is performed separately for the real and imaginary parts of the signal. Prior to background corrections the phase of the signal is quickly optimized such that the real component is maximized with respect to the imaginary component. At the start of the background correction the signal components are separated and to each of them a background is fitted and corrected along both dimensions. The real and imaginary parts are then combined back into the complex-valued background-corrected signal.

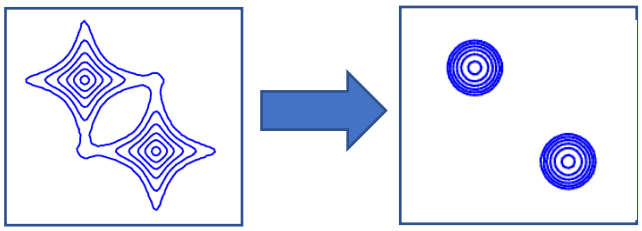
For non-NUS data this represents the end of the pre-processing and leads to the start of the processing.

3.2 Processing

The processing starts by zero-filling the pre-processed signal (i.e. appending a certain number of zeros to the end of the signal). By default, the program automatically sets this number to the dimensions of the signal so that the full information of the signal is recovered later in the Fourier transform. Any further zeros can be appended by modifying the Zero-filling edit boxes in the Processing panel for aesthetic changes. The zero-filling can be disabled by setting the values in the edit boxes to zero. However, this is not recommended due to the consequent loss of information.

3.2.1 Lorentz-to-Gauss transformation

Exponentially decaying time-domain signals such as in HYSCORE correspond to Lorentzian line shapes in frequency domain. The Lorentzian line shapes are characteristic for having long tails which can become very damaging in 2D-spectra such as HYSCORE's due to possible overlaps between crosspeak tails appearing as spurious crosspeaks [1].



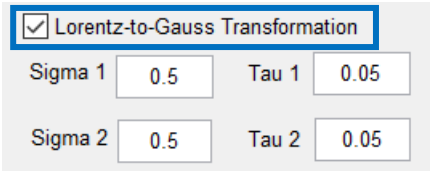
The Lorentz-to-Gauss transformation [2] (also known as Gaussian multiplication or double exponential transformation [3]) allows to convert the Lorentzian line shape to the Gaussian line shape which exhibits narrower tails . Therefore, such overlaps between crosspeak tails can be avoided. This transformation is achieved by applying the following window to the signal:

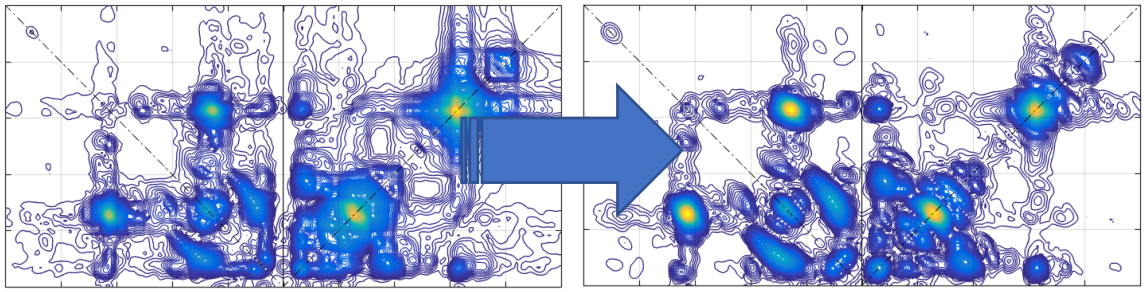
$$W(t_1, t_2) = e^{t_1/\tau_1 - (\sigma_1^2 t_1^2)/2} e^{t_2/\tau_2 - (\sigma_2^2\, t_2^2)/2}$$

where the parameters in Hyscorean are computed as:

$$\sigma_i = \frac{\widehat{\sigma_i}}{\tau_i} \qquad \tau_i = t_{i,max}\widehat{\tau_i}$$

The factors $\hat{\sigma}_i$ and $\hat{\tau}_i$ can be given through Hyscorean's Processing panel in the the Sigma1/Sigma2 and Tau1/Tau2 edit boxed. By default, Lorentz-to-Gauss transformation is disabled in Hyscorean until the corresponding Lorentz-to-Gauss Transformation check box is enabled:



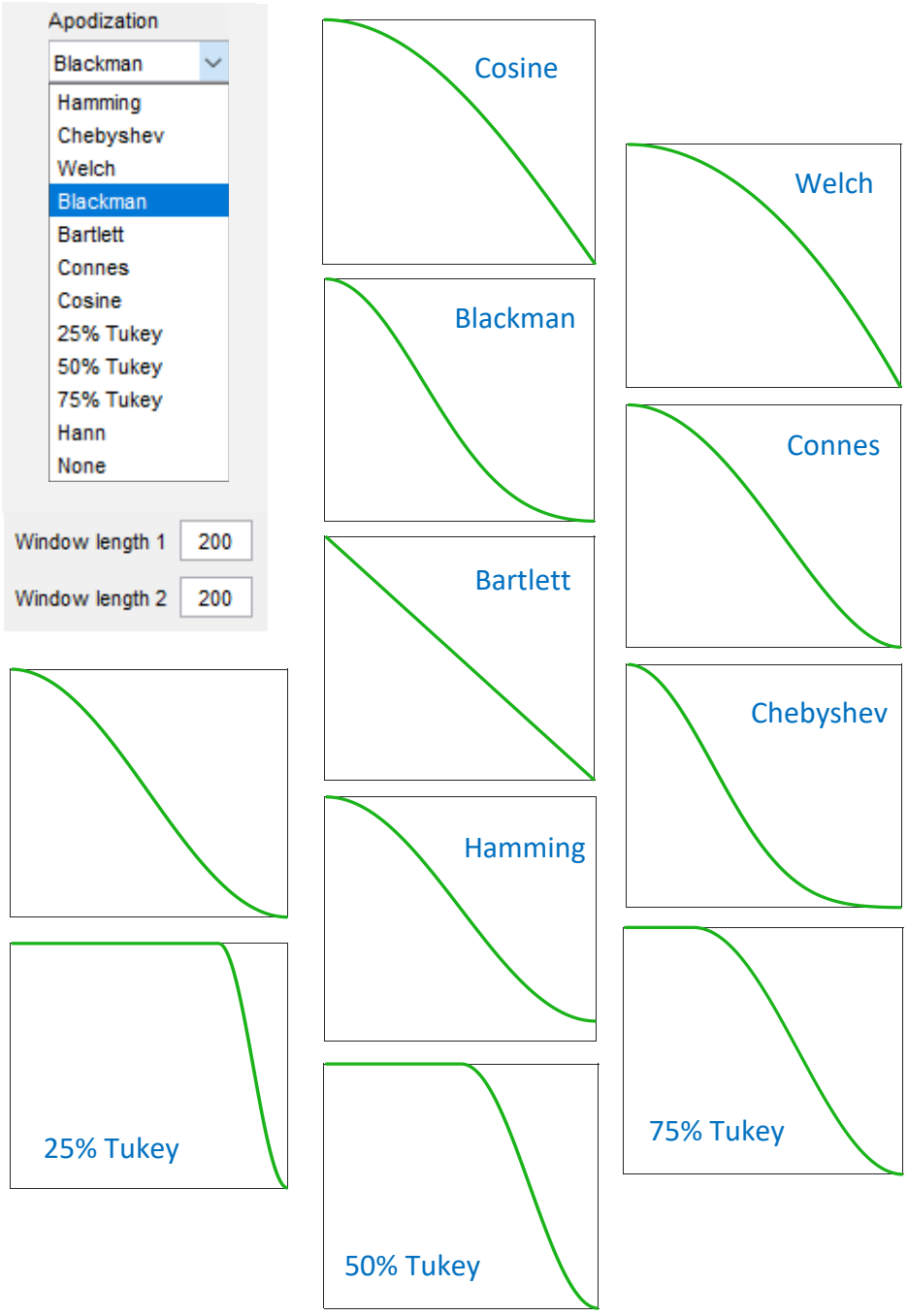


This technique allows the resolution densely populated regions of the spectra where many closely spaced peaks overlap with each other and with their tails. The values for the Lorentz-Gauss transformation are usually found in a trial-and-error approach and often not all parts of a spectrum can be equally well-resolved. The defaults given in Hyscorean ensure a strong cancellation of the signal decay.

NOTE:
While applying Lorentz-to-Gauss transformations one must make sure that the signal-to-noise ratio is large enough and that after transformation the apodization of the signal (see next section) is still working properly to avoid truncation artifacts.

3.2.2 Apodization

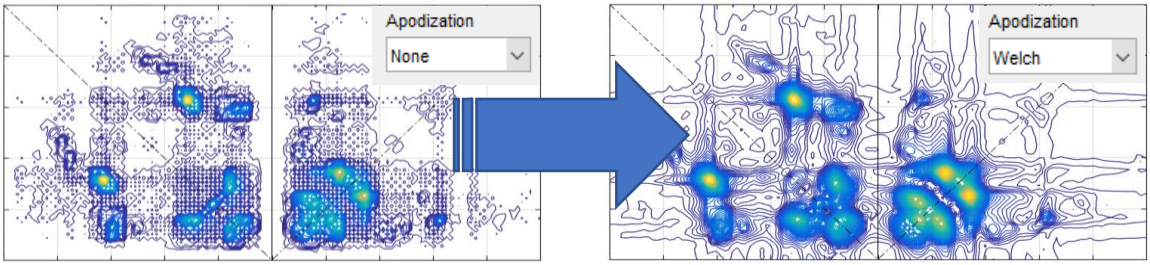
If the signal has not decayed fully to zero at the time the measurement finishes, the so-called truncation artifacts will appear in frequency-domain. This can be avoided by applying an apodization function (also called window function) which induces an artificial decay of the signal. This function suppresses the sidelobes leakage at the expense of a broadening of the lines resulting in decreased resolution. Hyscorean offers several different apodization windows which can be selected through the Apodization list box in the Processing panel:



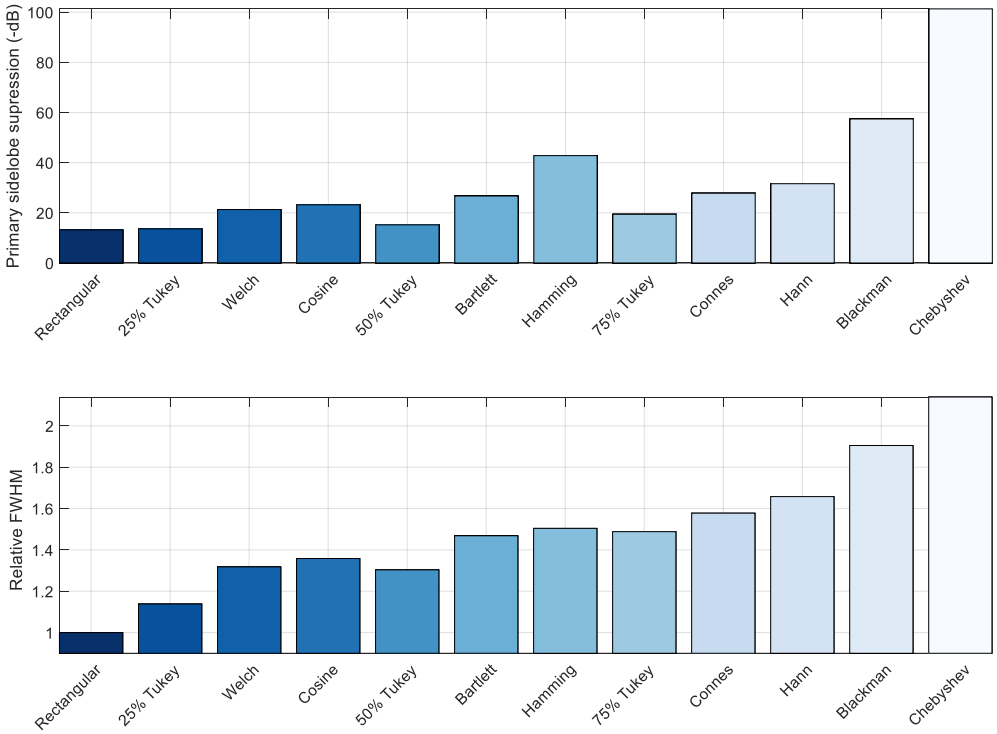
The program also allows the user to give the number of points after which the apodization window has decayed completely in each dimension as an input. This allows to generate asymmetric apodization windows in case one dimension needs to be more suppressed than the other. These values are first automatically set to the dimensions of the signal after its mounting and can be later

modified in the Window Length edit boxes near the window selection list box. If the window length along any of the dimensions exceed or are inferior to the signal dimensions, the apodization window is truncated or zero-filled, respectively. At any point apodization can be disabled and skipped by selecting the None options from the list box (not recommended).

A good measure for the capabilities of an apodization window in suppressing sidelobes is the primary sidelobe suppression [4]. The sidelobe suppression can be measured from the Fourier transform of the apodization window itself. For the non-apodised cases, the signal is truncated via a rectangular function, whose primary sidelobe is suppressed by about 13.2dB relative to the main frequency. This level introduces the serious artifacts mentioned previously, which can be cured via apodization:



Nonetheless, this is at the cost of broadening, which can also be measured from the full-width at half-maximum (FWHM) of the main frequency lobe of the Fourier transform of the apodization window [4]. The following graphs show the primary sidelobe suppression capabilities and broadening of all windows offered by Hyscorean

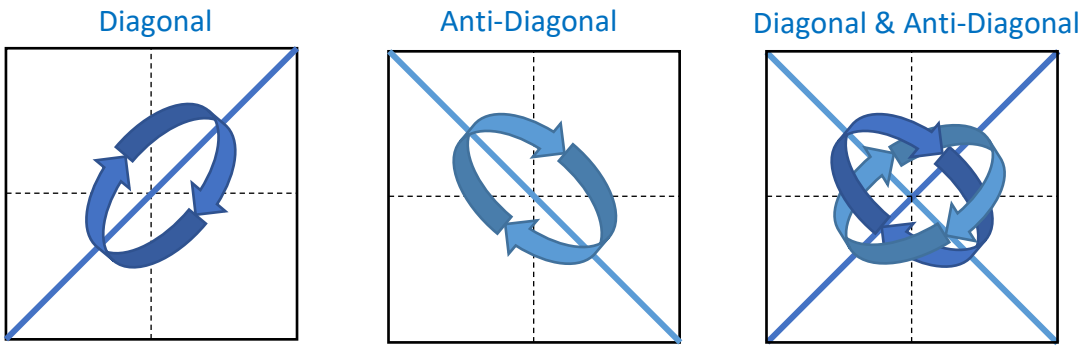


where it can be seen that strong sidelobe suppression comes at the cost of strong broadening of the signal peaks. Thus, the choice of apodization window can be made by finding a compromise between the broadening of the signal crosspeaks and the primary sidelobe suppression induced by the apodization. The user must decide where to set in this compromise and check for an optimal balance. Such an optimization (which is quickly performed) can result in strong changes in spectra with heavily populated regions.

3.2.3 Spectral Symmetrization

Thanks to the nature of the HYSCORE experiment, the obtained spectra are highly symmetric. All crosspeaks in HYSCORE spectra are symmetric along the diagonal and anti-diagonal for the strong and weak coupling quadrants, respectively.

This symmetry can be exploited to reduce artifacts arising from different sources, such as noise, background correction or non-uniform sampling. These artifacts are in nature not symmetric. Therefore, enforcing the symmetry of HYSCORE spectra via processing is a powerful tool to get rid of such "asymmetric" artifacts or at least to reduce them to noise level. This can be done by symmetrizing HYSCORE spectra along their diagonal, anti-diagonal or both.



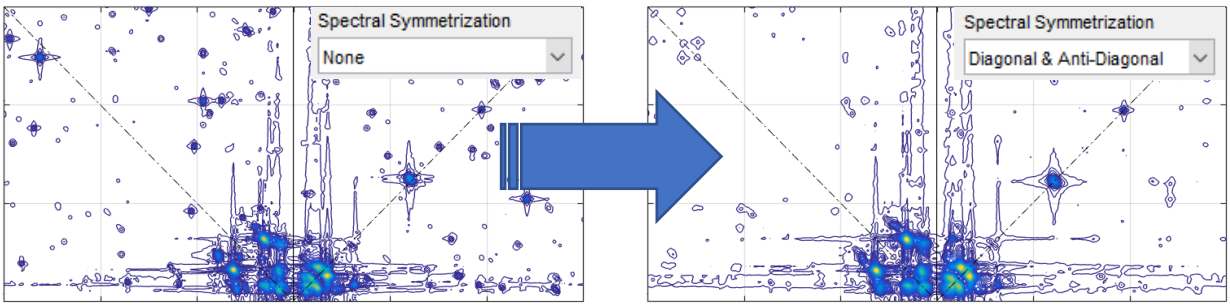
The symmetrization along the diagonal [5] is performed via the following matrix operation

$$S_{sym} = (SS^T)^{1/2}$$

and for the antidiagonal symmetrization, where Λ is a geometrical operator which flips the matrix left to right

$$S_{sym} = \Lambda\{[\Lambda(S)\Lambda(S)^T]^{1/2}\}$$

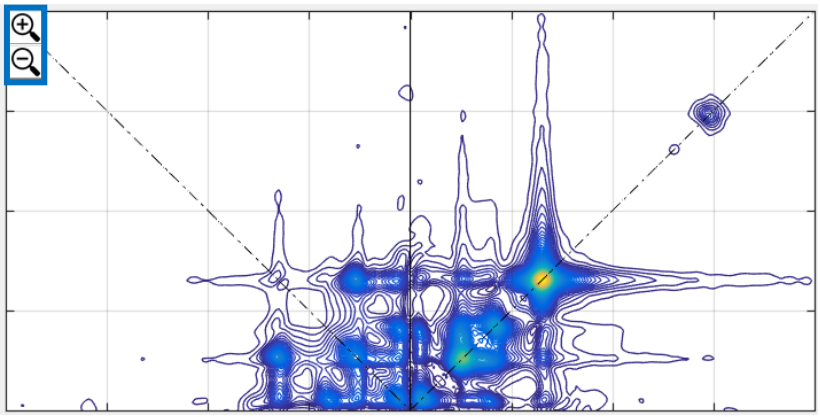
The symmetrization can be selected from the Spectral Symmetrization list box. There, diagonal and anti-diagonal symmetrization can be selected as well as the option to perform both symmetrization sequentially.



NOTE:
Symmetrization can also introduce fake crosspeaks, in case two crosspeaks have intense tails that extend a wide range of the spectrum. The user is recommended to compare the result with the normal spectrum and to judge the results.

3.3 Post-processing

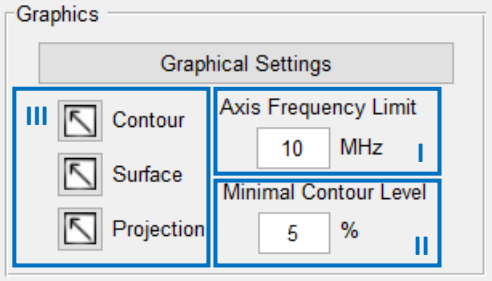
Once Hyscorean is done with the pre-processing and processing the resulting signal is Fourier transformed via Fast-Fourier transform (FFT) to the corresponding HYSCORE spectrum. This is then plotted as a magnitude contour plot (per default) on the main display. During the rendering of the spectrum, all GUI elements are disabled so that the user cannot overload the number of spectra to plot. This is denoted by the status message Rendering…



As visual guides, dashed lines are always displayed along the diagonal and anti-diagonal as well as a solid line along the first dimension's zero frequency. Additionally, the Zoom-In and Zoom-Out buttons are activated to allows the user to zoom in and out of the HYSCORE spectrum as in normal MATLAB figures.

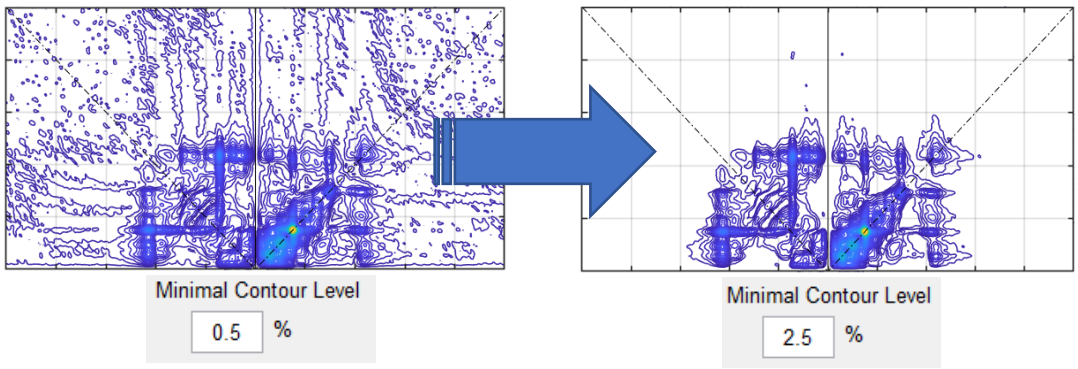
3.3.1 Graphical settings

At this point the user may change the appearance of the spectrum via the now activated Graphics panel:



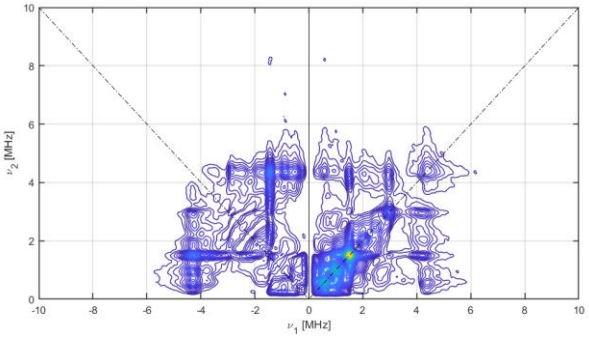
I. The Axis Frequency Limit edit box sets the maximal absolute frequency to which the spectrum is plotted (on its zoomed-out state) and given in megahertz. This value is later passed on to any other function which plots a HYSCORE spectrum of any kind.

II. Due to noise, contour plots can become crowded with many lines, obscure some features or just not be aesthetically appealing. This can be changed by setting the lowest contour level to a certain percentage of the maximal intensity. The contour levels are then equidistantly constructed between this lowest level and the maximal value of the spectrum. Anything below that minimal value is not plotted as a contour. This percentage value is given by the user in the Minimal Contour Level edit box. Setting the level to 0% will use the minimum in the spectrum.

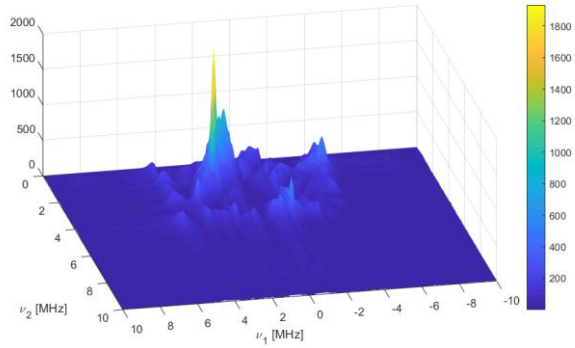


III. The spectrum can be detached into a separate new window by pressing one of the detach buttons on this group. Once detached the spectrum can be saved or copied just as normal MATLAB figures. Hyscorean spectra can be detached into three different formats: contour (just as in the GUI), surface (3D-surface with color bar) and projection (contour plot with inset displaying a skyline projection along each axis).

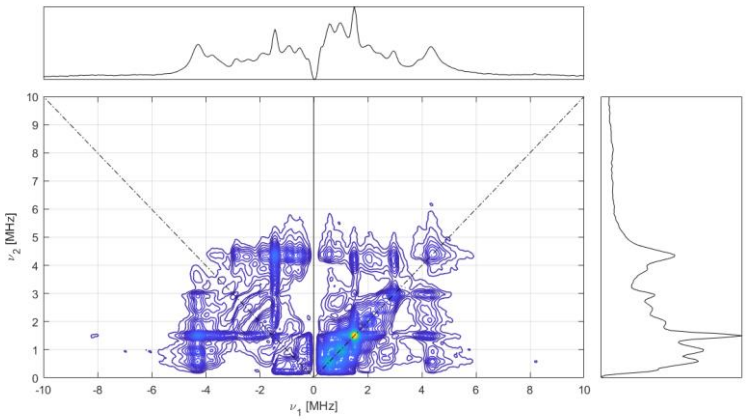
Contour



Surface

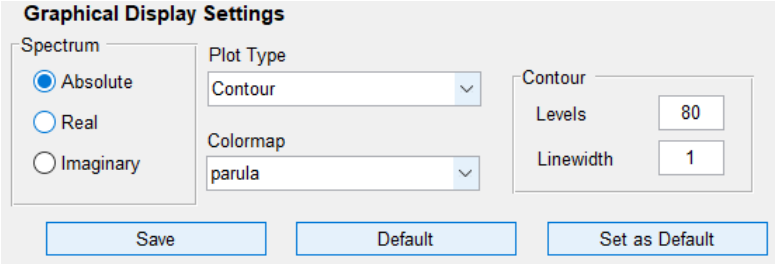


Projection



NOTE:
Any changes made in the Graphics panel will result in an automatic update of the spectrum with the updates settings. This is again notified by the status message: Rendering…. Depending on the settings and the graphics capabilities of the computer, the rendering times once the Graphical Settings window is closed can become substantial (see later).

Further graphical settings can be changed by pressing the Graphical Settings button. This will prompt a new GUI window to appear as follows:



Here different changes can be made:

- The Spectrum panel contains buttons which determine which component of the spectrum is to be plotted: real, imaginary or absolute (default) spectrum.

NOTE:
When plotting the real or imaginary components of the spectrum, the minimal contour level will be treated as 0%. This ensures the correct display of negative regions of the dispersion components in the spectrum.

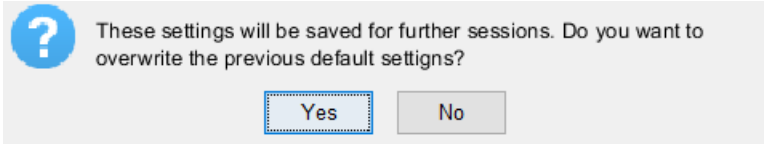
- The Plot Type box list allows the user to change the way the HYSCORE spectrum is displayed in Hyscorean: contour (default), pseudo-color or filled contour (in MATLAB: contour, pcolor and contourf respectively).

- The Colormap box list allows the user to select the colormap to be used from a list of all standard MATLAB colormaps.

- In the case of the contour or filled contour display modes the Contour panel allows to set the number of contour lines and their width.

NOTE:
Large numbers of contour lines can have huge impacts on the rendering times of the HYSCORE spectra for computers with old graphic cards or laptops.

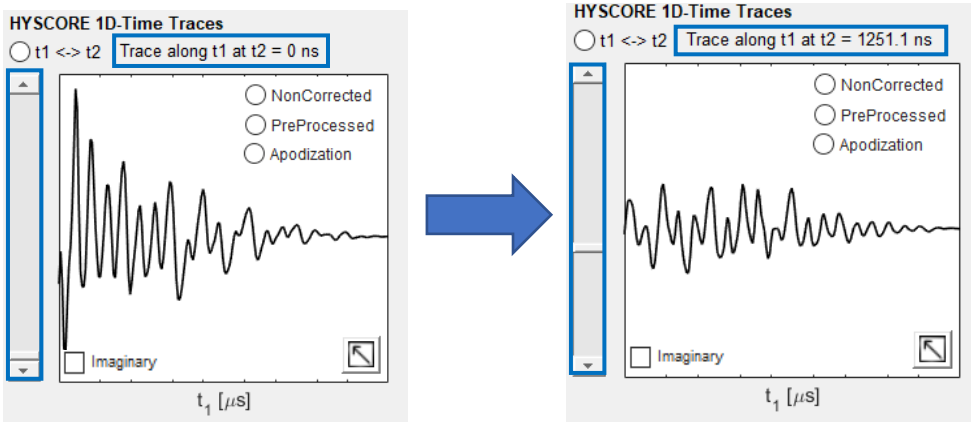
Once the desired settings have been selected, the Save button can be pressed to return to the Hyscorean main window and update the HYSCORE spectrum with the new settings. If the user desires to return the settings to their former default state, the Default button can be pressed. Should the user want to override the defaults with a new setting the Set as Default button can be pressed. This will prompt a confirmation window before the defaults are overridden. Selecting Yes then save the user-defined defaults for all further sessions.



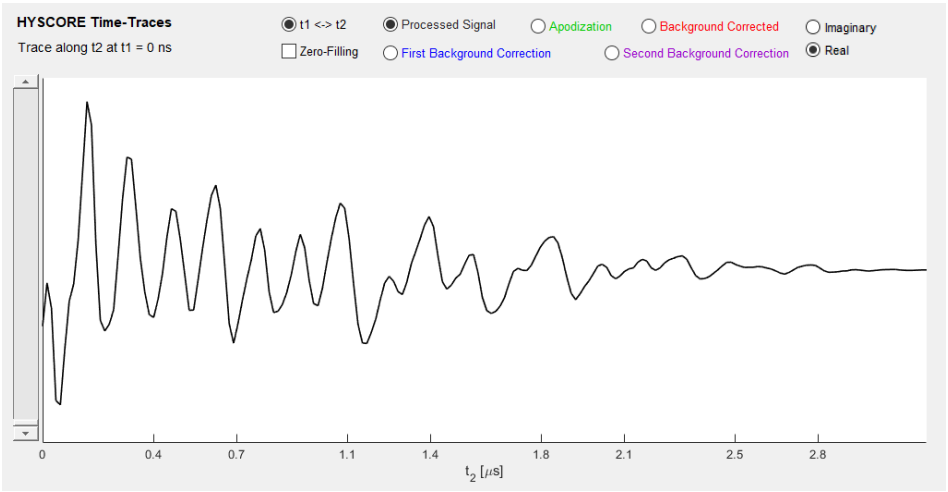
3.3.2 Time-domain signal monitoring

During the processing in Hyscorean, the time-domain signal undergoes several procedures which change it. Of course, ultimately the HYSCORE spectrum contains most of the information the spectroscopist is interested from signal. Nonetheless, monitoring the state of the time-signal during its processing is a good routine to adapt, since it allows a good check of whether information may be lost before obtaining the spectrum.

After the processing is finished, the processed signal is displayed as a trace in the smaller display next to the HYSCORE spectrum. By default, the traces are shown along the first dimension. However, by pressing the t1<->t2 button will invert the display between both dimensions. In both cases the other dimension can be swept by means of the slider next to the display. For complex-valued signals the user can switch the display between the real and imaginary components via the Imaginary checkbox.



For better display and options, the time-domain signal can be monitored from a detached window by pressing the detach button on the display. This will prompt a new GUI to open with a similar structure to the one in the main window:



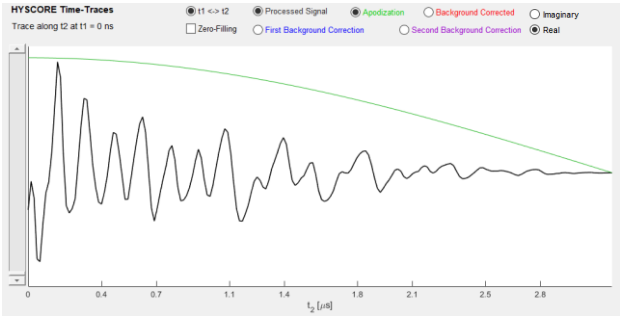
This new window works basically as described above with the addition that the signal at different states can be displayed/monitored by activating them via the highlighted buttons:

- Processed signal

  The fully processed signal from which the HYSCORE spectrum is obtained. The same signal that is automatically displayed on the main window display. The appended zeros by zero-filling can be displayed by activating the corresponding Zero-Filling check box.

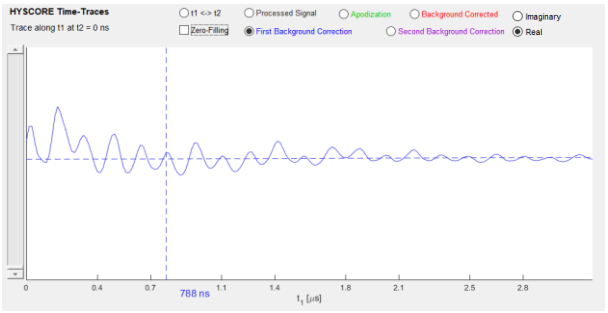
❖ Apodization

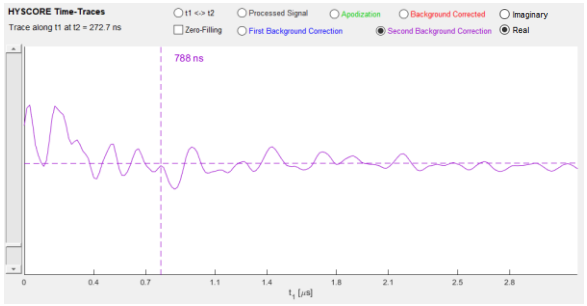
The apodization window function employed on the pre-processed signal at the current dimension is displayed as a green line (can also be displayed on the main window).



❖ First/Second Background Correction

Displays the signal before the first/second background correction overlaid with the corresponding fitted background (dashed line). A vertical dashed line marks the point at which the background starts to be fitted with and additional tag indicating the exact time.

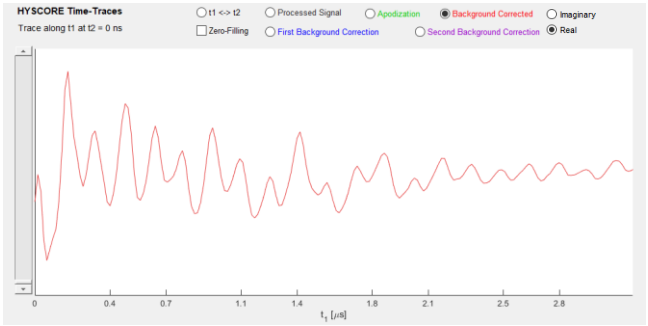




NOTE:
Due to the background correction being performed individually for each dimension, trying to change the dimension via the t1<->t2 button will not have any effect on these displays.

❖ Background corrected

The signal right after the pre-processing displayed as a red line (can also be displayed in the main window).



❖ Real/Imaginary

Switches between the real and imaginary components of the signals currently being displayed. The fitted backgrounds also are adapted to the component of choice.

All these displays can be enabled and disabled at the same time so that the user may compare the signal at different stages of the processing, e.g. check the fitted background during the background corrections. This detached window can be left open and should the user process the signal with new settings, Hyscorean will recognize that this window is open and update it automatically without the need to close and open it again.

For non-uniform sampled signals, all displayed signals except the processed signal will be automatically displayed as single points instead of lines. This allows to better visualize sparsely sampled signals and to better compare the experimental data to the reconstructed signal.

NOTE:
This signal monitoring window sets MATLAB under a uiwait state. This means that as long as the window is open, MATLAB will not be able to open any files. To open files, the signal monitoring window needs to be closed.

3.3.3 Blind spots simulator

In a spectrum of a signal obtained from a HYSCORE experiment:

$$\frac{\pi}{2} - \tau - \frac{\pi}{2} - t_1 - \pi - t_2 - \frac{\pi}{2} - \tau - \text{echo}$$

The intensity of the crosspeaks appearing at $(\pm\nu_1, \pm\nu_2)$ are given by [6]:

$$I(\nu_1, \nu_2) = \lambda z^2 \sin\left(\frac{2\pi\nu_1\tau}{2}\right) \sin\left(\frac{2\pi\nu_2\tau}{2}\right)$$

where λ is the modulation depth and z^2 a contribution resulting from different mixing of the nuclear Zeeman states in the α and β electron spin manifolds. From this equation it is apparent that the factor

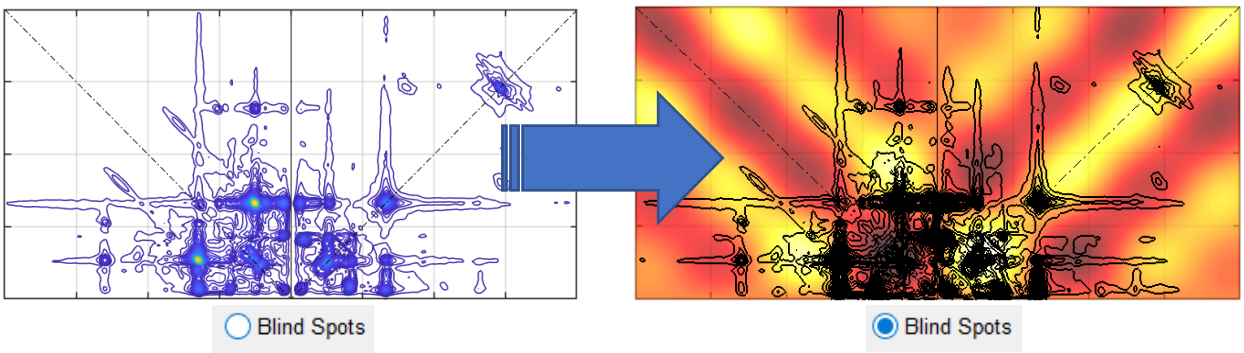
$$b(\nu_1, \nu_2, \tau) = \sin\left(\frac{2\pi\nu_1\tau}{2}\right) \sin\left(\frac{2\pi\nu_2\tau}{2}\right)$$

Introduces a τ-dependent blind spot behavior of the HYSCORE spectrum. These blind spots can be highly damaging and dangerous for the interpretation of HYSCORE spectra. This is a reason why HYSCORE experiments should be recorded at different τ-values.

In Hyscorean, the blind spots corresponding to the currently selected τ-values in the Tau-Selection list box can be simulated via

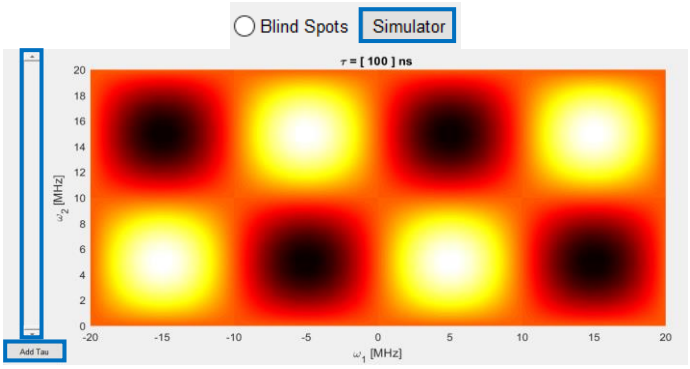
$$B(\nu_1, \nu_2) = \sum_{k=1}^{N} b(\nu_1, \nu_2, \tau_k)$$

where N is the number of combined τ-values. This blind spot map can be superimposed onto the displayed spectrum to see which parts of the spectrum may be suppressed by such blind spots. This can be enabled and disabled from the Blind Spots button.

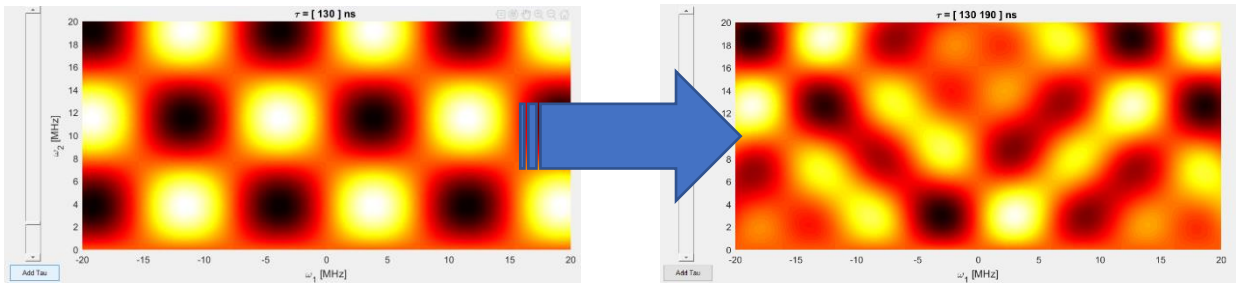


The blind spots are identified as the darker spots in the superimposed map, whereas the brighter spots represent the zones of maximal intensity.

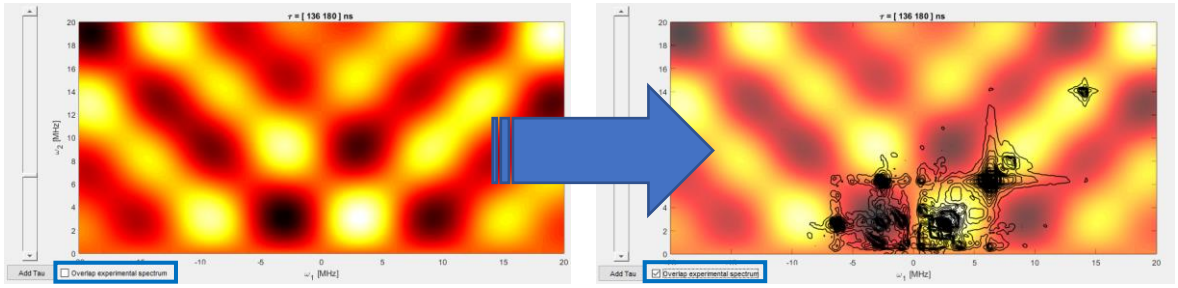
There is also the possibility to simulate the blind spots from a constructed sequence of τ-values. This is helpful to interactively find appropriate τ-values for an experiment. The simulator can be opened by pressing the Simulator button.



The current τ-values are given in the top of the figure. The current τ-value can be changed by moving the slider. Once a desired τ-value is found it can be added to the list by pressing the Add Tau button, which will fix the τ-value and add a next one to be changed:



The current experimental spectrum can be overlaid upon the simulated blind spots via the Overlap experimental spectrum checkbox.



NOTE:
The slider can adopt τ-values between 100ns and 350ns. Also the response time for the simulator can be longer than expected for computers with low graphics capabilities.

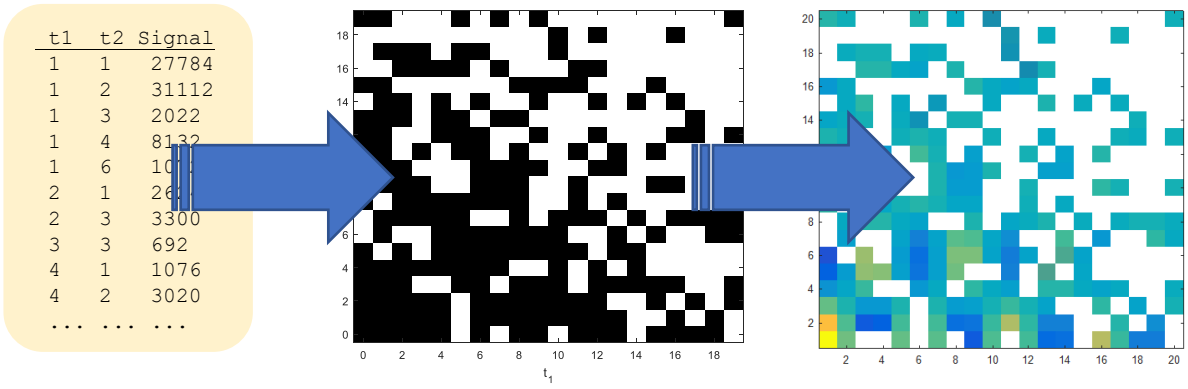
4. NUS HYSCORE Processing

The ability to mount and process non-uniform sampled HYSCORE data is one of the main novelties offered by Hyscorean. Non-uniform sampling (NUS) not only allows the measurements to be performed in several fractions of the usual measurement times but also the possibility to even gain in resolution and sensitivity.

The following sections will cover those parts of the processing which are unique to NUS data. The rest of the processing steps and protocols are identical to uniform-sampled data and, therefore, the user is referred to previous section for a detailed description of the other processes.

4.1 Mounting NUS HYSCORE data

The main change in the mounting procedures is the fact that the input data does not contain all points of a Nyquist grid. Therefore, the mounting now must extract the NUS grid from the measured t1 and t2 timings. From the NUS grid and the timings, the signal can be piecewise constructed from the data obtained from the measurements to obtain the NUS HYSCORE signal.



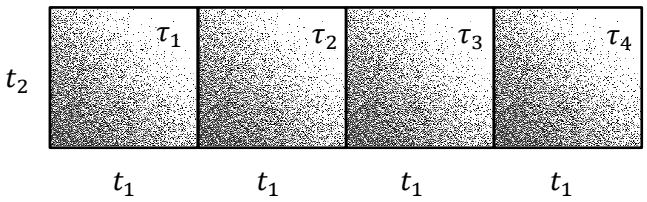
For NUS data, Hyscorean does allow the same input data formats as for uniform sampled data (i.e. DTA, DSC, mat, txt, ...). Nonetheless, the way the mounting is performed is slightly different for each of these formats and are, thus, briefly described in the following.

4.1.1 Mounting NUS Bruker BES3T data

Hyscorean automatically recognizes HYSCORE data obtained from commercial Bruker spectrometer data in BES3T data formats (DSC and DTA). This data can be obtained as described in section 9. NUS measurements on Bruker XEPR spectrometers and then loaded into Hyscorean as usual via the Load button.

The software recognizes that it is NUS data due to the DSC or DTA files containing only one-dimensional data and not two-dimensional data as is the case for uniform-sampled BES3T data. The t1- and t2-timings are then extracted from the abscissa according to the encoding described in 9.1. and the effective dwell times for each dimension are determined.

The empty NUS grid is then constructed from the effective dwell time and the maximal measured timing along each dimension. The sampled NUS grid points are then filled according to the t1/t2-timings combinations present in the abscissa of the DSC or DTA file.



The measured signal points are then extracted from the ordinate an arranged in a fashion similar to how the data was mounted for the uniform-sampled case (see 2.4.1.) where now all non-measured points are set to NaN-values. The τ-values employed during the experiments are extracted from the DSC or DTA file descriptors. It is important that the τ-values are defined in the CMTS parameter as described in section 9.1.

Once the NUS data has been mounted, all different τ-values and their possible combinations will be updated as usual for the user to choose from the Tau-selection list box.

4.1.2 Mounting NUS ASCII formatted data

The program takes the same ASCII data input files as in section 2.4.3. and checks whether the dwell time between all measured points is the same. If not, the data is identified as non-uniform and the NUS grid is constructed as in the previous section.

The empty NUS grid is constructed from the effective dwell time and the maximal measured timing along each dimension and the sampled NUS grid points are filled according to the t1/t2-timings combinations present in the corresponding ASCII columns. As in section 2.4.3. the program identifies the different τ -values employed for the measurement and then constructs the NUS signals from the ASCII columns with the help of the NUS grid.

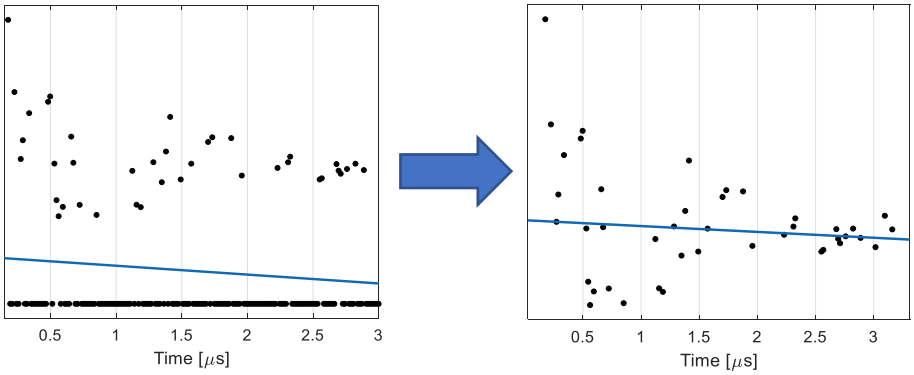
4.2 NUS HYSCORE Pre-Processing

The main difference between “standard” HYSCORE processing and NUS HYSCORE processing is the fact that an additional reconstruction step must be incorporated. Once the NUS signal is reconstructed all remaining steps for the processing remain unchanged. Nonetheless, small differences arise during other steps of the pre-processing and will be handled in the following sections.

4.2.1 NUS background correction

Once the NUS data has been mounted, the multiple signals measured at different τ-values are combined according to the user input. At this point the signal is zero-augmented and must be background corrected. Due to these artificially inserted zeros, a try to fit the background from the data would be ill-posed leading to wrong background fits.

Therefore, to avoid this, the zeros introduced by zero-augmentation are converted to NaN values and then background corrected. MATLAB fit functions are programed to ignore any NaN values from the data being fitted. Thus, only the measured signal points are fitted and then background corrected.

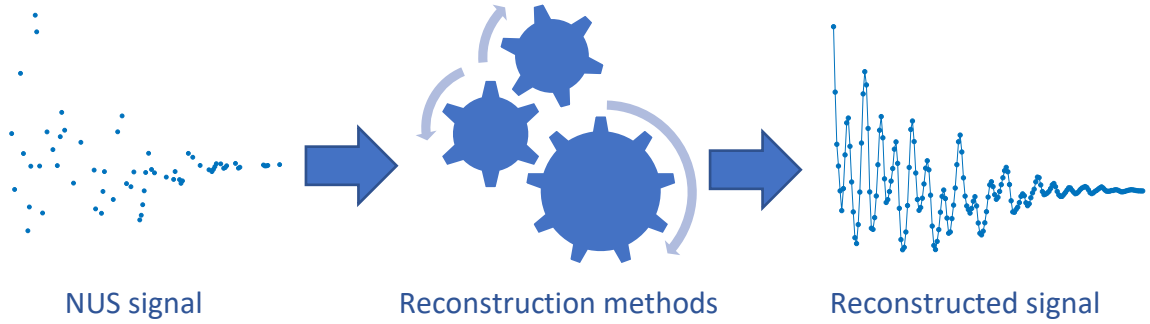


Once the background has been corrected, the NaN values in the signal are returned to zeros to give the background-corrected and zero-augmented NUS signal.

NOTE:
As the number of points in the traces get lower, the fitted backgrounds may start to behave strange or even look unphysical. This is, however, not an issue since the fit between the few points is actually good. For the non-measured points, it does not matter how the background is fitted since they are set to NaN and remain so even when a strange background is subtracted from them.

4.3 NUS signal reconstruction

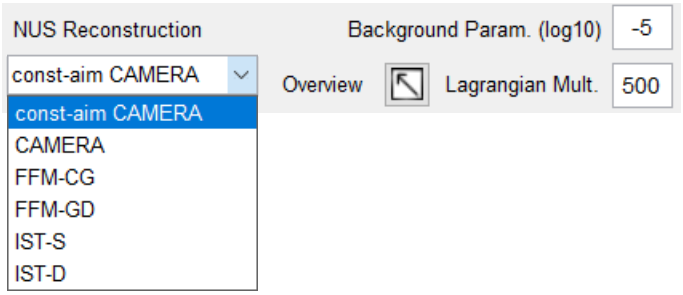
In this new step, the NUS signal must be reconstructed:



Due to the non-uniform nature of the measurement, the Fourier transform of NUS signals no longer corresponds to an expansion in an orthonormal Fourier basis set and therefore is not a transform anymore in a strict sense [7]. However, discrete Fourier transform (DFT) of the zero-augmented NUS signal still is possible and leads to a spectrum which in Hyscorean is referred to as the nuDFT (non-uniform DFT) spectrum.

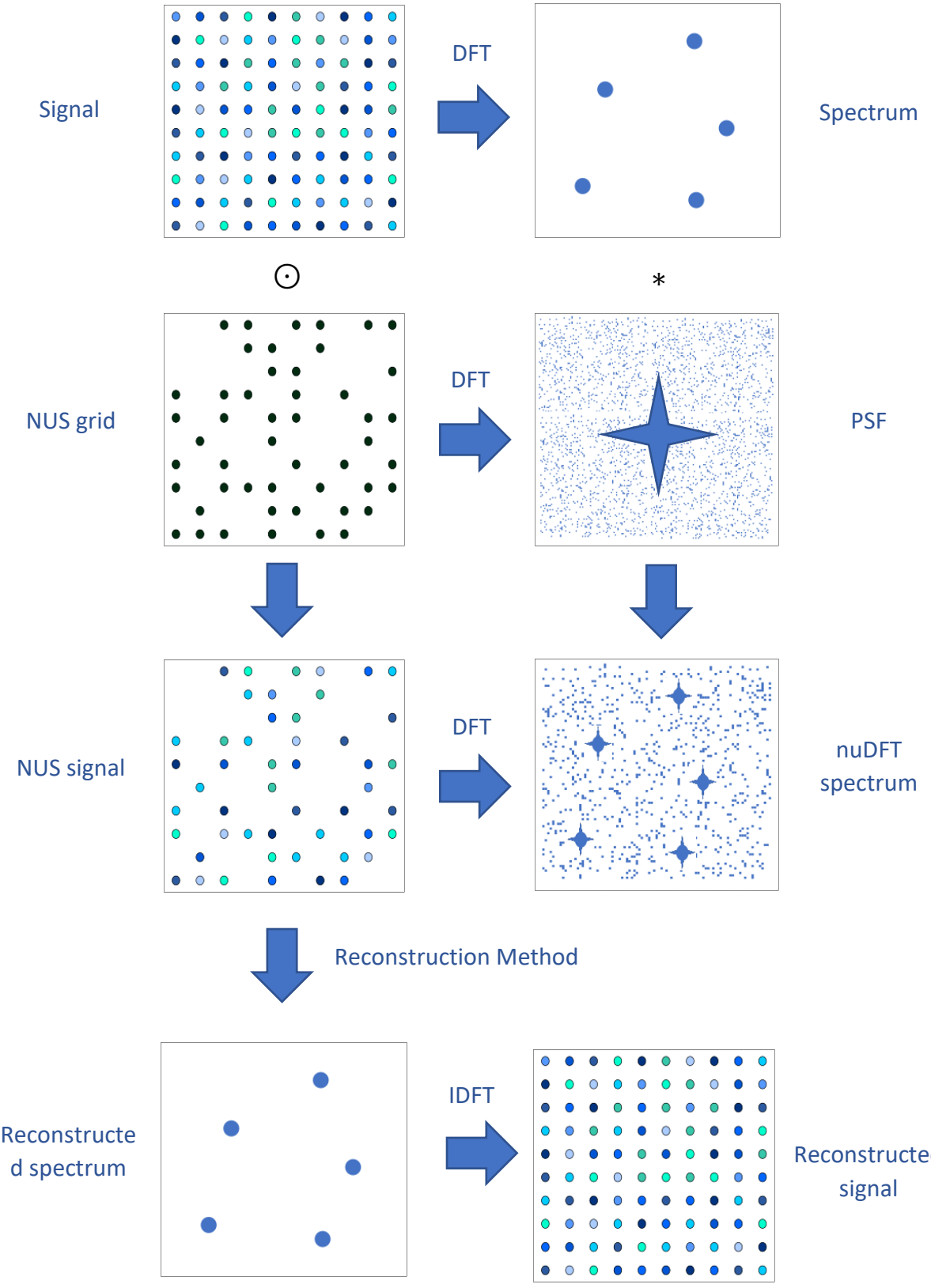
The nuDFT spectrum is the convolution of the uniformly sampled spectrum with the point-spread function (PSF) of the sampling grid, which can be obtained by Fourier transformation of the sampling grid. The reconstruction methods aim to deconvolve the PSF from the spectrum via regularization. This procedure is equivalent to reconstructing the missing points in the NUS signal.

During the data mounting Hyscorean will recognize if the data are NUS and will activate the NUS Reconstruction section in the Pre-processing panel:



The user can select a reconstruction method from the list. These can be categorized into two reconstruction families: maximum entropy (maxEnt) and iterative soft thresholding (IST) reconstruction. Depending on whether the selected method requires any additional input parameters, the corresponding elements in the GUI will be activated or deactivated

The following sections will provide a brief description on the reconstruction methods available on Hyscorean and their way of action.


Signal
DFT
Spectrum
⊙
*
NUS grid
DFT
PSF
NUS signal
DFT
nuDFT spectrum
Reconstruction Method
Reconstructed spectrum
IDFT
Reconstructed signal

4.3.1 Iterative Soft-Thresholding (IST) reconstruction

An ℓ_1-based regularization method based on the compressed-sensing principle, that through optimization, the sparsity of a signal can be exploited to recover it from fewer samples than required by the Nyquist theorem [8]. This method regularizes a spectrum $\boldsymbol{X}$ that minimizes the functional

$$\boldsymbol{X}_{\mathrm{IST}} = \underset{\boldsymbol{X}}{\operatorname{argmin}}\{ \|\boldsymbol{b} - \boldsymbol{DFX}\|_2 - \|\boldsymbol{X}\|_1\}$$

where $\boldsymbol{b}$ is the measured experimental signal, $\boldsymbol{D}$ is the NUS grid and $\boldsymbol{F}$ is the inverse Fourier transform kernel. This functional is solved by iterative soft-thresholding, which iteratively updates the solution $\boldsymbol{X}$ with all values in $\boldsymbol{F}^{-1}\boldsymbol{b}$ (i.e. the nuDFT spectrum) larger than the threshold value η.

I. The threshold is set to $\eta = \tau \left|\max(\boldsymbol{F}^{-1}\boldsymbol{b})\right|$, where τ is the damping factor.
II. The solution $\boldsymbol{X}$ is updated with all values in $\boldsymbol{F}^{-1}\boldsymbol{b}$ larger than the threshold η.
III. The input signal $\boldsymbol{b}$ is updated by subtracting the term $\boldsymbol{DFX}$ or $(\boldsymbol{1} - \boldsymbol{D})\boldsymbol{FX}$.
IV. The updated spectrum $\boldsymbol{F}^{-1}\boldsymbol{b'}$ will contain all artifacts and less meaningful peaks.
V. The threshold is updated to $\eta = \tau \left|\max(\boldsymbol{F}^{-1}\boldsymbol{b}')\right|$ and the next iteration is started.
VI. After sufficient iterations the solution $\boldsymbol{X}$ will contain meaningful peaks but no artifacts.

Two variants of this algorithm have been developed [8]:

- IST-D
  In Drori's IST (IST-D) the term $\boldsymbol{DFX}$ is subtracted from $\boldsymbol{b}$ and hence the measured points in $\boldsymbol{b}$ are updated and non-measured points in $\boldsymbol{b}$ are left to zero. This allows for the algorithm to find a balance between data agreement and sparsity in the solution.

- IST-S
  In Stern's IST (IST-S) the term $(\boldsymbol{1} - \boldsymbol{D})\boldsymbol{FX}$ is subtracted from $\boldsymbol{b}$ and hence non-measured points in $\boldsymbol{b}$ are updated and measured points in $\boldsymbol{b}$ are left unchanged. This enforces a strict accordance between the reconstruction and the measured data.

Due to the thresholding nature of the IST methods, the intensity response of the reconstructed spectrum is linear with respect to that of the experimental data. In Hyscorean both IST-D and IST-S can be chosen by the user as reconstruction methods. In both cases the damping factor τ is set to 0.99 and updated after each iteration k according to

$$\tau_k = \tau_0 \frac{N_{iter} - k}{N_{iter}}$$

which enhances the convergence speed of the algorithm. Nonetheless, the damping factor can be later validated via Hyscorean's validation module (see section 6).

NOTE:
Due to the thresholding nature of these algorithms, in cases with strongly pronounced PSF's (due to e.g. low sampling densities or low randomness) the sampling artifacts may be so pronounced that these lie at the level of true peaks and be incorporated into the solution via the thresholding. In such cases, IST-S and IST-D may not be able to recover any more signal and the use of maxEnt methods are recommended.

4.3.2 Maximum Entropy (maxEnt) reconstruction

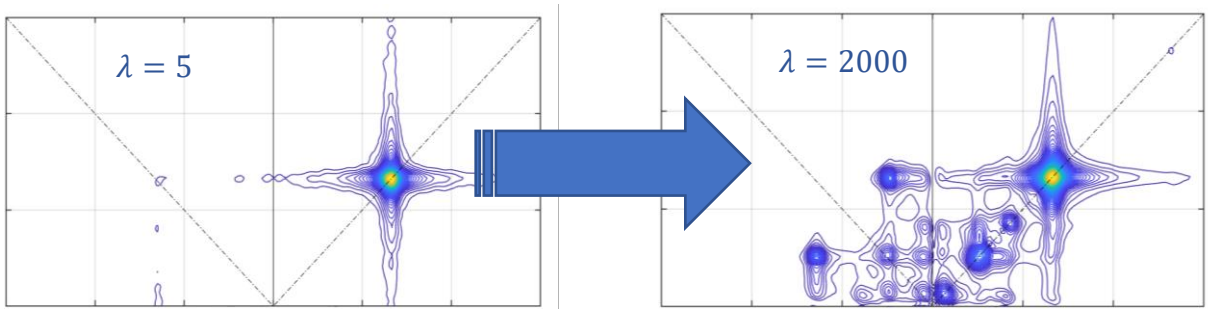
Maximum entropy (maxEnt) methods aim to find the spectrum that has minimal statistical information content (i.e. maximum entropy), while still maintaining agreement with the measured data. This is achieved by regularization of the reconstructed spectrum $\boldsymbol{X}$ via minimization (by discrepancy principle) of functionals of the form

$$\boldsymbol{X}_{\text{maxEnt}} = \underset{\boldsymbol{X}}{\text{argmin}}\{\lambda\|\boldsymbol{b} - \boldsymbol{DFX}\|_2 - S(\boldsymbol{X})\} \quad \text{under} \quad \|\boldsymbol{b} - \boldsymbol{DFX}\|_2 \leq \varepsilon$$

where again $\boldsymbol{b}$ is the measured experimental NUS signal, $\boldsymbol{D}$ is the NUS grid, $\boldsymbol{F}$ is the inverse Fourier transform kernel and ε is the noise level in the signal. The Lagrange multiplier λ controls the balance between the data agreement imposed by the first term in the functional and the entropy penalty. Therefore, depending on the choice of λ the intensity response of MaxEnt methods varies:

- Small λ Non-linear intensity response of the reconstructed spectrum.
- Large λ Linear intensity response of the reconstructed spectrum.

The effects of non-linearity can be observed in the following example:



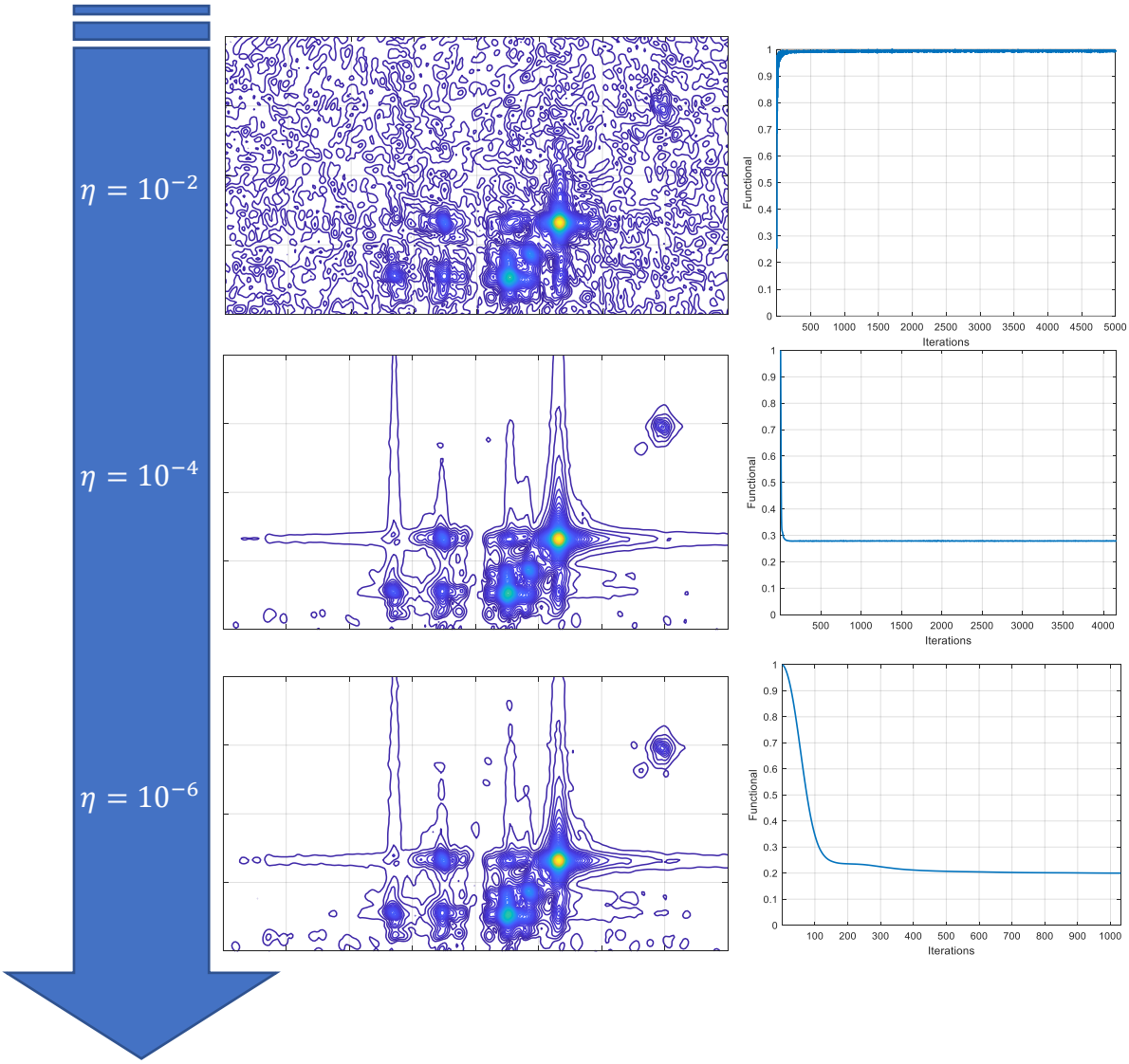
As the non-linearity increases the relative scaling of the peaks in the reconstructed spectrum becomes more drastic so that even true peaks are removed from the spectrum leaving only the most intensive ones. This does however not mean that non-linearity is to be avoided since it allows MaxEnt to recover the true spectrum from the noise and sampling artifacts (see later).

The term $S(\boldsymbol{X})$ represents the spectral entropy and in Hyscorean this is implemented with the Hoch-Hore entropy [9]

$$S(\boldsymbol{X}) = -\sum_{i=1}^{N} |X_i| \log\left[\frac{|X_i|}{\delta} + \sqrt{1 + \left(\frac{|X_i|}{\delta}\right)^2}\right] - \sqrt{|X_i|^2 + 4\delta^2}$$

which is monotonous in contrast to the Shannon and Skilling entropies as well as insensitive to the phase of the signal. The parameter δ is the so-called background parameter, which to a large extent determines the threshold at which the non-linear effects become significant. It can also be described to control the curvature of the maxEnt functionals, leading to smoother/rougher reconstructions.

- Small η Smooth, slow converging functional (i.e. long computation times).
  Smoother and generally more accurate reconstructions
- Large η Rough functional (i.e. short computational times) but with divergence risk
  Artifacts start to populate the reconstruction.



4.3.2.1 *Fast-forward maxEnt (FFM) reconstruction*

This maxEnt method represents the limiting case of the previous functional when $\lambda \to \infty$, which enforces a strict accordance with the measured data [10]. Thus, experimental measured points are directly set into the reconstruction and only the missing points are reconstructed. These are reconstructed via entropy gradient methods which in Hyscorean are called:

- FFM-CG
  Employs the conjugate gradient method to reconstruct the missing points.

- FFM-GD
  Employs the gradient descent method to reconstruct the missing points.

One advantage of this method is the absence of the Lagrange multiplier which the reduces to parameter space to just the background parameter η for the entropy. In Hyscorean this can be given as a logarithmic value in the Background Param. (log10) edit box. However, the FFM methods are known to over-fit noise at low SNR signals.

4.3.2.2 Convex Accelerated MaxEnt Reconstruction Algorithm (CAMERA)

CAMERA is a new approach to spectral reconstruction that exhibits fast, tunable convergence in both constant-aim and constant-lambda mode [9]. Hyscorean allows the operation of CAMERA on both modes:

- Constant-aim CAMERA

  On this mode CAMERA employs Nesterov's accelerated first-order convex optimization method where the Lagrange multiplier is iteratively updated according to

  $$\lambda = \max\{0, L\varepsilon^{-1}\|\boldsymbol{b} - \boldsymbol{DFX}_{t-1} + L^{-1}\boldsymbol{DF}\nabla S(\boldsymbol{X}_{t-1})\|_2 - L\}$$

  where L is a Lipschitz constant which depends on the background parameter η. Also, the reconstruction at iteration t is updated taking into consideration the solutions in the previous iterations in order to generate a momentum which allows the algorithm to reach the optimum solution faster than other optimization methods:

  $$\boldsymbol{X}_t = \boldsymbol{Y}_t + \left(\frac{t-1}{t+2}\right)(\boldsymbol{Y}_t - \boldsymbol{Y}_{t-1})$$

  where $\boldsymbol{Y}_{t-1}$ is the proximal gradient mapping step given by

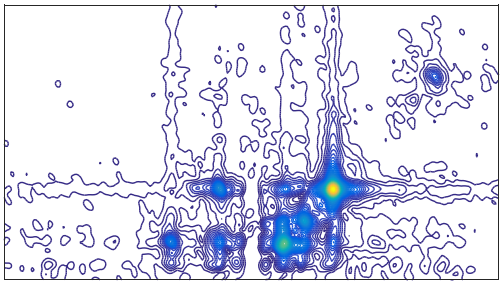
  $$\boldsymbol{Y}_t = (\boldsymbol{I} + \lambda L^{-1}\boldsymbol{D}^T\boldsymbol{D})^{-1}(\boldsymbol{X}_{t-1} + \lambda L^{-1}\boldsymbol{X}_0 - L^{-1}\nabla S(\boldsymbol{X}_{t-1}))$$

  and the Lipschitz constant is also updated iteratively according to $L \leftarrow \min\{2L, (2\eta)^{-1}\}$.

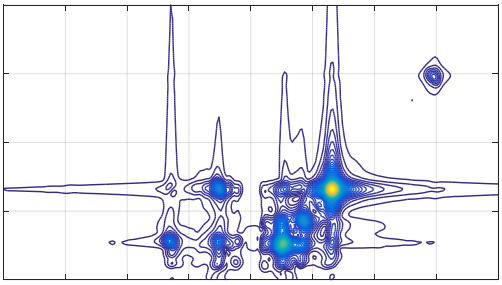
- Constant-λ CAMERA

  On this mode CAMERA does not update the Lagrange multiplier as before but fixes it to a user-defined value. This mode allows the user to control the non-linearity of the maxEnt reconstruction manually and to impose more de-noising of the spectrum.

Constant-aim CAMERA performs in a similar way as the FFM methods (as well as IST) and reconstructs the spectrum as well as in the uniform sampled case for moderate or good PSF (i.e. moderate sampling densities or good SNR). Nonetheless, constant-λ CAMERA allows the user to impose even further regularization onto the spectrum resulting in a de-noising effect. Also, in cases when the other reconstruction methods may fail, constant-λ CAMERA can still reconstruct the true spectrum by proper choice of the combination of background parameter and Lagrange multiplier.



constant-aim CAMERA
$\eta = 10^{-6}$ & $\lambda \approx 700'000 - 1'700'000$



constant-λ CAMERA
$\eta = 10^{-6}$ & $\lambda = 500$

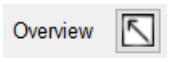
Constant-aim CAMERA tends to set the Lagrange multiplier to large values setting the reconstruction in a highly linear regime (meaning that even the noise level is reconstructed). However, constant-λ

CAMERA can set the algorithm into a highly non-linear regime which is necessary if the artifacts in the nuDFT spectrum are too large.

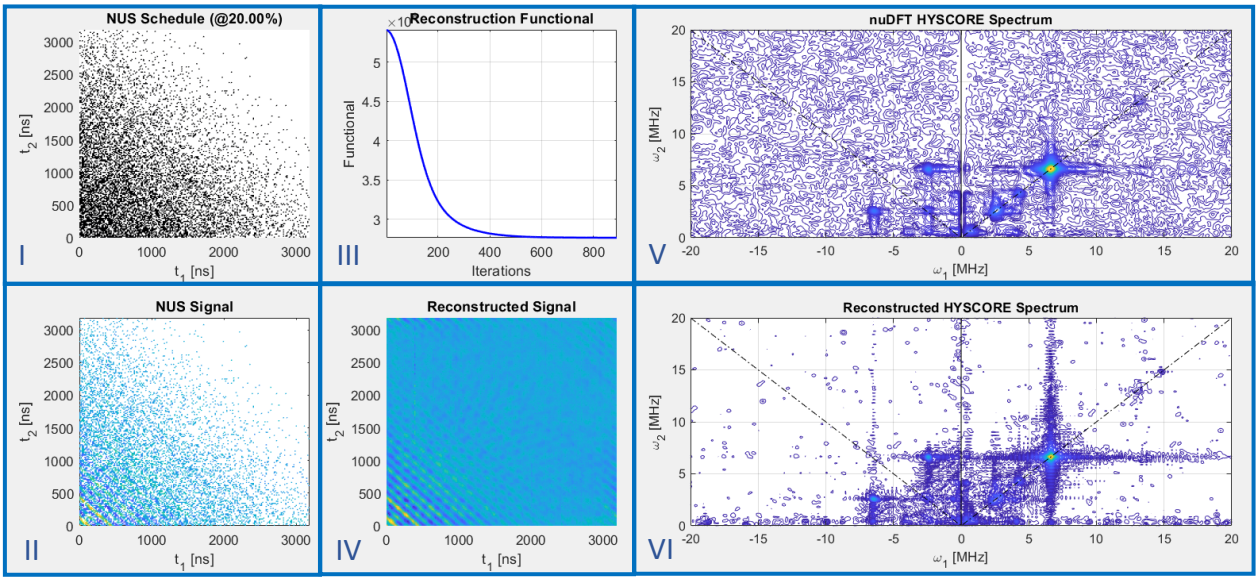
4.4 NUS HYSCORE post-processing

Once the NUS signal has been reconstructed, it undergoes all the processing steps remaining as normal. As in the rest of the steps of the pre-processing, pressing the Process button again without changing any of the NUS reconstruction settings will result in the reconstruction being skipped employing the pre-processed reconstructed signal already available.

An overview of the results of the reconstruction can be displayed by pressing the Overview detach button



which opens a new MATLAB figure with the following displays:



I. The NUS grid employed for the measurement. The black points indicate measured points.
II. The experimental NUS HYSCORE signal after background correction.
III. The functional values of the reconstruction methods at each iteration of the reconstruction.
IV. The reconstructed HYSCORE signal.
V. The nuDFT HYSCORE spectrum of the signal prior to reconstruction.
VI. The reconstructed HYSCORE spectrum right before the rest of the processing.

A comparison of the nuDFT and reconstructed HYSCORE spectra is already a good control point to check the quality of the reconstruction, since most of the true peaks are usually already visible from the nuDFT spectrum and may have been removed, e.g. due to strong non-linearity of maxEnt reconstruction.

5. Saving Hyscorean results

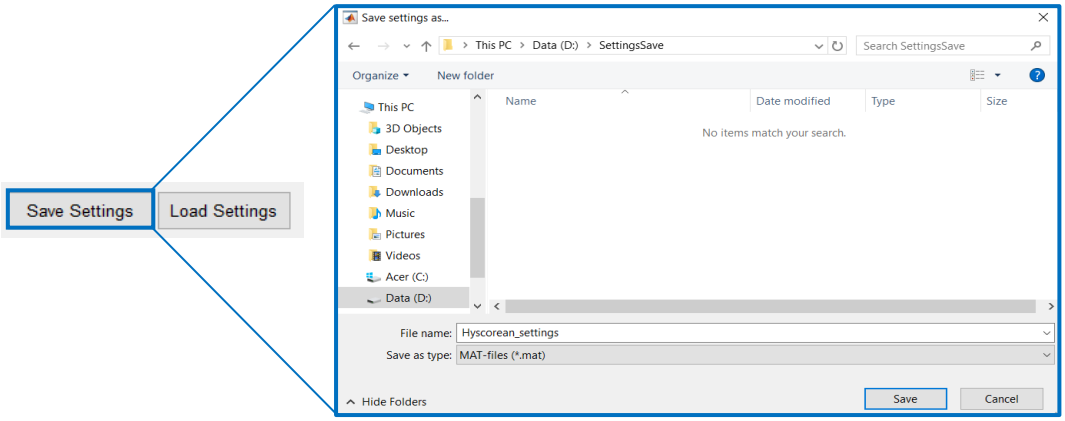
From the help provided in the previous sections the user can process HYSCORE data into a well-processed spectrum. Now the question comes how to save/export the data outside of Hyscorean. A straightforward way would be to detach the current spectrum (as described in 3.3.1) and save it via the MATLAB save interface.

Nonetheless, the processing of spectra is a process which requires many parameters and quickly generates many different spectra of similar interest. It is rather easy to get lost in similarly looking spectra and settings. Therefore, Hyscorean offers a saving system which not only allows for the save export of data but an organized book-keeping system of all processed spectra as well.

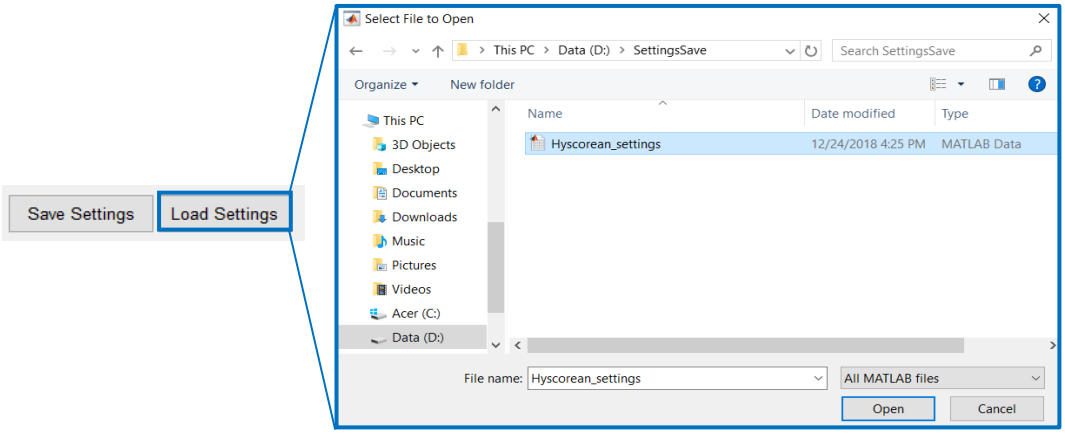
This section will describe the setup, description and operation of this saving system.

5.1 Saving/Loading settings

Hyscorean features a wide range of processing opportunities. This comes, however, at the cost of a large quantity of parameters and settings to be set. The user can save the current settings (no graphical settings) manually via the Save Settings button. The button will prompt a new OS window which will ask the user to select a folder and filename to save the settings to.



All settings (edit and list boxes in the GUI) will then be saved into a .mat file of the given name. These settings can then be loaded back by the Load Settings button, which will override all current settings by those in the loaded file.

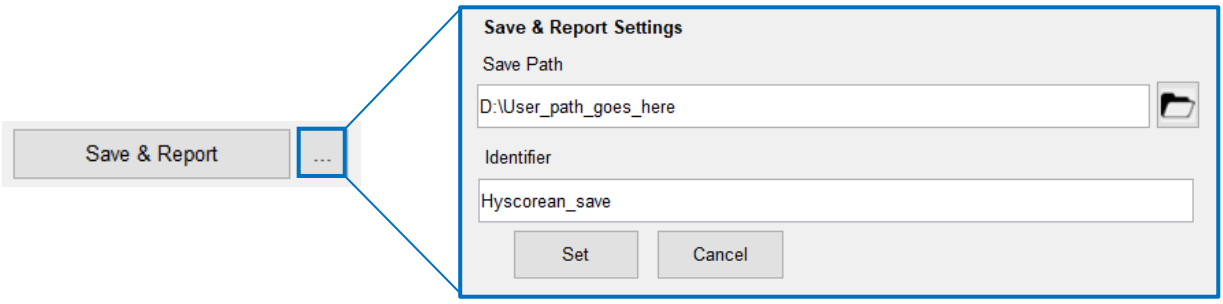


5.2 Saving Hyscorean's session

As mentioned in the introduction of this section Hyscorean offers a saving system which allows the user to maintain an automated book-keeping of all saved results. The user only must press the Save&Report button to prompt the automatic saving system. However, the saving path and name can be defined first by the user.

5.2.1 Setting the save environment

Prior to saving Hyscorean's session the user can change the save setting by pressing the (…) button on next to the Save & Report one, which will prompt a new window to appear:



This new window has two fields: Save Path and Identifier. The file management system works as follows:

- All saved files will be saved on the folder given by Save Path within a subfolder with the following automated name which includes the date of generation and identifier:

  `20181224_Hyscorean_save`

- The saved files will have a name constructed from the date, identifier and file type:

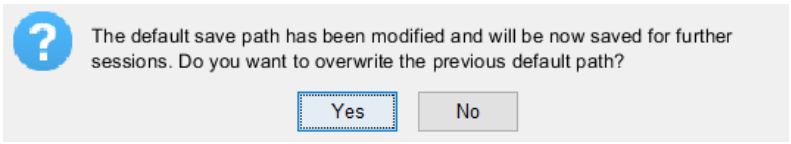
  `20181224_Hyscorean_save_OutputData` `20181224_Hyscorean_save_spectrum`

- Files saved with the same identifier will be have the same name with an appended number to indicate the order of creation:

  `20181224_Hyscorean_save_spectrum` `20181224_Hyscorean_save_spectrum_2`

Therefore, for every different date and identifier a different subfolder will be generated in the folder given by Save Path.

The user can enter the save path manually or select it via a OS window by pressing the  button. The Save Path field is stored between Hyscorean sessions and does not need to be specified again unless the user wants to change it. The Identifier field, however, is reset at each session to its default Hyscorean_save. The user-defined settings can then be stored by means of the Set button. If the save path was added the program will recognize it and ask the user for permission to overwrite the old path.



Answering Yes then closes the Save & Report Settings and return to Hyscorean's main window.

5.2.2 Save & Report

The Save & Report button will generate a set of files consisting the following:

Filename: Date_Identifier_

- settings.mat
  A MATLAB data file containing Hyscorean settings as the ones generated via the Save Settings button, which can also be loaded back via the Load Settings button.

- spectrum.fig & spectrum.pdf
  A MATLAB figure and PDF files containing an exact copy of the spectrum as in the Hyscorean main display.

- OutputData.mat
  A MATLAB data file containing a structure with the following fields:

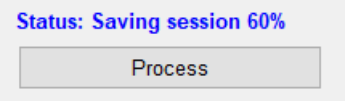
```
.RawSignal          % Signal mounted from raw experimental data
.ProcessedSignal    % Processed signal used for spectrum
.FrequencyAxis1     % X-axis of spectrum
.FrequencyAxis2     % Y-axis of spectrum
.Spectrum           % Spectrum (real, imag or abs depending of choice)
.TimeAxis1          % X-axis of time-domain signal
.TimeAxis2          % Y-axis of time-domain signal
```

- DataForFitting.mat
  A MATLAB data file containing a set of data necessary by the EasySpin Fitting module as input (see section 7.2).

- Report.pdf
  A PDF obtained via the report generator (requires the Report Generator MATLAB license). This creates a report containing all details (experimental conditions, processing parameters, NUS reconstruction details, …) about the signal and spectrum. This enables the user to have a detailed description of the processing in a compact and automated way enabling book-keeping of the measurements, signals and spectra in a comfortable way.

> NOTE:
> Any NaN in the report indicates that the corresponding parameter is not available from the loaded experimental file or does not apply to that kind of spectrometer at all.

The progress of the saving is reported via the status bar with an indication of the percentage of files saved.



Hyscorean - Processing Report

23-Jan-2019 08:56:15

File:
540 files
Path:
D:\lufa\projects\NUS EPR\experiments\20190116 MbStar(NMH) EDA\NUS HYSCORE 15% @3515G\

Table 1. Experimental Settings

MW Freq	9.4900 GHz	**SRT/SPP/Scans**	2 us / 160 / 1
B-Field	3515 G	**B-Field Offset**	0 G
X-Points/TimeStep	200 / 16 ns	**Y-Points/TimeStep**	200 / 16 ns
Video Gain	NaN dB	**Video Bandwidth**	NaN MHz
Pulse Lengths 90°/180°	16 ns / 16 ns	**Tau Values**	[128 160 192] ns

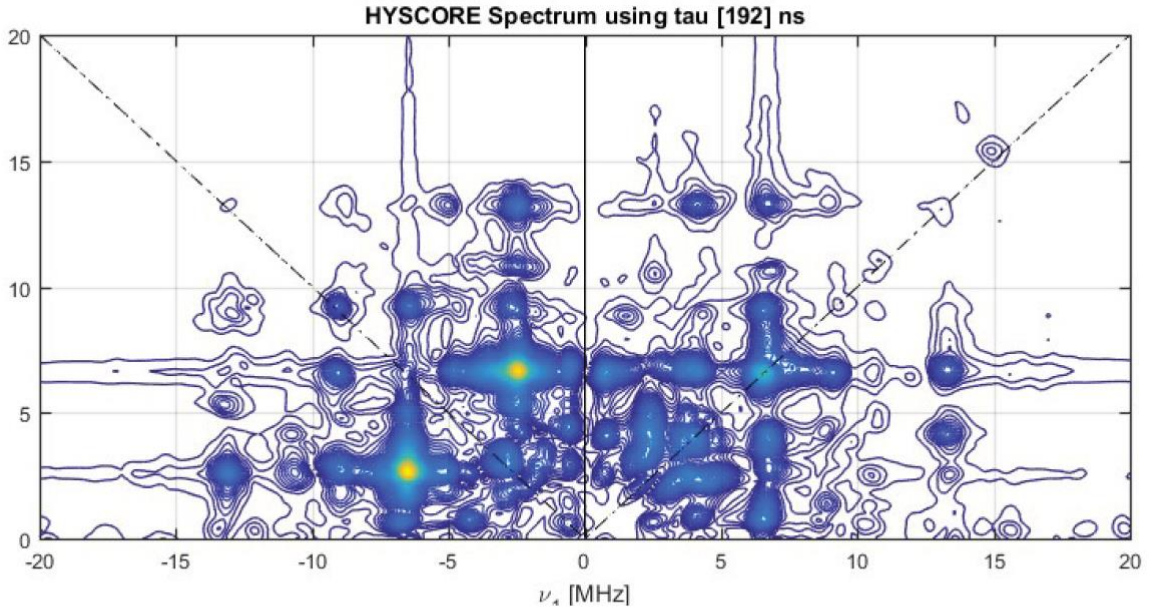
Table 2. Processing Settings

1st Corrected Dimension	**Fitting Model**	**Polynomial Order**	**Background Start**
1	Polynomial	1	0 ns
2nd Corrected Dimension	**Fitting Model**	**Polynomial Order**	**Background Start**
2	Polynomial	1	0 ns

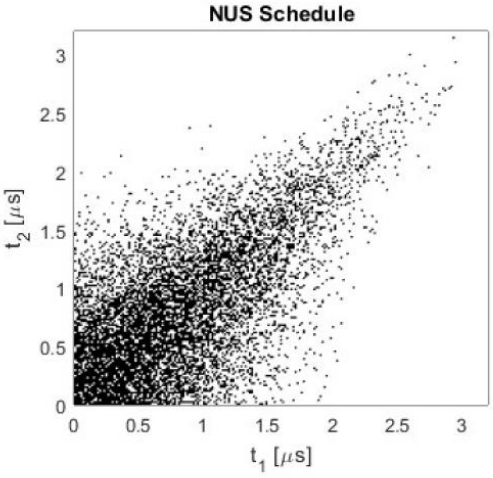
Lorentz-to-Gauss TF	**Apodization Window**	**Window Length 1**	**Window Length 2**
deactivated	blackman	200	200

Zero-filling in t1	**Zero-filling in t2**	**Symmetrization**
200 points	200 points	None

Contour Levels	**Minimal Contour Level**	**Spectrum**
100	1 %	Absolute



- This report has been automatically generated by Hyscorean -



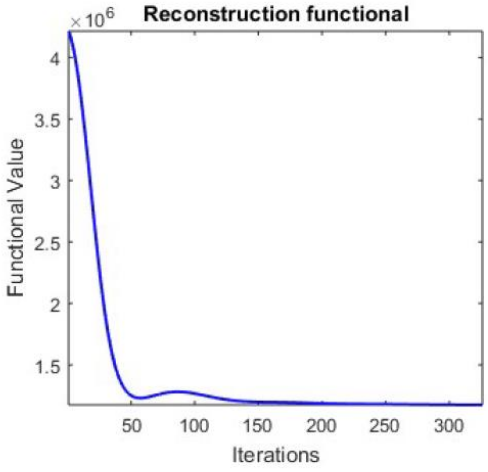
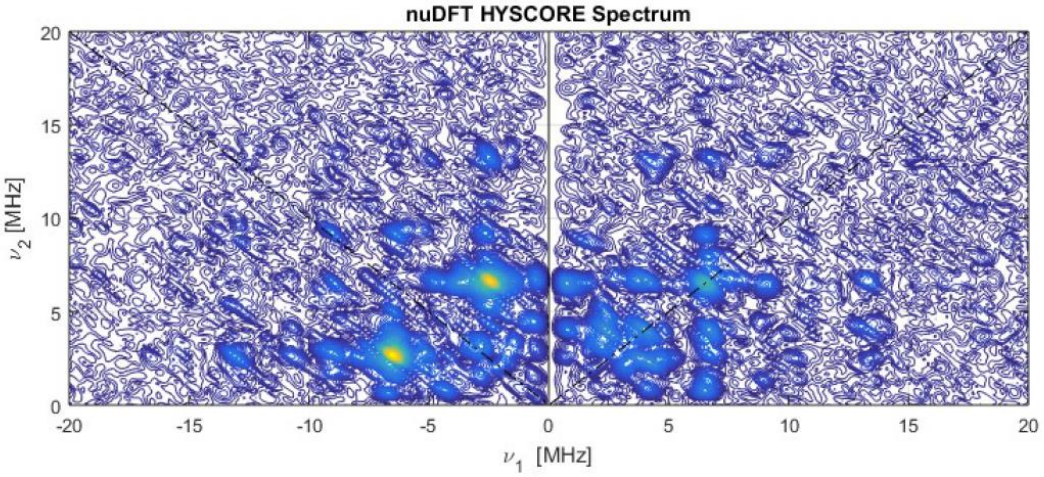


Table 3. Non-Uniform Sampling & Reconstruction Settings

Sampling Density	Reconstruction Method	Lagrange multiplier	Background parameter (log10)
15.00%	Constant-aim CAMERA	400	-5



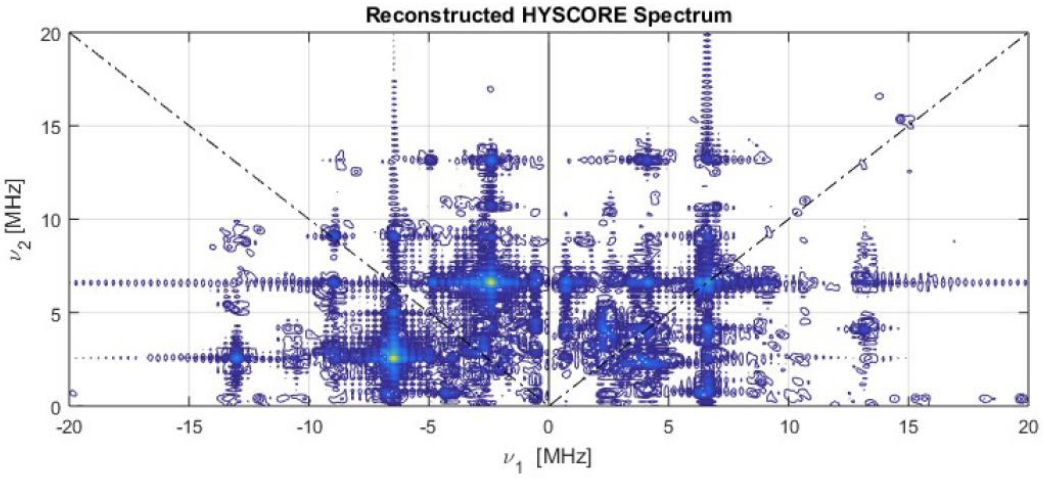


Figure 1. HYSCORE Time Trace - First Background Correction

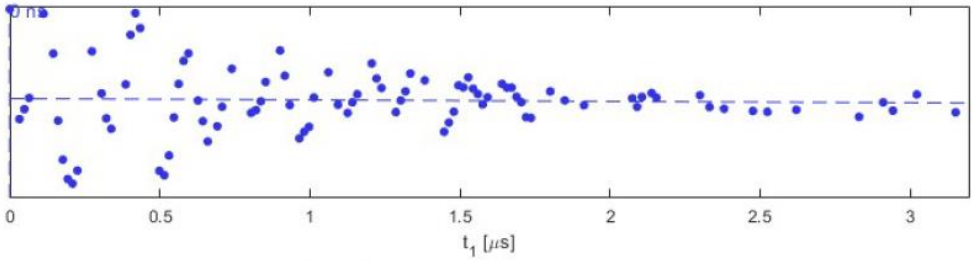


Figure 2. HYSCORE Time Trace - Second Background Correction

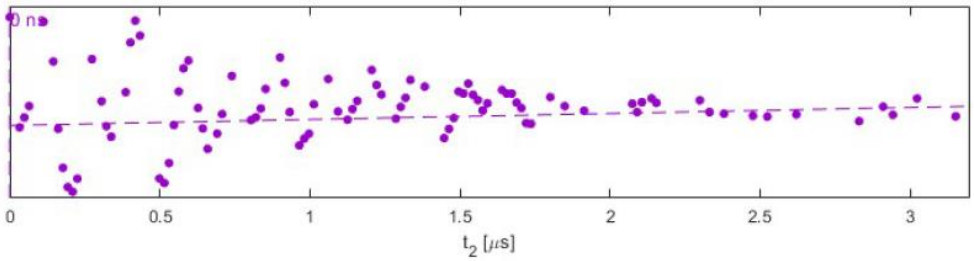


Figure 3. HYSCORE Time Trace - Processed Signal (First Dimension w/o Zero-Filling)

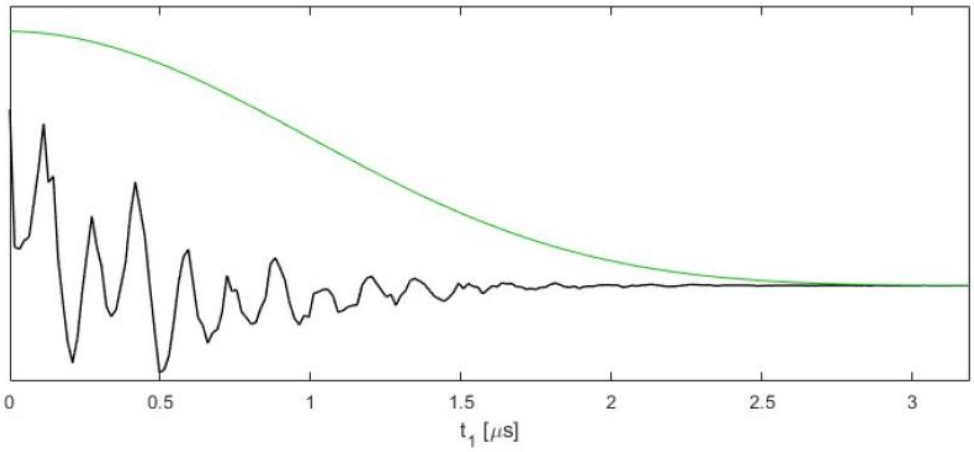
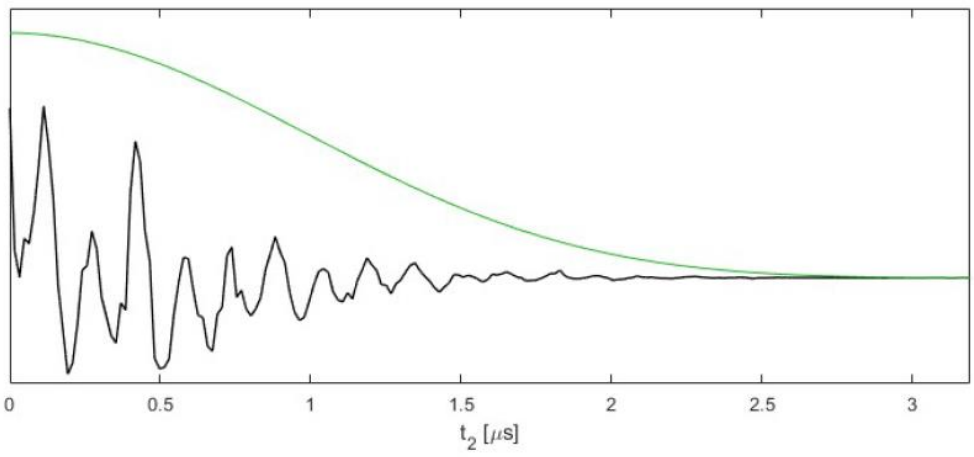


Figure 4. HYSCORE Time Trace - Processed Signal (Second Dimension w/o Zero-Filling)



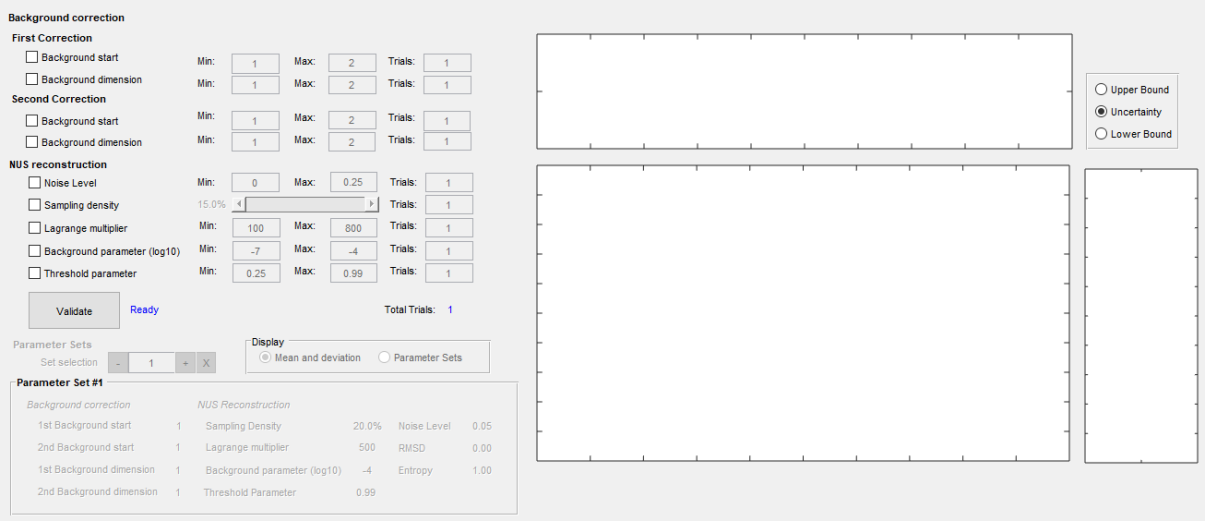
6. Validation module

The implementation of non-uniform sampling and spectral reconstruction represents a powerful experimental technique to reduce measurement times of HYSCORE experiments. The reconstruction methods available in Hyscorean (see section 4.3.) also represent a flexible framework to reconstruct the spectra. Nonetheless, these methods introduce not only a certain number of parameters but also an uncertainty into the spectrum which the user may not feel comfortable with. This uncertainty can have many origins such as the sampling density of the NUS schedule, the noise in the signal, reconstruction parameters or background correction. Also, the fact that spectral reconstruction is based on regularization methods makes it not immediately intuitive.

Therefore, Hyscorean offers the possibility to validate the HYSCORE spectra by validating all the factors which give raise to uncertainty. Validation is nothing more than a statistical error analysis that allows the user to get an estimate or measure of the uncertainty of the different regions in the spectrum by defining the uncertainty in the different parameters. It generates a set of HYSCORE spectra from different sets of parameters and then performs a simple statistical analysis of the obtained spectra to give different measures such as mean as well as upper and lower bound spectra.

The validation of HYSCORE spectra is not only useful to assess uncertainty of NUS HYSCORE spectra but can also be used to check the influence of the background correction parameters on normal HYSCORE spectra.

In Hyscorean the validation module can be started from the Validation button in the Pre-Processing panel. The button will only become active once the data has been processed for the first time. Pressing the button will prompt the validation module GUI to appear:

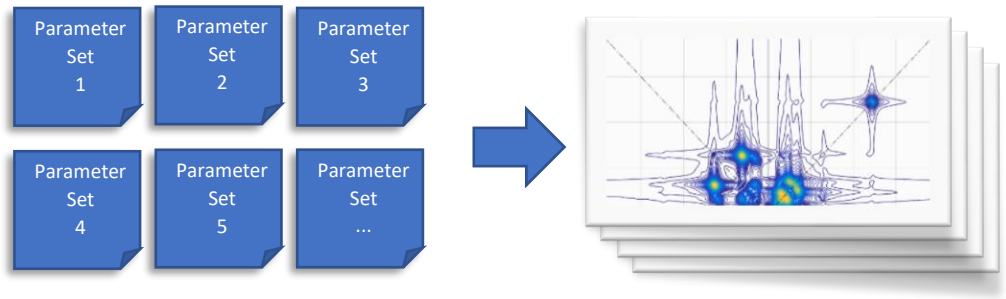


Depending on the type of data being processed (uniform or non-uniform sampled) different UI control elements will be disabled, meaning that the parameters they represent are of no relevance for the validation of the current data.

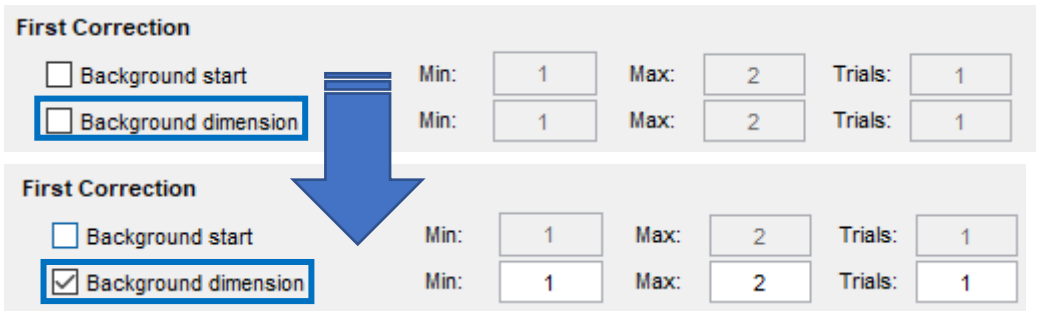
The following sections will cover the basics of Hyscorean's validation of background correction and NUS reconstruction.

6.1 Basics of the validation module

As mentioned in the previous section the goal of the validation module is to generate an ensemble of parameter sets and for each of them, generate a HYSCORE spectrum by going through the adequate processing protocols as in Hyscorean's usual processing.



By default, all validation parameters are deactivated, they can be activated by pressing the corresponding check boxes:

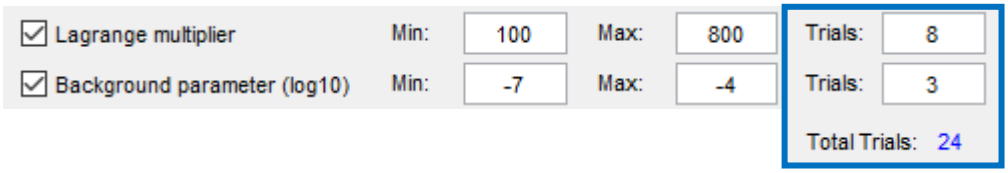


The program then constructs the parameter sets as follows. For any activated parameter, a vector of different parameter values is generated. This is a linearly spaced vector with minimal value given by the Min edit box, maximal value given by the Max edit box and number of elements corresponding to the edit box Trials. The vector is constructed in MATLAB as follows:

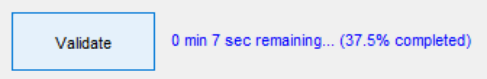
```
ValuesToValidate = linspace(Min,Max,Trials);
```

For all deactivated parameters, the vector contains a single default value. These values are automatically set to match the values employed in the Hyscorean processing at the time the Validation button was pressed.

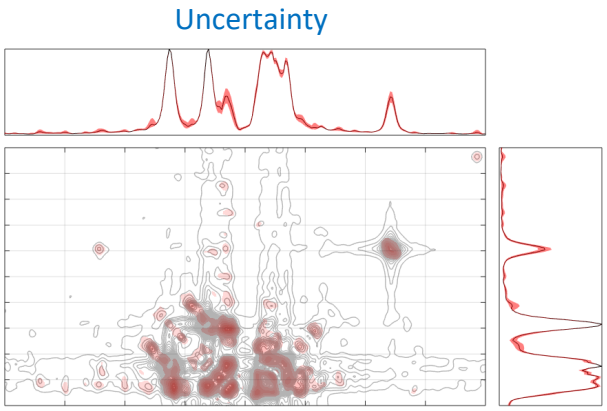
Once all vectors are constructed, a parameter set is constructed for each possible combination of the vector elements of all parameters. The total number of parameter sets is automatically updated each time a change is done and displayed under the Total Trials field.

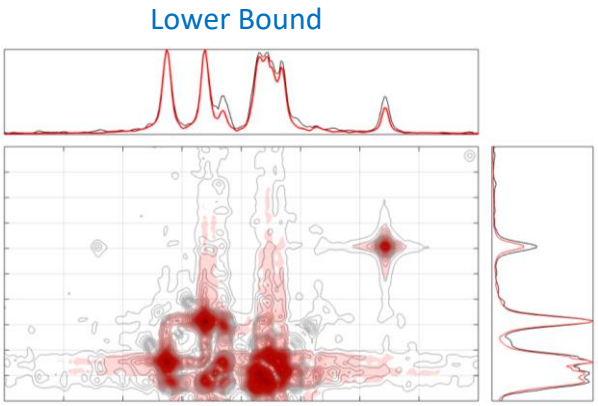


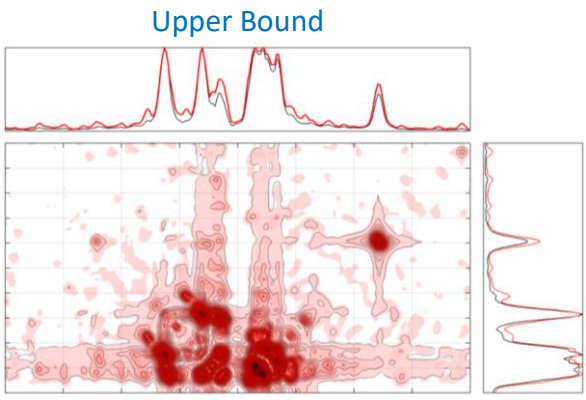
Once the desired validation settings have been set, the validation can be started via the Validation button. This will generate the parameter sets and start processing the corresponding HYSCORE spectra. Since the program automatically generates all possible combinations of parameters, the number of total trials can adopt very large numbers easily. This is usually not an issue since the processing of HYSCORE spectra is quick. However, for NUS reconstruction the computation times can be moderately larger and therefore lead to large validation times. For this reason, the program gives an estimate of the remining time for completion of the validation as well a percentage of completed parameter sets.



Once finished the validation results are presented in the main display. The mean HYSCORE spectrum is displayed as a grey contour plot superimposed by a red colormap. This red colormap represents the value selected in the box next to the display and can either be the Uncertainty, Lower Bound, or Upper Bound of the validation. In the insets the corresponding skyline projection along each dimension are displayed.



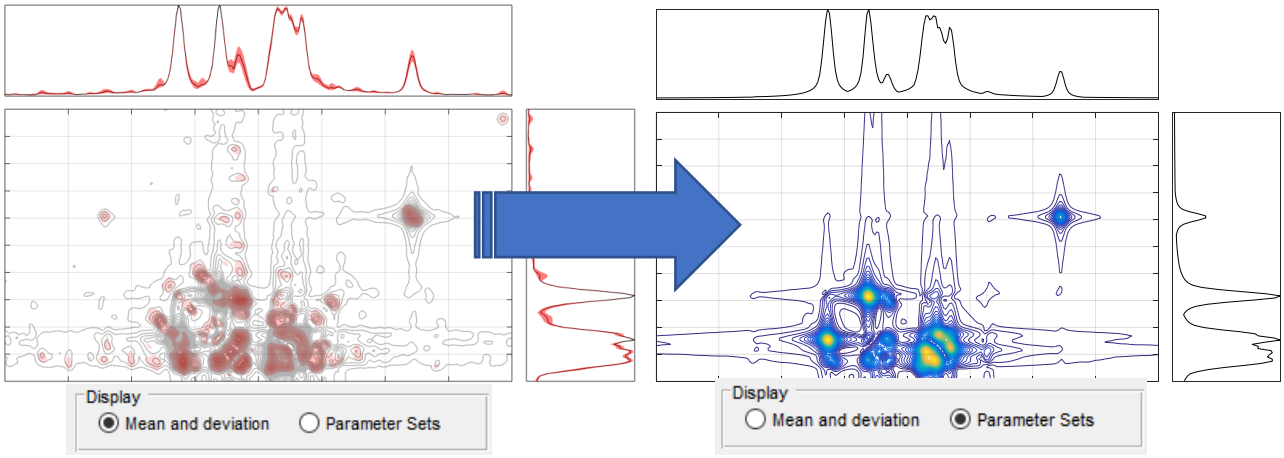




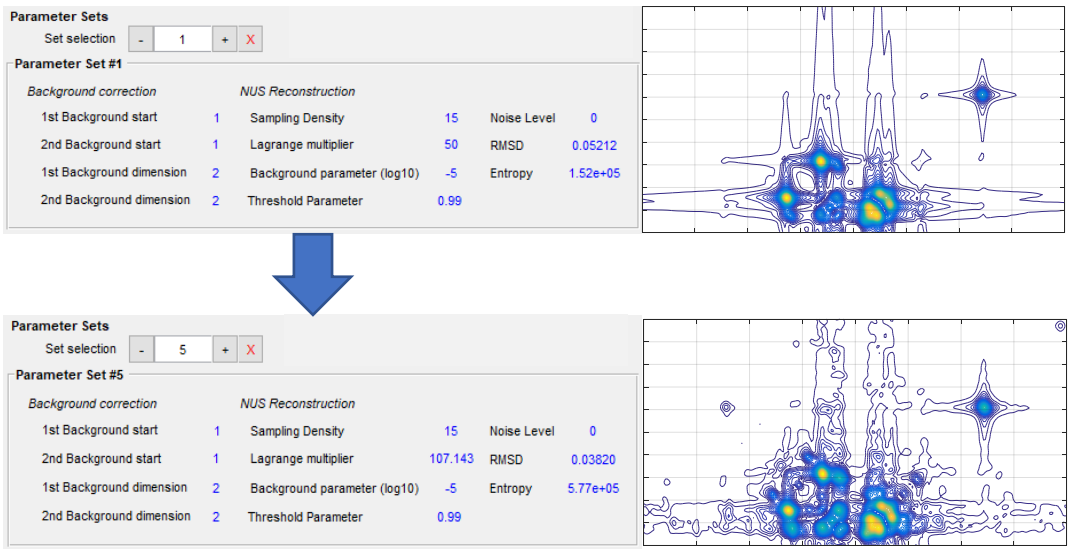
The uncertainty is defined as the standard deviation of all HYSCORE spectra, whereas the lower and upper bounds of the 95%-confidence interval computed according to the two-sigma rule of the 68-95-99.7 rule [11].

The can again be used to zoom-in and out of the display and its insets and the detach button can be used to copy the current display to a separate figure where it can be saved or copied.

This display mode shows the validation results and can be used directly to assess the uncertainty in the spectrum. However, the user can also access the individual HYSCORE spectra computed during the validation as well as the corresponding parameter sets. The display mode can be changed via the Mean and deviation and Parameter Sets buttons in the Display panel.



The user can then browse through the different parameter sets and spectra via the Set Selection buttons. The current parameter set values are always displayed in the Parameter Set # panel. Additionally, the RMSD and entropy values of the reconstruction methods are shown in case NUS reconstruction is part of the validation.



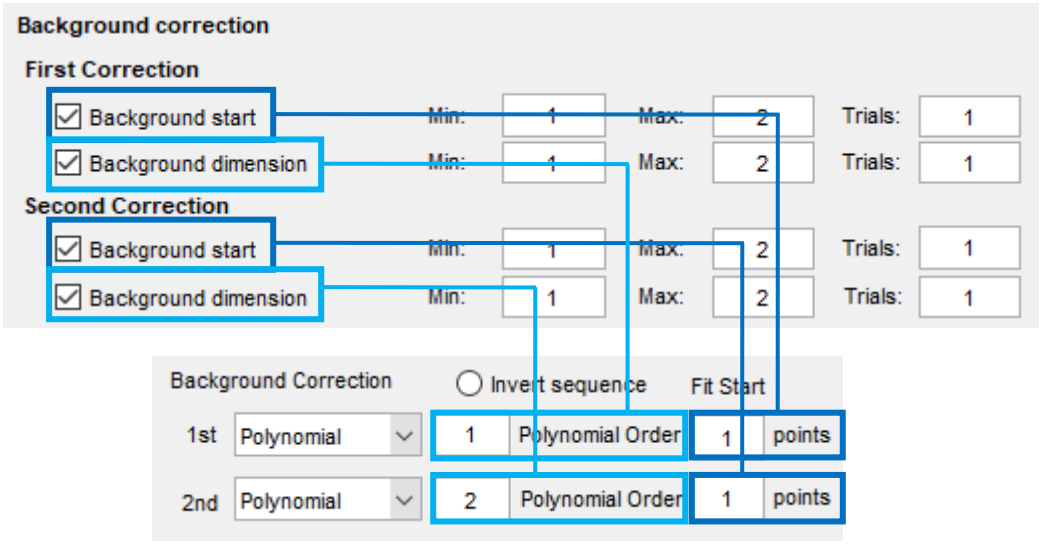
At any point after the validation, parameter sets can be deleted via the X button from the Set Selection buttons. After deleting the parameter set, the mean HYSCORE spectrum as well as the uncertainty are updated automatically. Any deleted set cannot be recovered once deleted unless the validation is started from the beginning.

6.2 Validation of background correction

As mentioned at the beginning of section 6, Hyscorean allows the validation of the background correction for HYSCORE data obtained from uniform and non-uniform sampled experiments. Validation of the background can become quite relevant even in uniformly-sampled HYSCORE data when presented with signals containing pronounced backgrounds and crosspeaks at lower

frequencies. Background correction can easily remove such low frequencies. For NUS data the background fitting can also become very sensible to the parameters of choice due to the smaller amount of points to fit the background.

As the background correction in Hyscorean is performed for each dimension sequentially the validation can be performed for each of the corrections individually as well. The Background dimension parameters represent the values given in Hyscorean for each model which depending on the model selected appear with different names (see section 3.1.). The Background start parameters serve to validate the starting values employed for the fit. The relation between the validation parameters and the Hyscorean Pre-Processing panel are:



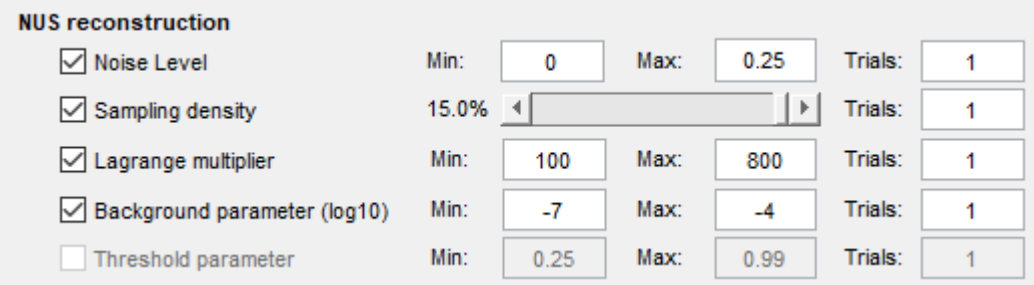
NOTE:
The background validation does not validate the model employed to fit the background. This is directly taken from the chosen models in the Background Correction section of the Pre-Processing panel in Hyscorean. If the model wants to be changed, the validation module needs to be closed and the models changed there before opening the module again.

The Min and Max values for these parameters only accept integer numbers. Therefore, if the number of trials given in Trials exceeds the number of integers between Min and Max, the Trials value will be automatically adapted to the maximum possible.

6.3 Validation of NUS reconstruction

The validation of the reconstruction of non-uniform sampled (NUS) spectra is a powerful tool to assess the uncertainty introduced by the reconstruction methods introduced in 4.3. and, therefore, to be able to trust the results ad conclusions obtained from such spectra. This is especially useful (if not critical), when employing methods such as the constant-λ CAMERA reconstruction methods (see 4.3.2.2.), whose parameter space can be substantial and where no a priori criteria are known to choose such parameters.

The validation module allows the user to validate all input parameters given in the NUS reconstruction section. The Lagrange multiplier and Background Parameter (log10) fields are only enabled when the NUS HYSCORE data has been reconstructed with maxEnt methods and the Threshold parameter is only enabled when the IST methods are used. Note that the threshold parameter of the IST methods is not given as an input in Hyscorean since there it is always fixed at a value of 0.99. Nonetheless, this can be validated in here.



NOTE:
The validation does not validate the reconstruction method. This is directly taken from the chosen method in the NUS Reconstruction section of the Pre-Processing panel in Hyscorean. If the method wants to be changed, the validation module needs to be closed and the method changed there before opening the module again.

The reconstruction methods are regularization methods and thus the results obtained via them have a dependency on the noise present in the measurement. Therefore, the validation module offers the chance to validate the noise effects on the results. Of course, it is not possible to change the innate noise in the signal but an upper bound can be assessed by adding additional white noise to the signal. Therefore, the field Noise Level determines the level of the added white noise relative to the signal maximum.

Another dependency of the results introduced by non-uniform sampling is the dependency on the NUS grid employed for the measurement. The program validates the sensibility of the reconstructed spectra to changes in sampling density. Again, it is impossible to validate sampling densities above the measured one, but a lower bound can be assessed by validating lower sampling densities.

The Sampling Density field contains a slider whose values range from zero to the sampling density employed for the measurement and there the minimal value can be set for the validation. The program then constructs the linearly spaced vector with the number of elements given by Trials and values between the slider value and the experimental sampling density. To reach the validation sampling density the program randomly removes sampled points until the desired sampling density is reached.

NOTE:
Validation of the Sampling Density and Background Parameter fields can lead to substantial validation times, especially for low values. This must be considered when planning the number of trials.

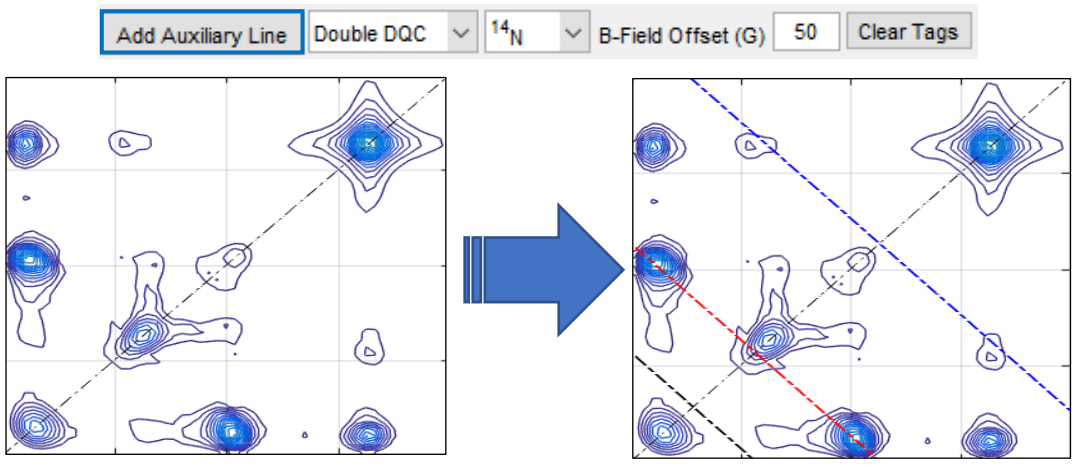
7. EasySpin fitting module

Hyscorean creates the link between processing and fitting by including a fitting module based on EasySpin [12] simulation routines.

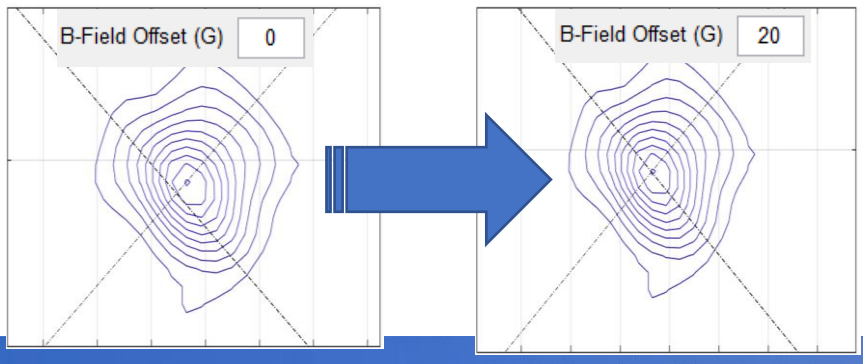
7.1 Auxiliary lines & Field offset

One of the first steps to prepare for the fitting is to determine the magnetic field offset of the current measurement. This offset can be determined from a matrix peak on the HYSCORE spectrum from a known nuclear Larmor frequency. Hyscorean facilitates this procedure by including auxiliary lines which can be used for calibrating the field offset.

These lines can be added from the Add-Ons panel on top of the main display, by pressing the Add Auxiliary Line button which will appear at the resonance frequency of the nuclei currently selected on the list box of the same panel. The user can also select the type of transition to be computed: single-quantum coherence (SQC), double-quantum coherence (DQC) or double double-quantum coherence (double DQC). Depending on the coherence-type of choice the added dashed line will have a different color: black for SQC, red for DQC and blue for double DQC.The resonance frequency is computed by means of the gyromagnetic ratio of that nuclei, the magnetic field employed in the measurement and the magnetic field offset given by the B-Field Offset edit box in the same pan el. All lines can be removed from the spectrum by means of the Clear Lines button.



Then to calibrate the magnetic field offset one can adjust its value via the B-Field Offset edit box and add a new line. The offset value corresponds to that at which the line crosses the center of the matrix peak, e.g. proton matrix peaks. The value can be fine-tuned by zooming in the corresponding matrix peak and adding the line:



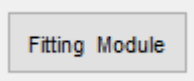
7.2 Starting the fitting module

Hyscorean's EasySpin fitting module is a revamped version of the well-known EasySpin esfit function adapted for HYSCORE spectra. However, a big difference to EasySpin (besides the display) is the fact that each simulation performed by EasySpin to fit the HYSCORE spectra is processed the same as the experimental spectrum in Hyscorean (with exception of the background correction which is performed differently). This can make big differences in the RMSD-based fit of EasySpin when using processing techniques such as the Lorentz-Gauss transformation or some apodization windows.

Hyscorean makes this connection between processing and simulation automatically and without the need for the user to give any input. There exist two routes to start the fitting module of Hyscorean depending on the needs of the user.

7.2.1 Loading single spectra

The most straightforward way to start the fitting module is directly from the Hyscorean main window by pressing the Fitting Module button.



This will open the fitting module GUI and load all required data for the fit: spectrum and axis, processing settings, experimental values (e.g. magnetic field and offset, τ-values, time-sweeps steps, …) and construct all corresponding EasySpin structures.

7.2.2 Loading multiple spectra

It may be of interest for the user to fit several HYSCORE spectra of a same system measured at different magnetic field positions. Hyscorean's fitting module allows the simultaneous fitting of several spectra at the same time via the same interface as for single spectra. This, however, is started in a different way although with minimal effort for the user:

I. Process all HYSCORE spectra in Hyscorean

II. Save them individually via the Save&Report button. As described in 5.2.1. this will create an output file with the suffix DataForFitting.mat containing all necessary input for the fitting module.

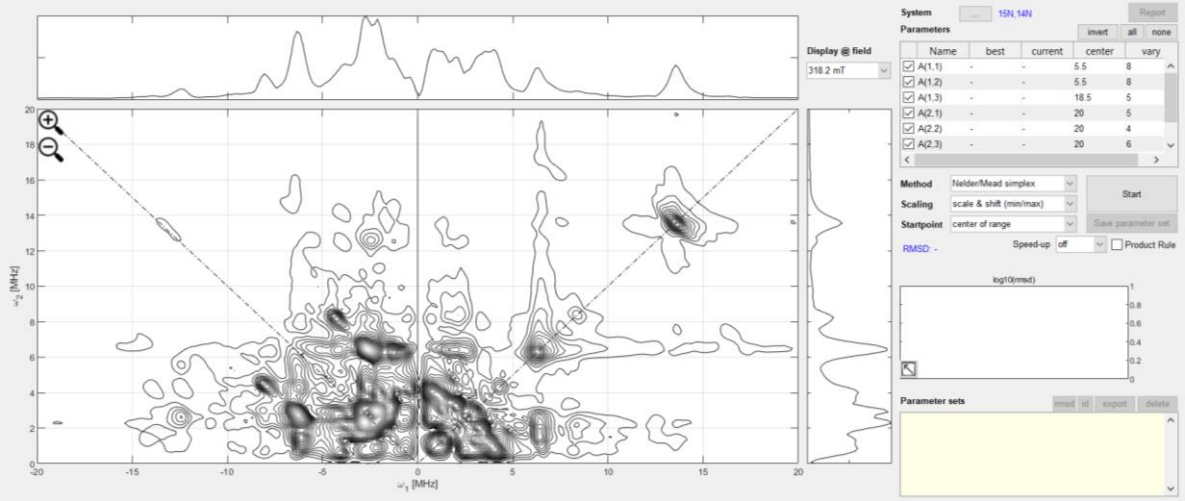
III. Create a MATLAB script which defines a variable containing the paths and filenames of those files and call the validation module as a standalone GUI via the launch_Hyscorean_fit function. The script should look as follows:

```
clear Files

Path = 'D:\This\Is\An\Example\Path';
Files{1} = '20181206_FieldPosition1_DataForFitting';
Files{2} = '20181206_ FieldPosition2_DataForFitting';
Files{2} = '20181206_ FieldPosition3_DataForFitting_2';

launch_Hyscorean_fit(Files,Path)
```

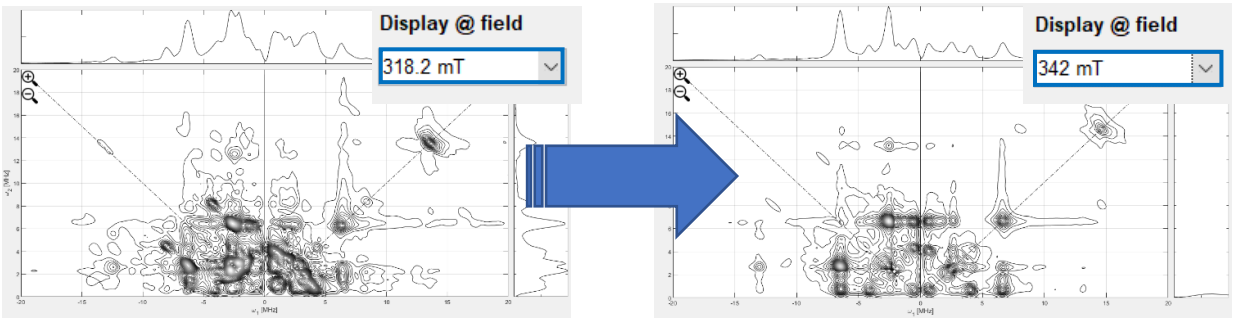
IV. By running the script, the fitting module will be started, and all of the spectra will be loaded along their individual experimental and processing settings.

7.3 Fitting HYSCORE spectra

Once the Hyscorean fit module is started as described in the previous section the following window will open. Users familiar with the EasySpin fitting tool GUI will recognize familiar elements on this version:



The loaded HYSCORE spectra will be automatically displayed as a monochromatic contour plot in the main display with skyline-projection plots along each dimension in the insets. The spectra can be switched at any time (even during fitting) by means of the Display@field list box. This box contains the field positions of the spectra loaded into the fit module. By selecting a different field position, the program will switch the main display to the corresponding spectrum.

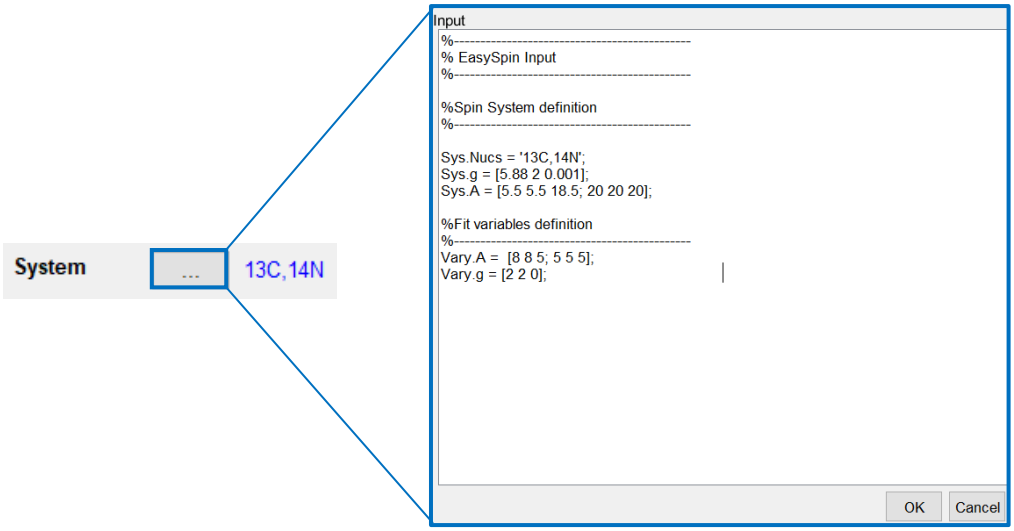


NOTE:
If the user changes the field position to be displayed while the fitting algorithm is running, the spectrum will not be changed until the current iteration is finished and the program updates the spectrum. Therefore, should simulations take long computation times the change of display will also require the same time.

As in the main Hyscorean window, the spectra can be zoomed-in and out via the same buttons. The insets adjust automatically to the zoom status of the main display. The limits to which the zoom-out button resets are given by the Axis Frequency Limit in Hyscorean set during the processing.

7.3.1 Defining the spin system

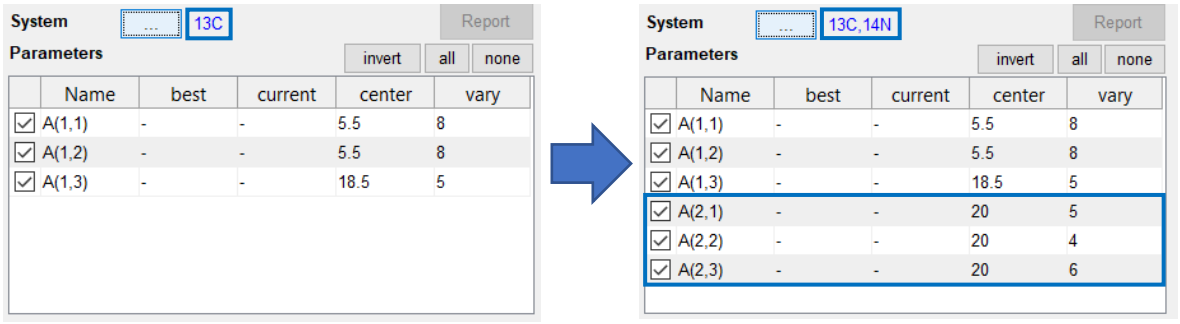
In contrast to EasySpin in Hyscorean the spin system is defined via the GUI and allows the user to modify it without the need to start the module anew. This can be done via the (...) button next to the System tag. The current system's nuclei considered are displayed next to this button as a blue text.



By pressing the button, a new window is opened which allows the user to modify a text box where the EasySpin Sys and Vary structures can be constructed using MATLAB notation. The spin system is defined by the Sys structure and the fitting parameters are given in Vary. For further details, please refer to the EasySpin documentation.

> NOTE:
> The Exp and Opt variables usually required by EasySpin for simulations are automatically constructed from the experimental parameters in the measurement files and do not need to be constructed by the user. The user can, however, define further fields on those structures from the same window.

By pressing the OK button, the window is closed, and the input is compiled. Should there be any errors in the spin system definition, the program will recognize this and prompt an error window and request the user to check the input again. Once the input has been compiled, the new Vary structure parameters will be updated on the Parameters table immediately.



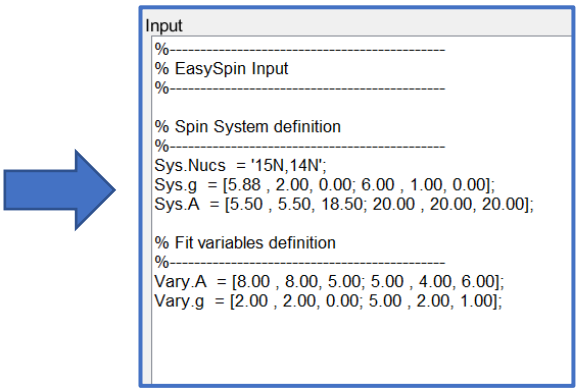
The table rows indicate the different parameters of the spin system which are to be fitted and the columns indicate the following properties:

- Name — Descriptor of the parameter to be fitted as defined in the input
- Best — Value of the parameter in the best fit spectrum
- Current — Value of the parameter for the current iteration spectrum
- Center — Value given to the parameter in the Sys variable
- Vary — Variation range of values which the parameter can adopt

Next to this table a set of buttons can be pressed to Invert the check boxes in the table, select them All or None.

The user has also the option to pass the spin system programmatically from MATLAB scripts as an input to the launch_Hyscorean_fit function, which then will pass the Sys and Vary as variables to Hyscorean's fit module. This can be done by passing the Sys and Vary structures as fields of the structure which is given as input:

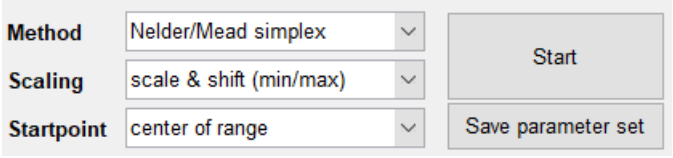
```
Sys.Nucs = '15N,14N';
Sys.g = [5.88 2 0.001];
Sys.A = [5.5 5.5 18.5; 20 20 20];

Vary.A =  [8 8 5; 5 4 6];
Vary.g = [2 2 0; 5 2 1];

Input.Sys = Sys;
Input.Vary  = Vary;

launch_Hyscorean_fit(Files,Path,Input)
```



NOTE:
Hyscorean stores the text used for the spin system definition as a MATLAB preference. This means that once the program is closed, the next time it is opened the text in the window (and thus the spin system definition) will be exactly the same as when it was closed.

7.3.2 Starting/Stopping the fitting

With the spin system defined, the fit settings can be selected to start the fitting. All the options available for the fitting can be chosen from the list boxes next to the Start button.



❖ Method

The optimization algorithm to be employed to find the best fit spectrum corresponding to the minimum of some optimization functional. The details can be found on the EasySpin online documentation [13]:

EasySpin Documentation

Beyond a good starting parameter set or search range, the performance of the fitting depends crucially on two things: the choice of the optimization algorithm, and the choice of the target function. Let's have a look at each of them in turn.

Optimization algorithms

EasySpin provides several optimization algorithms that are in widespread use: (1) the Nelder/Mead downhill simplex method, (2) the Levenberg/Marquardt algorithm, (3) Monte Carlo random search, (4) a genetic algorithm, (5) a systematic grid search, as well as others.

The first two are local search algorithms, which start from a given starting set of parameter values and try to work their way down a nearby valley of the parameter space to find the minimum. Both methods are quite fast, although there are some differences in general performance between them: The downhill simplex is somewhat slower than Levenberg/Marquardt, but it is more robust in the sense that it does not get stuck in a local minimum as easily as Levenberg/Marquardt.

The latter three are global search methods: they do not have a single starting parameter set, but use many, distributed over the entire parameter search space. The Monte Carlo method simply performs a series of random trial simulations and picks the best one. It is very inefficient. The systematic grid search is better: It covers the parameter space with a grid and then does simulations for each knot of the grid, in random order. Thus, no point is simulated twice, and the method is more efficient than the Monte Carlo search. However, if the minimum is between two grid points, it will never be found.

The third global method is a genetic algorithm: It makes simulations for several, let's say N, parameter sets (called a population), computes the fitting error (called the fitness) for all of them and then proceeds to generate N new parameter sets from the old ones using mechanisms like mutation, cross-over and reproduction. This way, a new generation of parameter sets is (pro-) created, just like in biological evolution. The benefit of this algorithm is that if a good parameter is encountered, it is likely to propagate down the generations and across the population.

Fitting EPR spectra, EasySpin Documentation, Stoll et al.

❖ Scaling

The different ways the spectrum can be scaled once it has been simulated to compare it with the experimental spectrum. The options are:

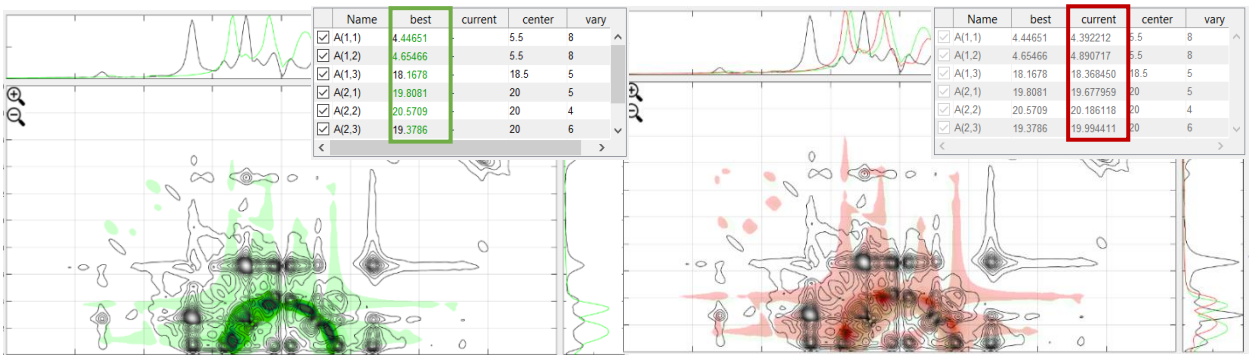
- scale & shift
  The spectra are scaled to the absolute maxima and their absolute minima are shifted to match
- scale only
  The spectra are scaled only to the absolute maxima.
- no scaling
  The spectra are used as simulated without scaling

❖ Startpoint

The way the initial values of the fitting parameters are selected at the start of the optimization routine. The options are:

- center of range
  Starts with the values given in the center column of the Parameters table.
- random within range
  Starts at a random value within the range given by the values given in the center and vary columns of the Parameters table.
- selected parameter range
  Starts at the values saved in the currently selected parameter set in the Parameter Sets panel (see later).

Once the user has chosen the settings for the fitting, this can be started via the Start button. Hyscorean employs the following color coding to show the state of the fitting in the main display as well as in both insets:



❖ Black

Experimental spectrum as processed and saved by Hyscorean.

❖ Green

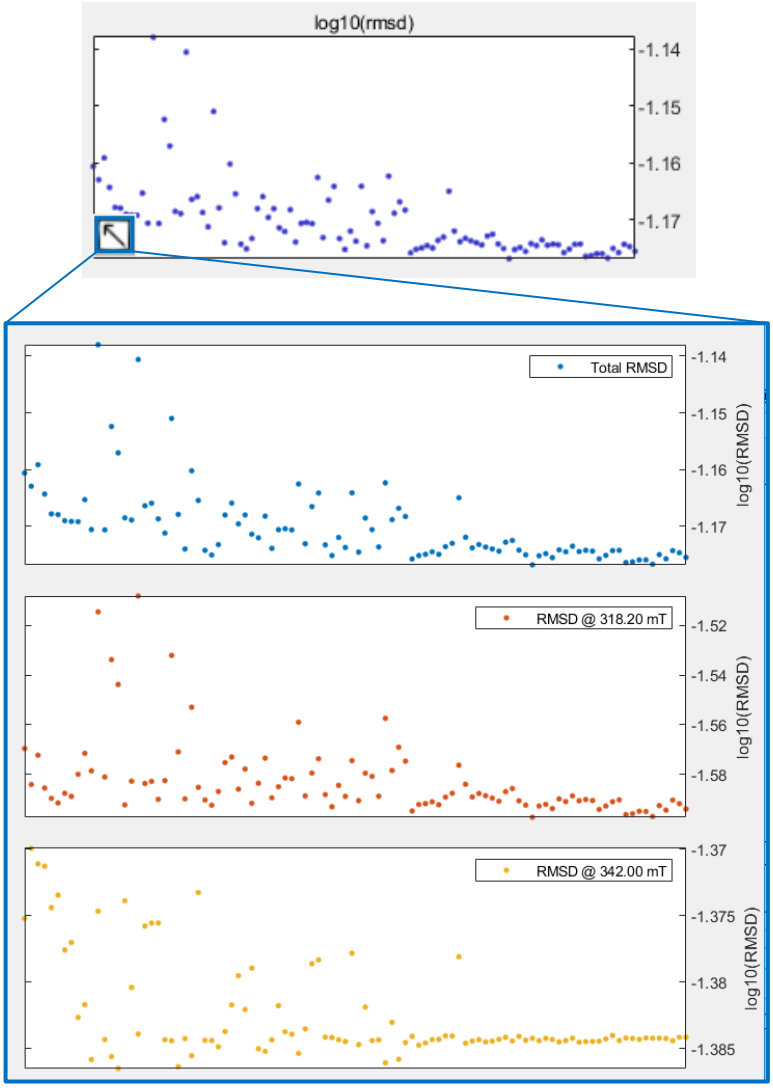
Best fit spectrum given by the parameters in the best column of the Parameters table. Displayed during and after the fitting.

❖ Red

Spectrum corresponding to current iteration with the parameters in the current column of the Parameters table. Displayed only during the fitting.

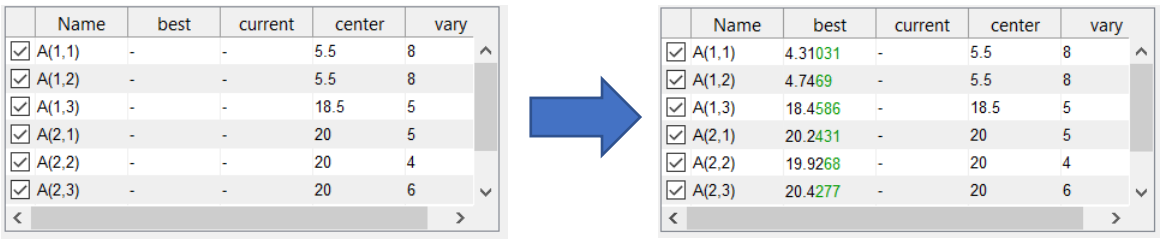
The fitting can be also monitored via the evolution of the RMSD values of the fitted spectra in the display below the Start button. The RMSD displayed in here is the total RMSD computed from the sum over the individual RMSD values of the different spectra being fitted (e.g. if several field positions are being fitted).

Nonetheless, in case the user wants to monitor the RMSD evolution of the individual fitted spectra the user can press the detach button to prompt a new window which will contain an individual display for each of the spectra RMSD as well as one for the total RMSD.



Once the fitting has started, all UI control elements are disabled and the Start button is changed into the Stop button. This Stop button can be used to stop the fitting prematurely even if the optimization algorithm has not found the minimum yet. Otherwise, the fitting will continue until a minimum is found, point at which the fit ends and all UI control elements are enabled again.

In either case, once the fitting routine has stopped, the parameters of the best fitted spectrum are updated at the Parameters table under the best column:



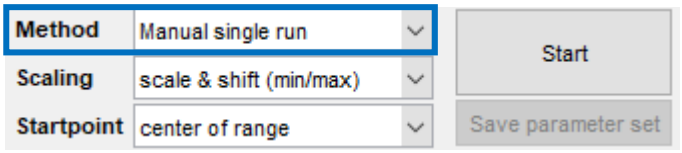
NOTE:
By pressing the Stop button, a trigger is sent to the program to stop the optimization at the next iterate. Therefore, pressing the button will not result in an immediate abortion of the fitting. If the simulations are computationally costly then the program may require a prolonged lapse of time until it is fully stopped. Nonetheless, the user can abruptly kill the fitting by pressing the Crtl+C combination in the MATLAB console. This will result in a complete loss of the fit data.

7.3.3 Manual fitting

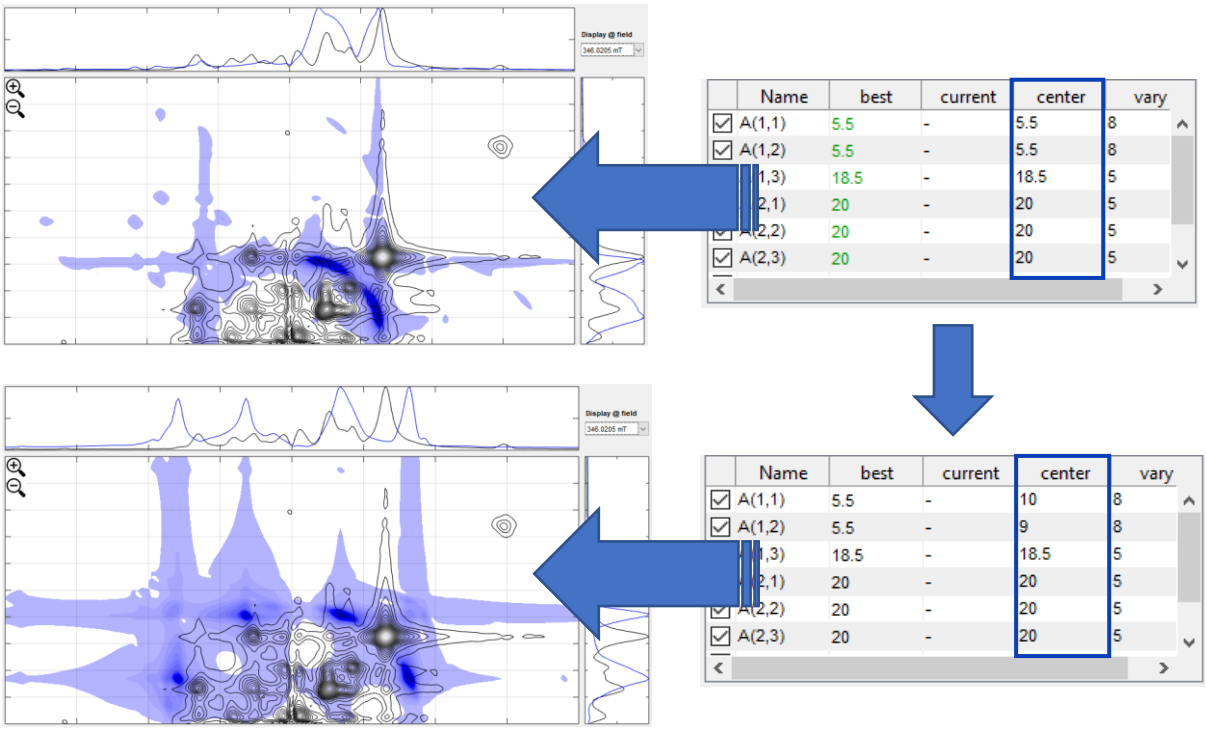
If the user wants to, the spectra can be fitted manually without the need to rely on an optimization algorithm. This can be beneficial to find a first good approximation of the fit parameters or to just try to fit the spectrum by hand when the optimization methods fail to.

This procedure is nothing else than a single-run simulation of a HYSCORE experiment with the parameters given in the center column of the Parameters table. The current values can be modified directly from the table without the need to redefine the spin system. Thus, any changes made directly in the table will not affect the input given in the System definition window.

To enable the manual fitting, the Manual single run option in the Method list box must be selected. By pressing the Start button the simulation with the current parameters will be simulated and return the RMSD of the manual fit and update the best column as in the usual fitting.

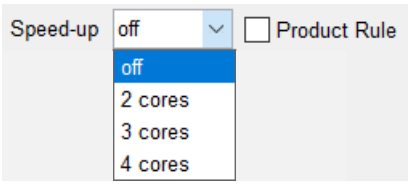


To visually better differentiate manual from automatic fitted spectra, the color coding for manual fitting is changed with respect to the automatic fitting mentioned in the previous section. The fitted spectra are now represented as a blue colormap on top of the usual black contour of the experimental spectrum.



7.3.4 Speeding-up the simulations

HYSCORE spectra often require large spin systems with multiple nuclear spins and the fitting of several magnetic field positions. These facts combined contribute to potentially very long simulation times for EasySpin which overall slowdown the fitting and lead to long computational times. Hyscorean offers a couple of options to potentially speed-up those fitting times.



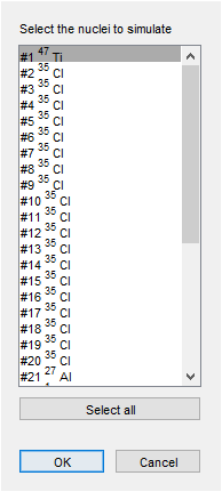
If the spin system contains more than two nuclear spins it may be beneficial to use the Product Rule options of the saffron simulation function since it may enhance simulation times. By default it is disabled until the check box is clicked.

Another feature available in Hyscorean is to distribute the load of simulating different spectra in different workers of the computer. This allows the different field positions to be simulated at the same time in parallel largely enhancing the computation times. Hyscorean automatically detects the available workers to be set in parallel and these can be selected from the Speed-up list box. This requires the Parallel Computing Toolbox of MATLAB to be properly installed and with a valid license.

Further speed-up options integrated in EasySpin can be added to the Opt structure via the System input window. For details, see the EasySpin documentation for the saffron function.

7.3.5 Simulating HYSCORE from ORCA results

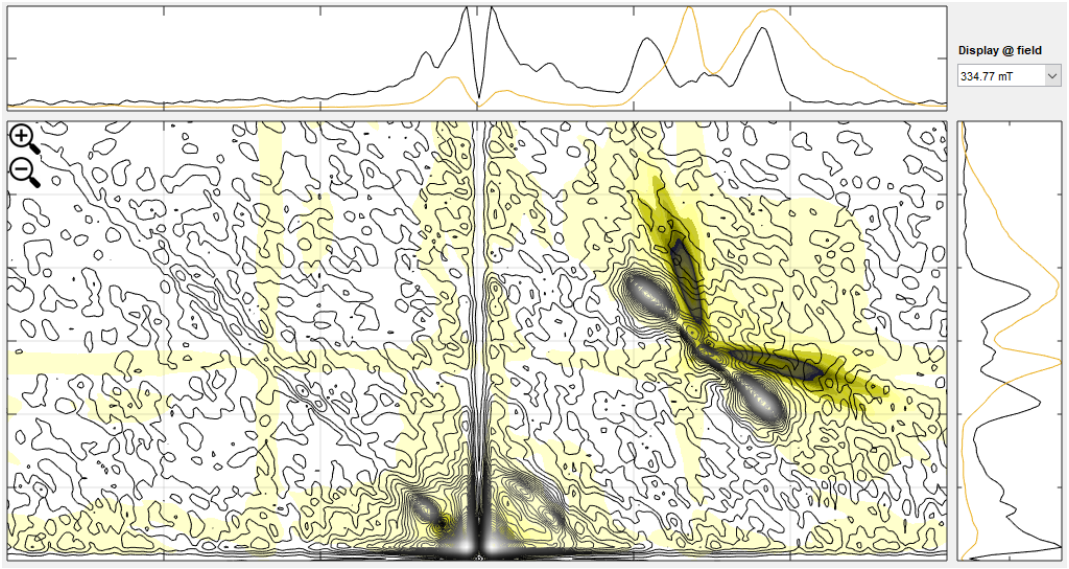
An additional feature of Hyscorean's fitting module is the possibility to fit the experimental spectrum with a simulation obtained using the spin Hamiltonian parameters computed by the ORCA quantum chemistry program system. One of the outputs generated by ORCA is a binary file that contains the atomic coordinates and all calculated spin system properties.



To perform such a simulation with the fitting module the user can press the ORCA button next to the Parameters table. This will prompt an OS window to appear where the binary .prop ORCA output file can be selected. Once the file is selected, Easyspin will load the data from the file and Hyscorean will then display list with all the atoms contained in the ORCA calculation. The indices displayed along each atom in the list are the same indices assigned in the ORCA output files.

The user can choose one or several atoms to be included in the simulation just by selecting them on the list and pressing the OK button. The spin system will be constructed according to the nuclei selected and sent to Easyspin to be simulated with saffron, which will be monitored by a message window displaying: "Simulating ORCA system...".

If the simulation is completed successfully the resulting simulation will be updated in the main display in a similar fashion as in the previous section. Again, to better differentiate the nature of the fit spectrum in the current display, all ORCA simulation results are displayed in a different color coding. For the ORCA data the fitted spectrum is a colormap using orange colors. As in the other cases, the simulation will be processed using the same settings as for the experimental spectrum.

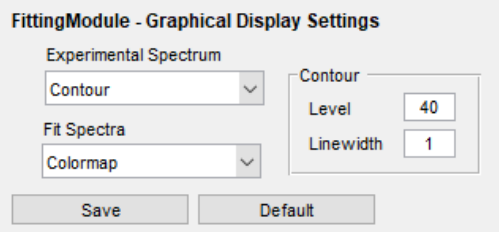


Since the spin system definition and all parameters are given from an outer source, when the simulated spectrum is displayed no changes will be done to the Parameters table. If another fit (either normal fit or manual) is started, the spin system will be taken again from the definition in System and undergo as usual. The ORCA simulation feature is just for display and quick comparison of the ORCA simulations with the experimental spectra.

7.4 Changing the fitting module graphics

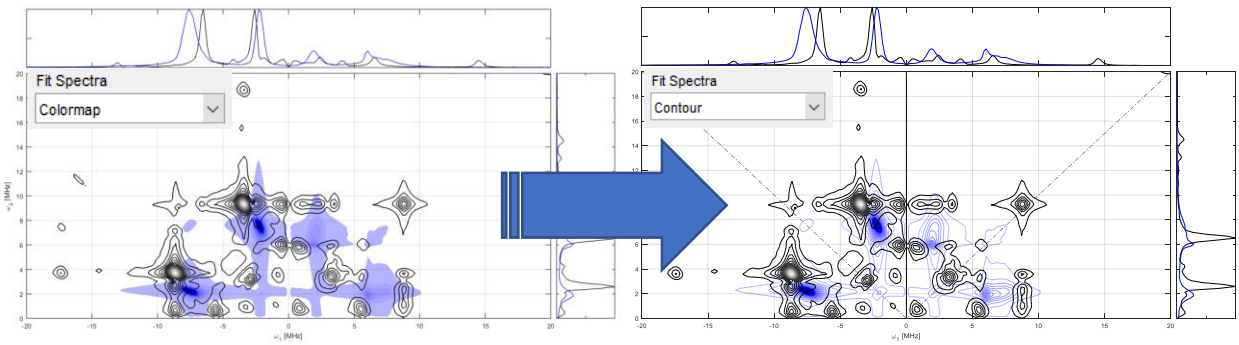
When fitting data there are two main criteria to choose a fit. One of them is the metric chosen to evaluate the fits (e.g. in this case the RMSD values) whereas the other criterion is a purely visual judgment of the fit by the user. Therefore, the way the data are represented can become relevant for some cases where visual judgment may be critical. Because of this, the fitting module of Hyscorean allows to change the graphical settings of the experimental and fitted spectra.

To change the current settings the Graphics button must be pressed, prompting the Fitting Module – Graphical Display Settings window to appear.



There the display type for the experimental and fit spectra can be selected via the Experimental Spectrum and Fit Spectra list boxes. For both the options are either a colormap plot or a contour plot. For contour plots the number of contour levels and their linewidth can be given in the Contour panel.

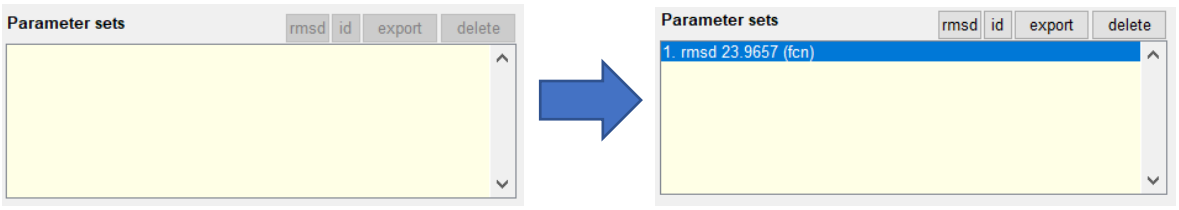
At the start of each session, the graphical settings of the fitting module are set back to the default values displayed above. Once changed the user can set back these values via the Default button. In either case the window can be closed via the Save button. By doing this, all spectra currently displayed will be updated according to the new settings. While the new graphics are rendered the user will be informed via a modal window with the message "Rendering new graphical settings...".



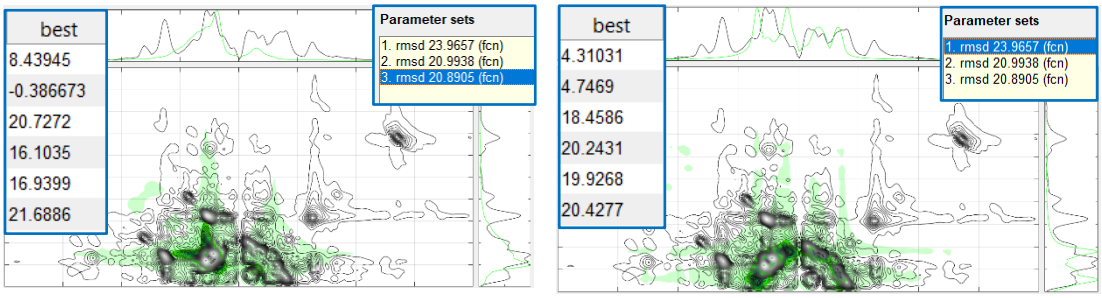
NOTE:
If any of the spectra are being displayed as contour plots the number of contour levels set in the graphical settings can have a dramatic effect on the speed of the fitting. It is recommended thus to keep the contour levels to the minimum required to all the peaks of interest to be displayed.

7.5 Saving the fit results

If the user finds the current fit of the spectrum satisfactory, wants to save the current parameters for later or use them as starting point for another fitting round, the current parameter set under the best column can be stored for later use via the Save parameter set under the Start/Stop button. Doing this will update the Parameter sets panel with the newly saved parameter set. The sets are named according to their total RMSD value.



Once several sets have been saved, these can be displayed in the Parameter table and their corresponding spectrum by just clicking their names in the Parameter sets panel.



The Parameter sets panel also possesses a set of buttons with the same functionalities as in EasySpin:

- rmsd
  Sorts the saved parameter sets by in the Parameter sets panel by increasing RMSD value.
- id
  Sorts the saved parameter sets by in the Parameter sets panel by order of creation.
- delete
  Removes the selected parameter set from the panel.
- export
  Saves the selected parameter set to the MATLAB workspace as a structure variable with name fit# and the following fields:

```
.rmsd       % Total RMSD of the parameter set
.fitSpec    % Cell array of fitted spectra
.expSpec    % Cell array of experimental spectra
.residuals  % Cell array of residuals
.bestValues % Array of the parameter values in the best column
.Sys        % Structure with spin system definition
.ID         % ID of the parameter set
```

Any fit done via the Manual single run method can be saved as well. By doing so, the parameters in the center column (the values used for the manual fit) will be transferred to the best column and saved. Note that once the parameter sets are loaded these will be overwritten in the best and not in the center column. The user must adapt the center values manually.

An exact copy of the current display in the fitting module can at all times be detached via the detach button in the upper-right corner of the display. From the detached figure, the display can be saved and/or exported via the MATLAB interface.

As for the processing, Hyscorean's fitting module includes a report generator to keep an organized record of the fits and their results. The report can be generated by pressing the Report button next to the Parameters table in the upper-right corner of the window.

The report contains the following:

- An exact copy of the input given to EasySpin with the definitions of the Sys and Vary variables.
- A summary of the fitting options employed and a copy of the Parameters table containing the best values of the fit.
- For each fitted spectrum the report includes a copy of the experimental spectrum overlaid with the best fit as in the GUI. Additionally, both the experimental spectrum as the fitted spectrum are displayed separately for a better visualization.
- A summary of the evolution of the total RMSD as well as the RMSDs of the individual fitted spectra.

Once the Report button is pressed an OS window will prompt for the user to select the name and folder under which the report is the be saved.

Hyscorean - EasySpin Fit Report

29-Dec-2018 19:04:33

File(s):

Path(s):

Spin-System Input Definition:

```
%------------------------------------------
% EasySpin Input
%------------------------------------------
% Spin System definition
%------------------------------------------
Sys.Nucs = '15N,14N';
Sys.g = [5.50 , 3.00, 0.01];
Sys.A = [5.50 , 5.50, 18.50; 20.00 , 20.00, 20.00];
% Fit variables definition
%------------------------------------------
Vary.A = [8.00 , 8.00, 5.00; 5.00 , 4.00, 6.00];
Vary.g = [2.00 , 2.00, 0.00];
```

Method	Scaling	Startpoint
Nelder/Mead simplex	scale & shift (min/max)	center of range

Fit Variable Name	Best Fit	Initial guess	Vary
A(1,1)	5.500	5.500	8.000
A(1,2)	5.500	5.500	8.000
A(1,3)	18.500	18.500	5.000
A(2,1)	20.000	20.000	5.000
A(2,2)	20.000	20.000	4.000
A(2,3)	20.000	20.000	6.000
g(1)	5.700	5.500	2.000
g(2)	3.000	3.000	2.000

Figure 1. Total RMSD (Best fit: 26.5368)

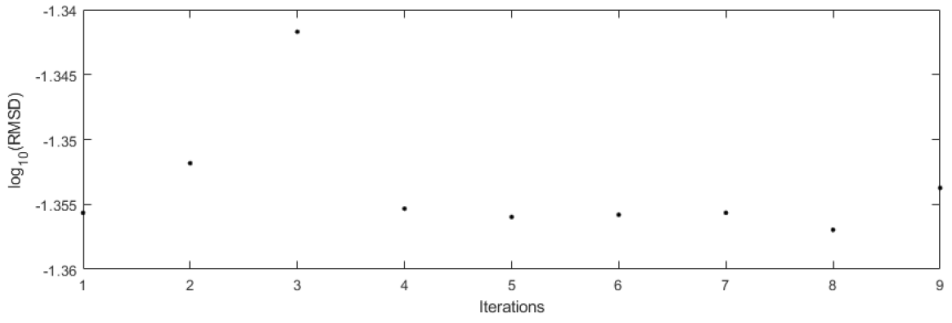
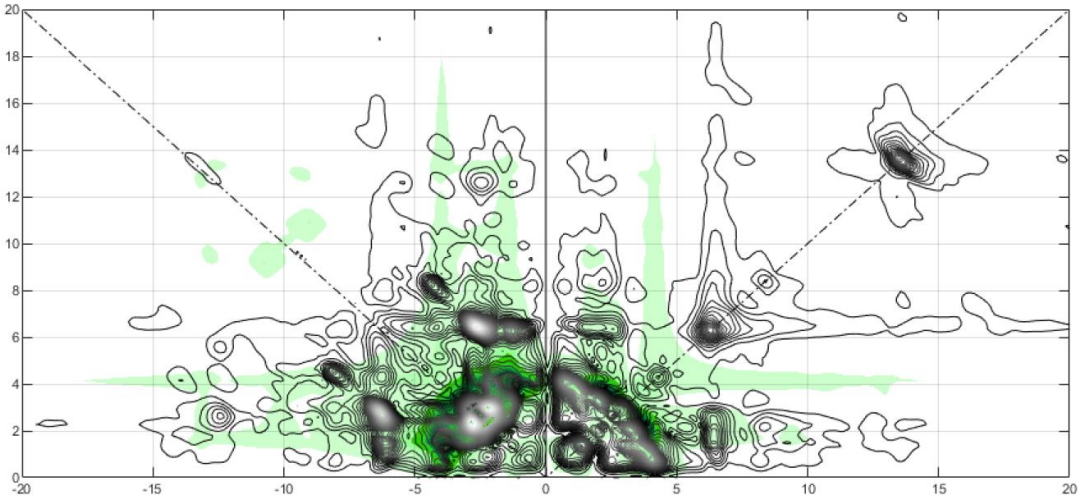
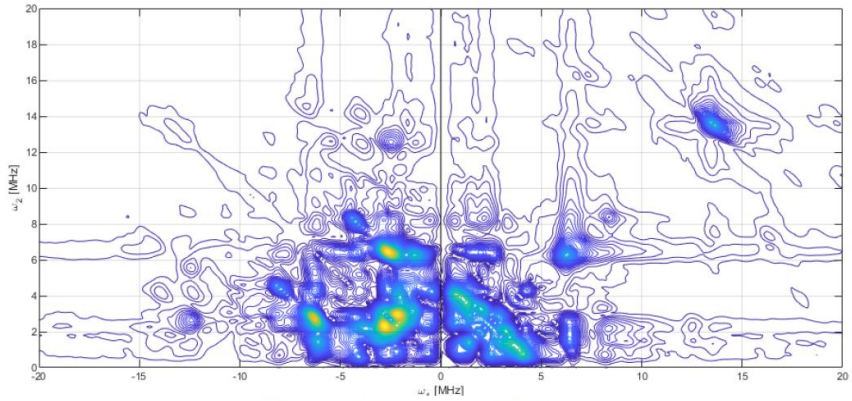
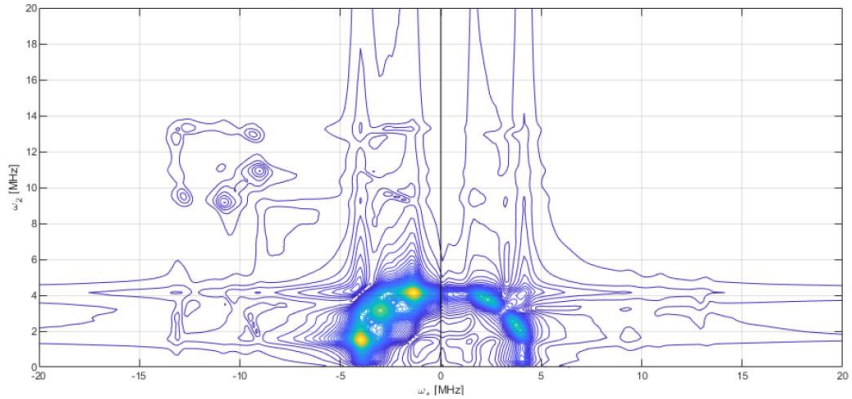


Figure 2. Spectrum @318.20 mT (Best Fit RMSD: 11.6138)



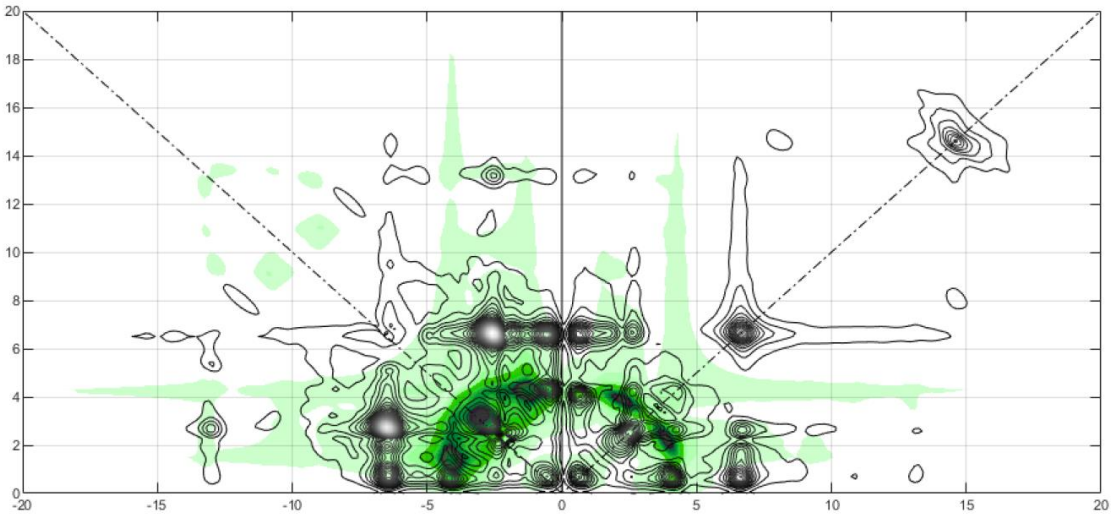


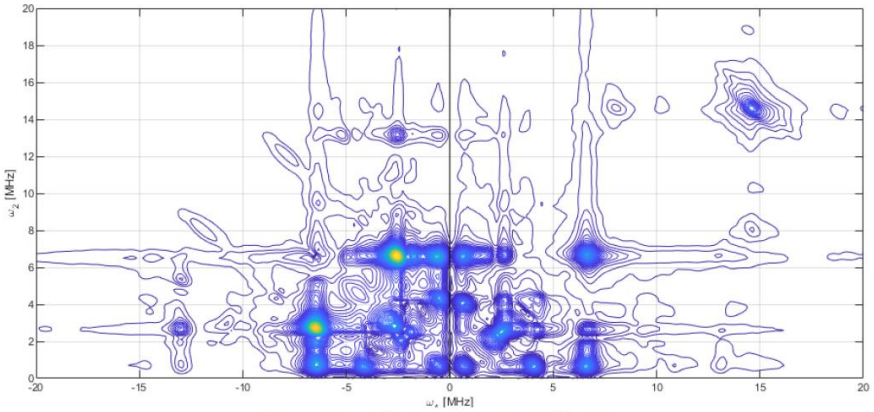
Experimental spectrum @318.20 mT



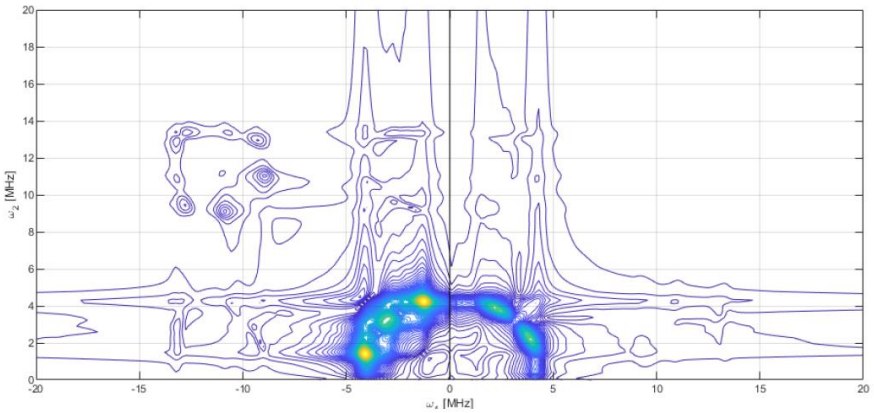
Best fits pectrum @318.20 mT

Figure 3. Spectrum @342.00 mT (Best Fit RMSD: 14.9231)





Experimental spectrum @342.00 mT



Best fits pectrum @342.00 mT

Figure 4. RMSD Evolution of Spectrum @318.20 mT (Best Fit RMSD: 14.9231)

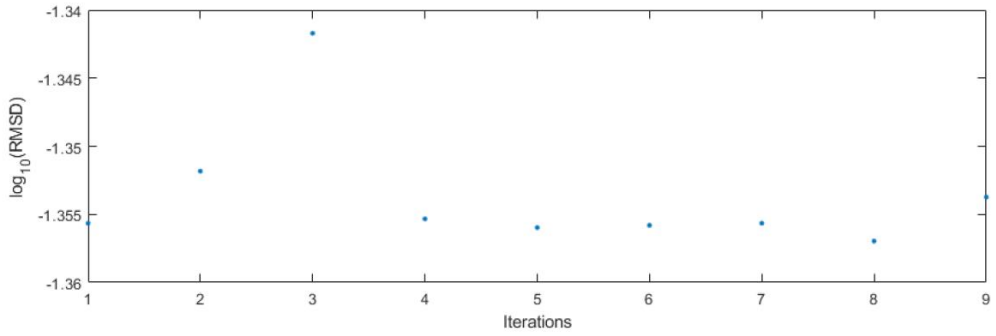
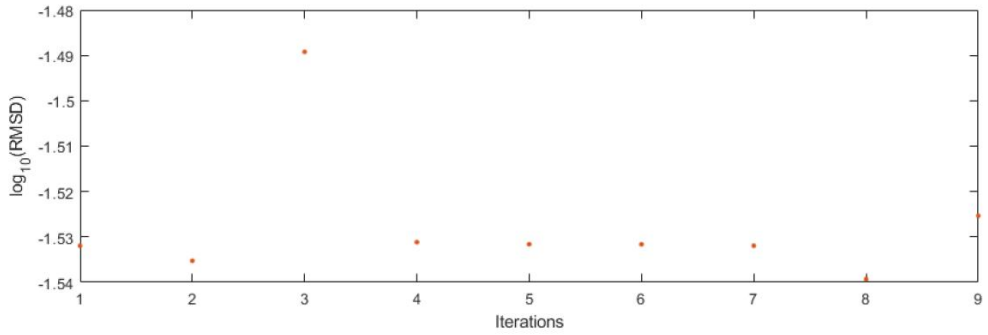


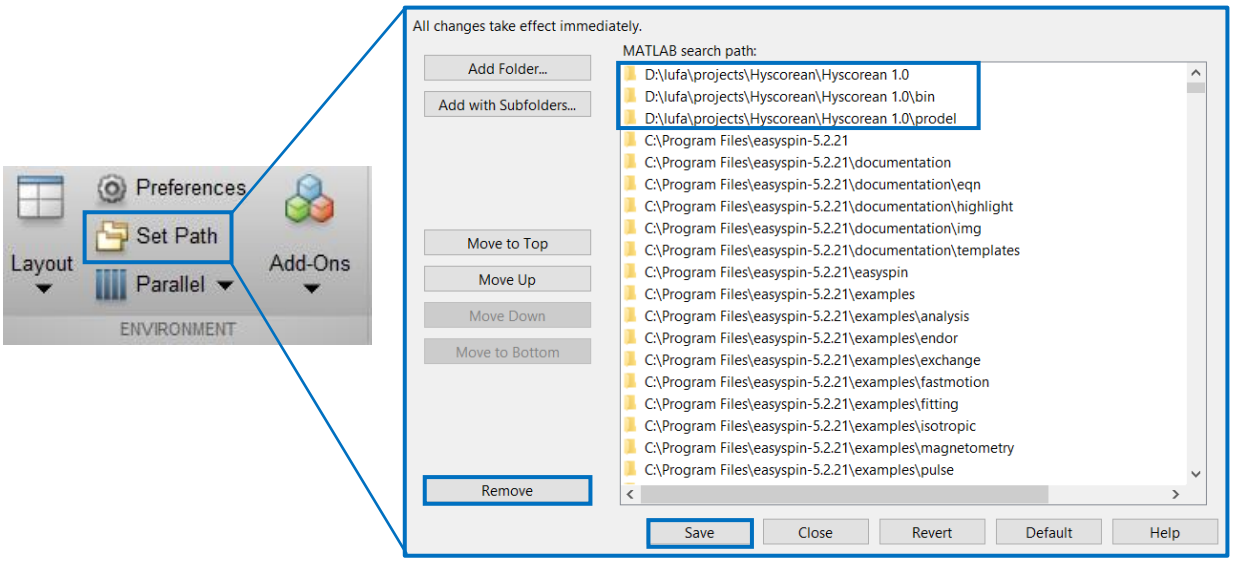
Figure 5. RMSD Evolution of Spectrum @342.00 mT (Best Fit RMSD: 14.9231)



8. Uninstalling Hyscorean

To remove Hyscorean from the computer and from MATLAB the user must follow these instructions:

I. Remove Hyscorean from the MATLAB search path. This can be done via the MATLAB GUI selecting the Home>Environment>Set Path option and then in the new window selecting all Hyscorean paths and removing them via the Remove button. Make sure to exist the window via the Save button to ensure the paths are removed in further session.



II. Remove the MATLAB preferences set by Hyscorean by executing the following command in the MATLAB console.

```
rmpref('hyscorean')
```

III. Remove all files from the computer by deleting them from the location chosen by the user to save them.

At this point Hyscorean will be completely removed from the system as prior to installation. Should the user want to re-install Hyscorean, the installation will have to be done as described in 2.1. at the start of this manual.

9. NUS on Bruker XEPR spectrometers

Hyscorean offers the possibility to comfortably process non-uniform sampled (NUS) HYSCORE experimental data. Nonetheless, the acquisition of such data can be a challenge with the hardware and software commonly available to perform HYSCORE experiments. Of course, home-built spectrometers which run on home-built software are easy to adapt to NUS measurements. Implementation is also commonly easy due to the programming languages this are built upon.

Commercial Bruker spectrometers running on the XEPR software are not able to perform NUS measurements of any kind due to their sweep-experiment nature. Nonetheless, to allow NUS HYSCORE to be applied as widely as able Hyscorean offers an add-on as a possibility to measure NUS HYSCORE in commercial Bruker machines.

In the following sections a description will be provided on how this is achieved, as well as a detailed guide to set up NUS HYSCORE measurements on Bruker spectrometers running on XEPR.

9.1 Non-uniform sampling in XEPR

A common trait of EPR experiments is that some dimension is swept (either by incrementing or decrementing some time interval or frequency) uniformly and for a defined sweep length. Hence, most experiment definitions in XEPR are defined around this concept. Therefore, the software is inherently not prepared or designed to allow non-uniform sampling.

XEPR offers the well-known PulseSPEL language to define custom experiments (although with a limited variable definition set and rather limited buffer memory). This allows the design of custom HYSCORE experiments. A direct implementation of NUS HYSCORE would be a PulseSPEL program which would contain single-point measurements for all of the HYSCORE points to be measured according to some NUS schedule (which could be generated from some other program, e.g. a MATLAB script). This is, however, not possible due to the limited buffer memory of the PulseSPEL programming panel and its compiler.

Another tool offered by XEPR is the Procedure Description Language (ProDeL) which allows a more detailed experiment design with the possibility of performing arithmetic operations. This language is still limited in its flexibility and its functions are non-intuitive making it a difficult programming language to work with. The program still suffers from some buffer memory problems for some variable definitions, but it is not limited in its length.

Considering these tools, the following ideas were considered for NUS HYSCORE implementation:

- Use PulseSPEL to define an experiment which measures a single HYSCORE point and return
- Use ProDeL to set the NUS schedule timings from a loop into the PulseSPEL program and call iteratively. After each PulseSPEL execution store the measured point in a dataset.
- NUS schedules need to be brought in some manner into the software, either by copying them into the ProDeL code or by loading them via the program.
- The output needs to be as compact as possible for the user's comfort and to be compatible with Hyscorean to be processed.

By taking this into consideration NUS HYSCORE experiments in XEPR were implemented via a combination of PulseSPEL and ProDeL programming. Next, a detail description of the implementation will be provided.

First, a PulseSPEL program is defined which takes only single-point measurements. Due to the limited number of variables in PulseSPEL not many delays can be defined. Considering the ideas listed above, one would want to measure a single HYSCORE point and return to the ProDeL program to set the next timings and measure the next point. This is, however, not possible in PulseSPEL due to the language limiting the minimal number of points to be measured to at least four. Therefore, the ProDeL program will have to pass the timings in sets of four.

This program is available from the prodel directory inside Hyscorean under the name NUS_HYSCORE.exp. A fragment of the code is shown in the box on the right. The variable definitions and how to set them will be discussed on section 9.3. The d10-d13 delays represent the t1-timings and the d20-d23 delays represent the t2-timings of the different four points being measured and which will be needing to be set by the PulseSPEL program.

Another issue is the way the timings in ProDeL are obtained from the NUS schedule. Again, a straightforward implementation would be to hardcode the timings directly into the ProDeL program code generated via some script. A more elegant solution would be to be able to load a file containing the NUS schedule and directly read the timings from there. XEPR and ProDeL, however, are extremely limited on the type of data they can read and load. The only type of data ProDeL can read and manipulate are BES3T data formats.

Hyscorean then also offers a MATLAB script named generate_NUS_schedule.m which constructs the NUS schedule and saves it first as a MATLAB variable file and as BES3T data files (see next section for details on the NUS schedule construction). With this, the NUS HYSCORE timings can then be loaded into the ProDeL program and set in the different PulseSPEL program executions.

```
begin exp "NUS HYSCORE" [INTG QUAD]
for k=1 to n
totscans (n)
scansdone (k)
;
x=1
        shot i = 1 to h
            p1[ph0]
            d1
            p1[ph0]
            d2
            d10
            p0 [ph1]
            d3
            d20
            p1 [ph2]
            d1
            d0
            acq [sg1]
        next i
x=2
        shot i = 1 to h
            p1[ph0]
            d1
            p1[ph0]
            d2
            d11
            p0 [ph1]
            d3
            d21
            p1 [ph2]
            d1
            d0
            acq [sg1]
...
(continues for x=3 and x=4)
...
next k
;
```

The next issue is the output of the program. For the later processing in Hyscorean, the output BES3T data files need to contain the measured dataset, the experiment descriptor (where the PulseSPEL experiment and variable definition are stored) as well as the NUS schedule to be able to reconstruct the HYSCORE spectrum correctly. The first two points are not a big issue since they are automatically set as defaults in BES3T data files. However, the passing of the NUS schedule becomes a more difficult

task. It could of course be passed to Hyscorean as an additional file along the measured NUS HYSCORE data file, but this is inconvenient and prone to errors due to loading of the wrong schedule file.

Passing the NUS schedule timings along the data is not an easy task due to the limitations of the ProDeL dataset manipulation possibilities. In the Hyscorean implementation the NUS timings, are encoded in the abscissa for each ordinate point. Given a measured point at t1 and t2, the corresponding abscissa point of the dataset is constructed as follows:

$$X_{absc}(t_1, t_2) = t_2 + \frac{t_1}{10000}$$

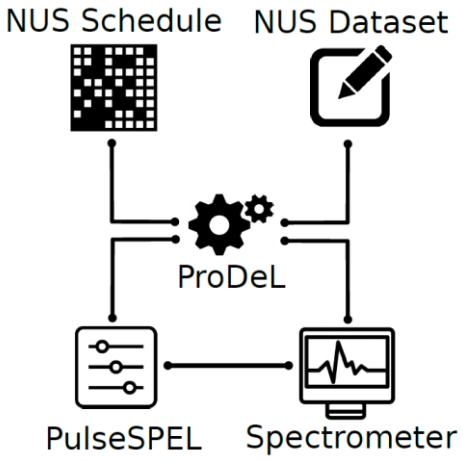
Since the timings in XEPR are defined in ns integers with a minimal resolution of 2ns (4ns for some spectrometers) and no HYSCORE timing can here exceed the 10µs range, all timings can be reconstructed in the processing from the abscissa values of the dataset, e.g.

$$X_{absc} = 182.0016 \quad \rightarrow \quad t_1 = 16\ \text{ns} \quad t_2 = 182\ \text{ns}$$

and avoid the need to load the NUS schedule at a later point. For XEPR to allow non-integer abscissa values, ProDeL sets the output dataset abscissa to the indexed-gauged type.

To also ensure compactness of the output data, all the different tau-values at which the NUS HYSCORE experiment is to be measured can be defined at the beginning of the ProDeL program (see section 9.3.) and then will be successively measured in a manner analogous as described in section 2.4.1. The values used for the experiment are then passed as a comment variable in the BES3T descriptor section of the files.

The NUS HYSCORE ProDeL program then follows this protocol:



I. The tau-values and dimension are defined by the user and printed into the comments variable.
II. The NUS schedule dataset is copied from the Secondary viewport (see section 9.3)
III. The experiment parameters on the Primary viewport (see section 9.3) are loaded into the program
IV. A dummy measurement is run to get a dataset with full descriptors and then emptied for the real measurement.
V. The tau-values loop is started, and current tau-value set into the PulseSPEL program.
VI. The timings in the NUS schedule are updated in sets of four into the PulseSPEL program and the measurements run. The obtained four points are then copied into the real dataset at the corresponding abscissa points.
VII. Once the loops are finished the dataset is copied to the Primary and Secondary viewports where it can be saved from the XEPR GUI.

For a detailed description of the program please refer to the source code of the NUS HYSCORE ProDeL program: ProDeL_NUS_HYSCORE_program.ppp.

NOTE:
This ProDeL program was tested with XEPR version 2.6b.124 (2015) and may work properly with older versions.

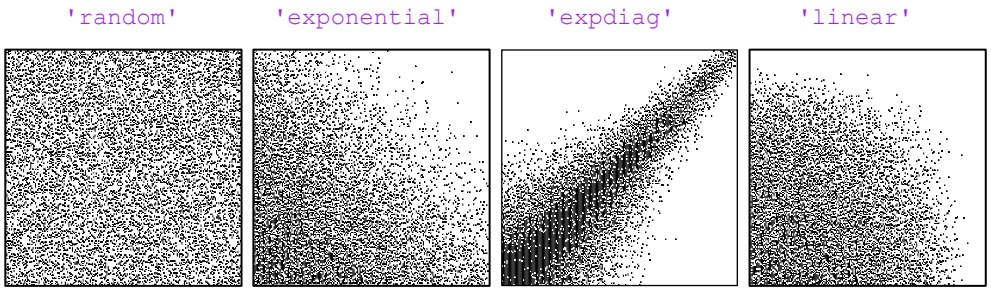
9.2 Generation of the NUS schedule

The NUS schedule defines at which timings the HYSCORE experiment is performed and therefore determines the duration and performance of the experiment. As described in section 4.3. each NUS schedule has an associated point-spread function (PSF) which determines how bad the sampling artifacts will be in the spectrum. Although most of these artifacts can be removed by reconstruction methods, a good PSF will ensure better reconstruction results and later better spectral quality.

Good sampling schemes are those which are based on random sampling schemes. Since no real random numbers can be achieved, pseudo-random number generators are used. Additionally, envelope-matched sampling (EMS) can be used to tailor the randomly generated samples to a given probability density function (PDF) of choice.

Hyscorean offers the MATLAB script generate_NUS_schedule.m which allows the user to generate a NUS schedule and save it in MATLAB and BES3T formats so that they can be used for NUS HYSCORE experiments on Bruker spectrometers. The grids are constructed from a set of parameters given by the user in the script as the following variables:

- TimeStep1/TimeStep2
  Effective dwell time in ns of the NUS grid for each dimension. Minimal resolution of 2 ns (or 4ns for some spectrometers).
- Dimension1/Dimension2
  Size of the grid along the first and second dimension.
- SamplingDensity
  Percentage of points in the grid to be sampled. Must be a number between 0 (no sampling) and 1 (all points sampled).
- Envelope
  Probability-density function to be employed for the envelope-matched sampling. For more details, type help NUS_scheduler.m in the MATLAB console.

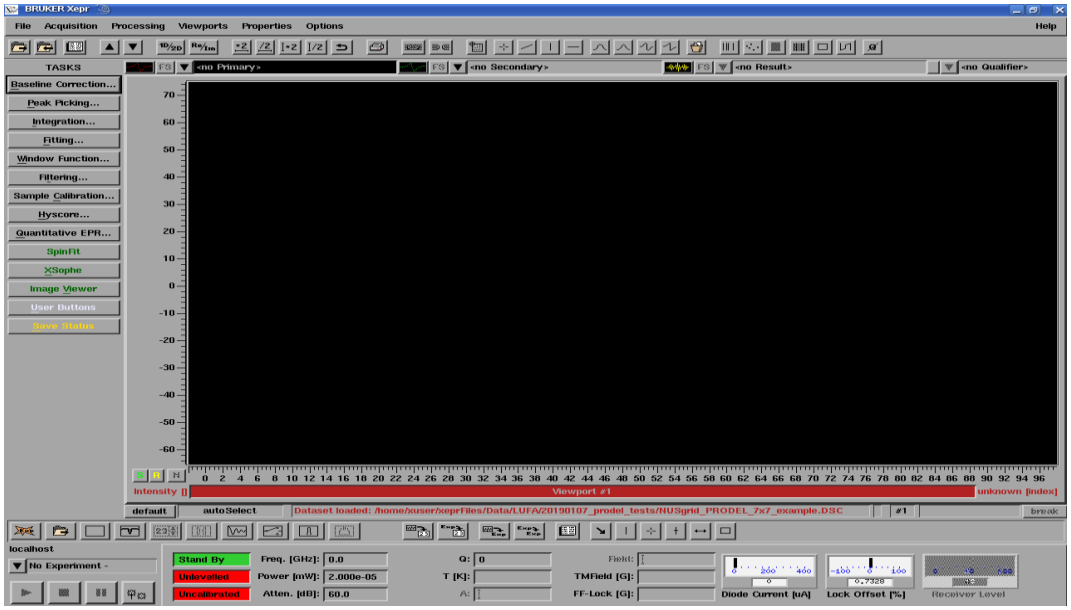


- Decay
  Decay rate of the envelope functions. If empty, is set by default to half of the maximal timing.
- RandomGenerator
  Type of generator to be used for the pseudo-random numbers. Is set by default to `'rand'` which uses the rand function of MATLAB or can be set to `'lhs'` which then uses Latin Hypercube Sampling (LHS) to generate the random numbers.

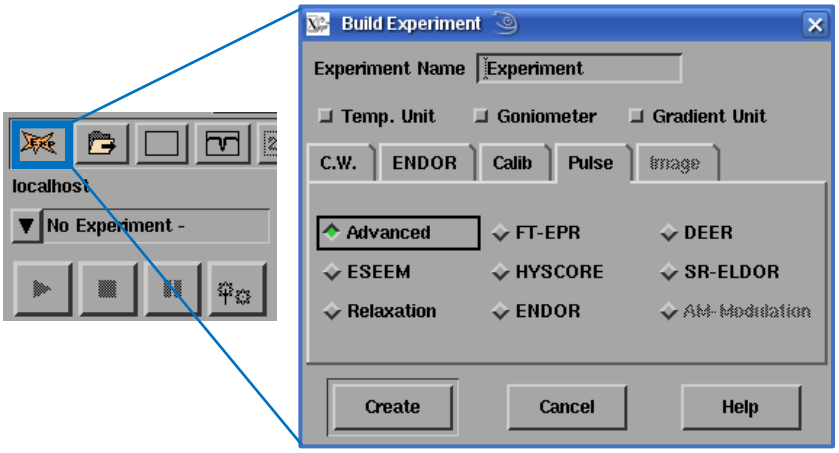
NOTE:
Independently of the settings chosen for the NUS schedule construction, the first and last point of the sampling grid are always measured.

9.3 Setting up the NUS measurement

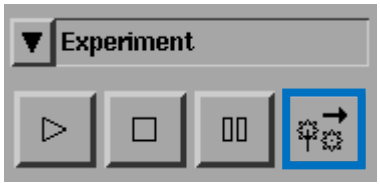
Once the XEPR software is started and the spectrometer connected, the graphical user interface of XEPR should be as follows:



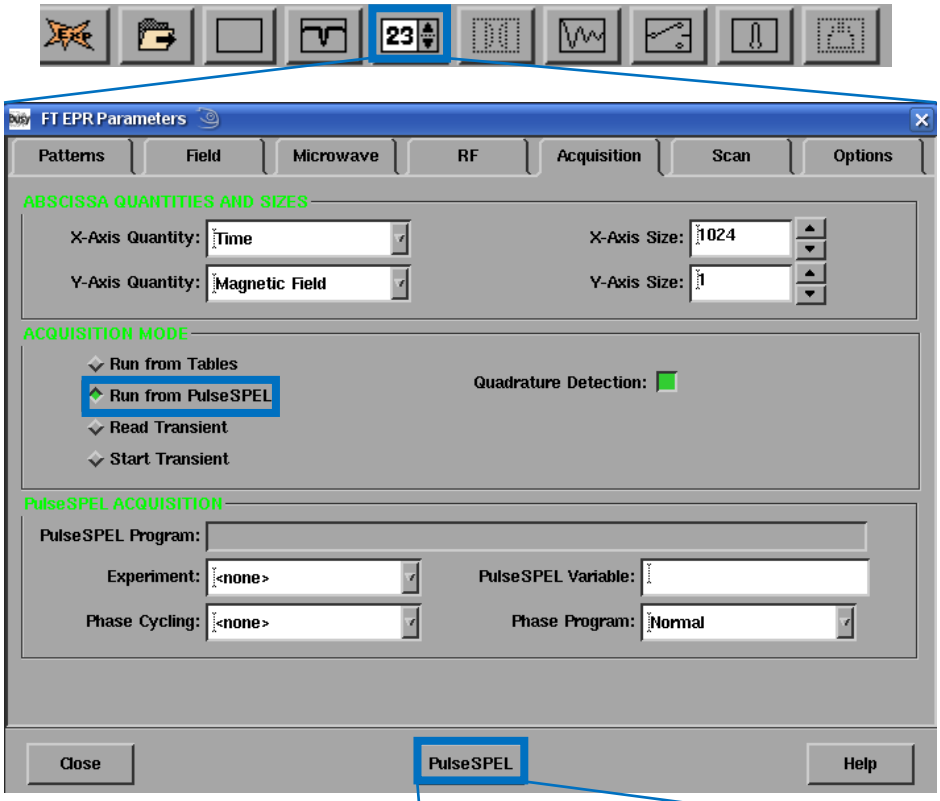
A new experiment must be created via the  button in the bottom taskbar. Pressing the button prompts a new window in which the experiment can be defined. Any experiment identifier can be given via the Experiment Name field by the user. It is required that the experiment is defined as advanced via the Pulse>Advanced option and then generated by pressing the Create button.



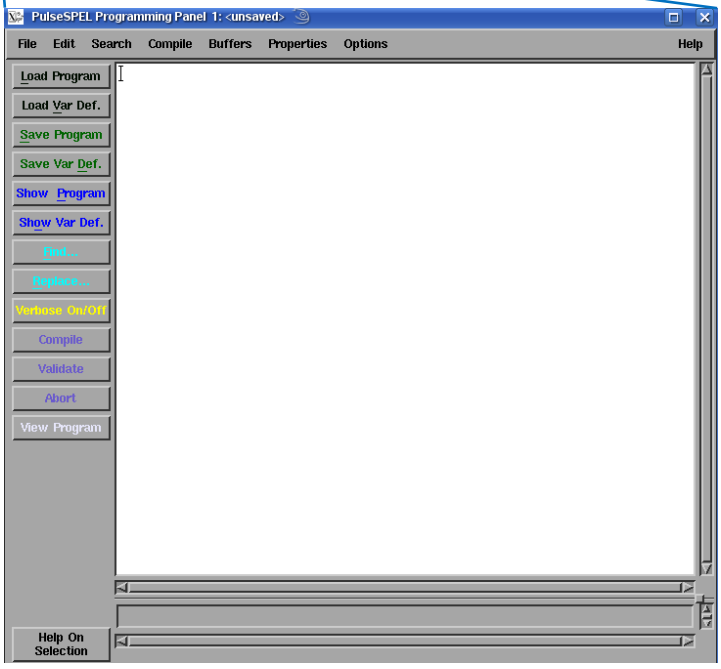
Once the experiment has been created its identifier will appear listed under the toolbar. The experiment parameters need to be passed to the spectrometer via the following button:



Next, the experiment acquisition must be set so that it runs via a PulseSPEL program instead of the default tables. To set this press the FT EPR Parameters button again in the bottom toolbar to prompt a new window. There the option under Acquisition>Run from PulseSPEL needs to be selected.



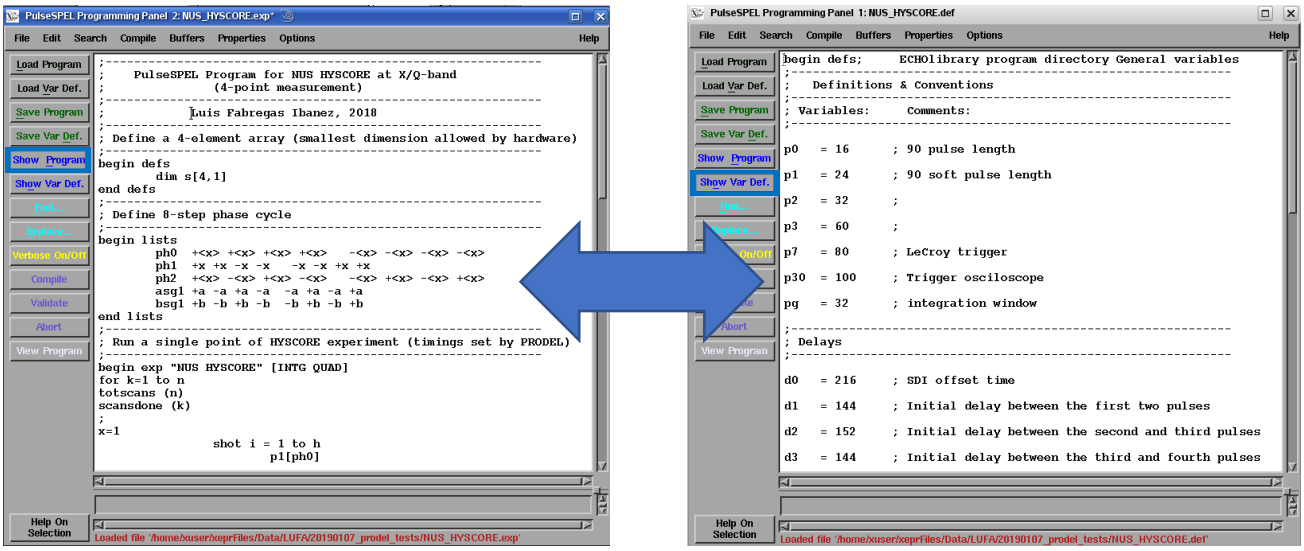
Next the PulseSPEL program and variable definition files need to be loaded. For this select the PulseSPEL button at the bottom of the current window to prompt the PulseSPEL Programming Panel. There load the .exp and .def files via the buttons Load Program and Load Var Def.



NOTE:
The default names for the PulseSPEL files included in the Hyscorean package are:

NUS_HYSCORE.exp
NUS_HYSCORE.def.

At any point the file displayed can be changed via the Show Program and Show Var Def. buttons.



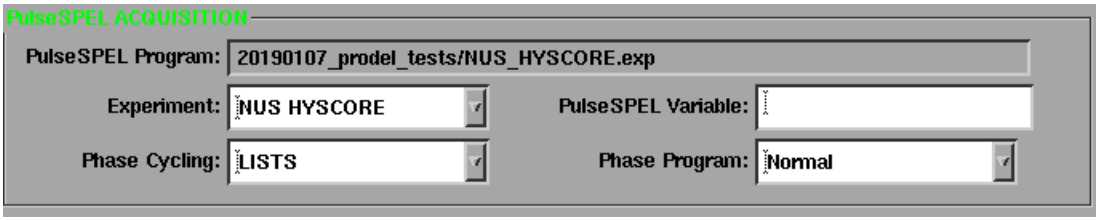
The user does not need to make any adjustments to the PulseSPEL program and only needs to compile it via the Compile button. Any changes (besides the phase cycling definition) in this program are discouraged since the ProDeL program may later not be able to execute properly. If compilation is successful, the following message will be prompted:

```
-->  Second pass ended.
```

The user can modify the variable definition file to set the experimental parameters. All parameters except for d1, d10-d13 and d20-d23 can be freely modified. These parameters are an exception since these will be later be automatically set and modified by the ProDeL program and therefore the given values here have no influence on the experiment later. Once the parameters have been set, the file must be compiled again via the Compile button. If compilation is successful, the following message will be prompted:

```
-->  The variable values are set up.
```

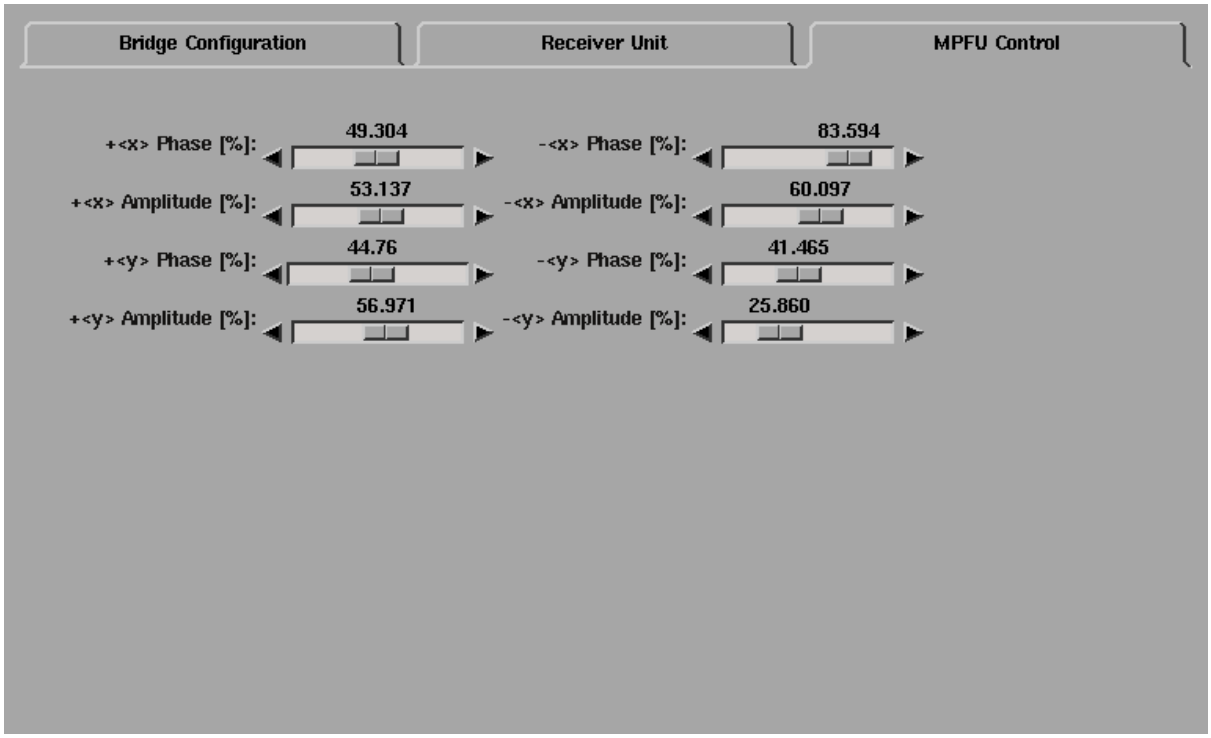
At this point the PulseSPEL Programming Panel can be minimized or closed and the FT EPR Parameters window should be updated as follows:

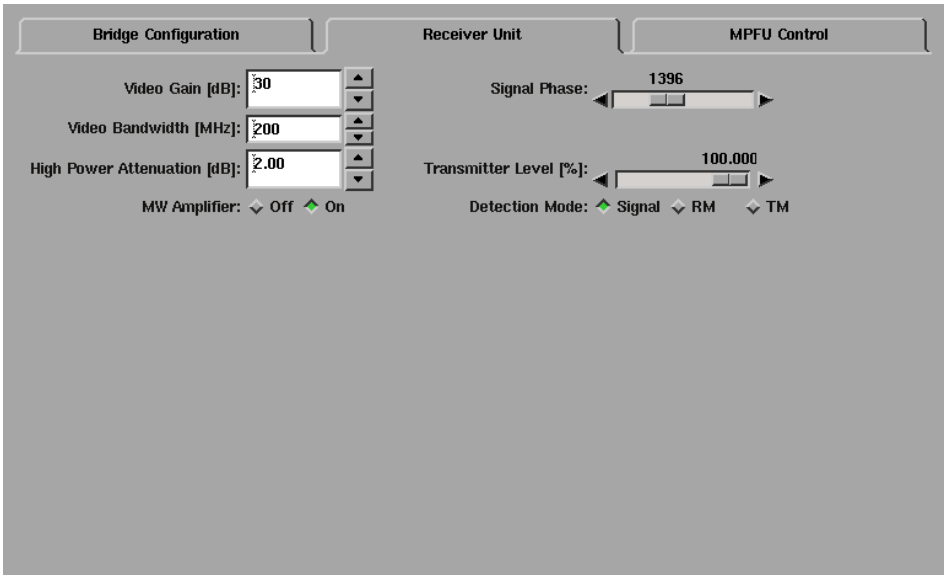


NUS_HYSCOREAN.def

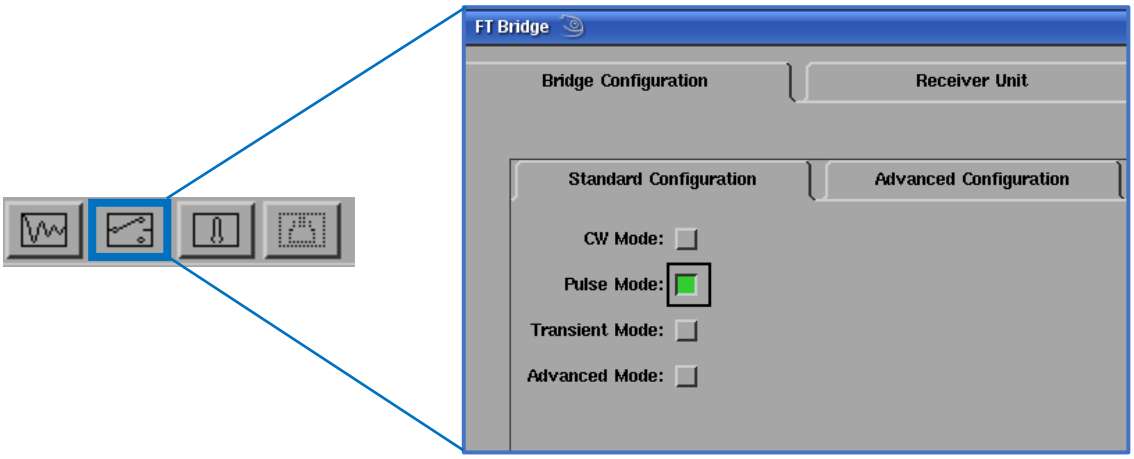
```
;-------------------------------------------------------------
; PulseSPEL Variable Definition for NUS HYSCORE at X/Q-band
;               (4-points measurement)
;-------------------------------------------------------------
;                Luis Fabregas Ibanez, 2018
;-------------------------------------------------------------
begin defs;     ECHOlibrary program directory General variables
;-------------------------------------------------------------
;   Definitions & Conventions
;-------------------------------------------------------------
; Variables:     Comments:
;-------------------------------------------------------------
p0   = 16      ; 180 pulse length
p1   = 24      ; 90 soft pulse length
p7   = 80      ; LeCroy trigger
p30  = 100     ; Trigger osciloscope
;-------------------------------------------------------------
; Delays [ns]
;-------------------------------------------------------------
d0   = 216     ; SDI offset time
d2   = 152     ; Initial delay between the second and third pulses
d3   = 144     ; Initial delay between the third and fourth pulses
d4   = 60      ; Extra delay for the matched pulse
;-------------------------------------------------------------
; DO NOT CHANGE (These are automatically updated)
;-------------------------------------------------------------
d1   = 144     ; Initial delay between the first two pulses
d10  = 16      ; t1-timing of NUS schedule [0]
d20  = 16      ; t2-timing of NUS schedule [0]
d11  = 16      ; t1-timing of NUS schedule [1]
d21  = 16      ; t2-timing of NUS schedule [1]
d12  = 16      ; t1-timing of NUS schedule [2]
d22  = 16      ; t2-timing of NUS schedule [2]
d13  = 16      ; t1-timing of NUS schedule [3]
d23  = 16      ; t2-timing of NUS schedule [3]
y    = 1       ; Second dimension position]
;-------------------------------------------------------------
; Others
;-------------------------------------------------------------
srt  = 500 * srtu ;Minimum shot repetition time
h    = 1024       ;Number of shots/loop ( counter: I )
n    = 1          ;Number of sweeps to accumulate ( counter: K )
t    = 1          ; second time axis sweep length ( counter: Y )
w    = 800        ; Split point of sweeps (ns) ,i.e. pulse separation where TWT
; gate pulse can be split.
;
end defs
```

NOTE:
The pulse channels need to be tuned before the experiment is started. To set the phase of the pulses for the HYSCORE experiment it is recommended to run a single trace of a uniform-sampled HYSCORE experiment to properly set the phase. HYSCORE traces do not appear in the XEPR viewports using the ProDeL program and thus cannot be used for setting the phase.

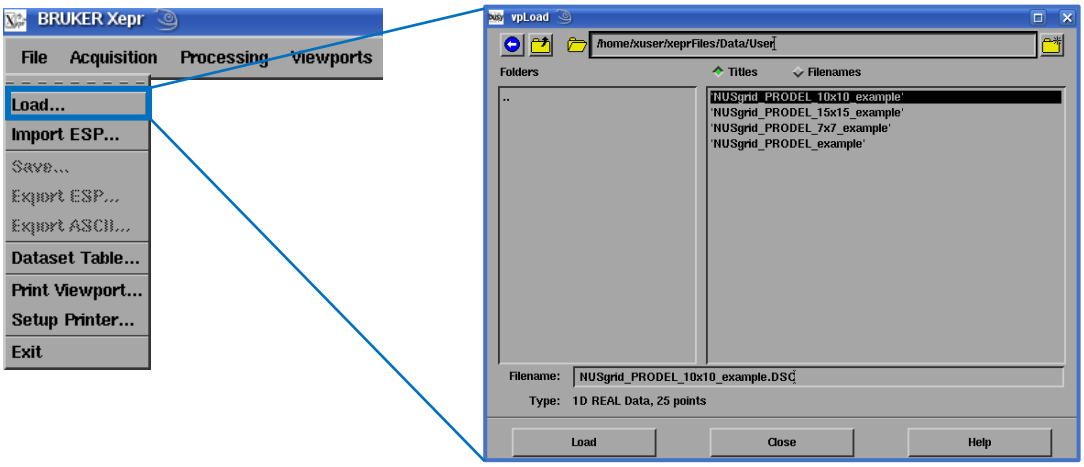

Bridge Configuration
Receiver Unit
MPFU Control
+<x> Phase [%]:
49.304
-<x> Phase [%]:
83.594
+<x> Amplitude [%]:
53.137
-<x> Amplitude [%]:
60.097
+<y> Phase [%]:
44.76
-<y> Phase [%]:
41.465
+<y> Amplitude [%]:
56.971
-<y> Amplitude [%]:
25.860


Bridge Configuration
Receiver Unit
MPFU Control
Video Gain [dB]:
30
Video Bandwidth [MHz]:
200
High Power Attenuation [dB]:
2.00
MW Amplifier:
Off
On
Signal Phase:
1396
Transmitter Level [%]:
100.000
Detection Mode:
Signal
RM
TM

It is important to ensure that the microwave bridge is set to its pulse mode. This can be done via the FT Bridge button on the bottom toolbar and then in the new prompted window via Bridge Configuration>Standard Configuration>Pulse Mode.



The next step is to load the NUS schedule generated in section 9.3. to the XEPR software. This is most easily done via the File>Load… button in the XEPR toolbar. There the DSC file containing the NUS schedule of choice can be loaded.

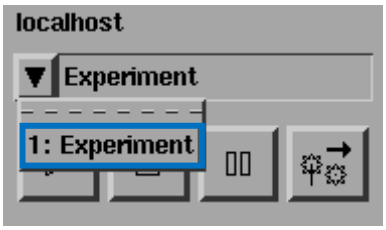


If the NUS schedule is loaded successfully the corresponding dataset will be displayed in the primary viewport with the same name as the corresponding loaded file:

NOTE:
Recall that the t_1 timings of the NUS schedule are encoded in the abscissa of the dataset while the t_2 timings are encoded in the ordinate of the dataset. Therefore, the displayed data on the primary viewport may appear odd at first. This display can be ignored since it does not represent real data.

Next it is important to remove this dataset from the Primary viewport via the ▼ button and selecting the <no Primary> option. Following this the NUS schedule dataset must be set into the Secondary viewport again via the corresponding ▼ button and selecting the same name as previously displayed in the Primary viewport.

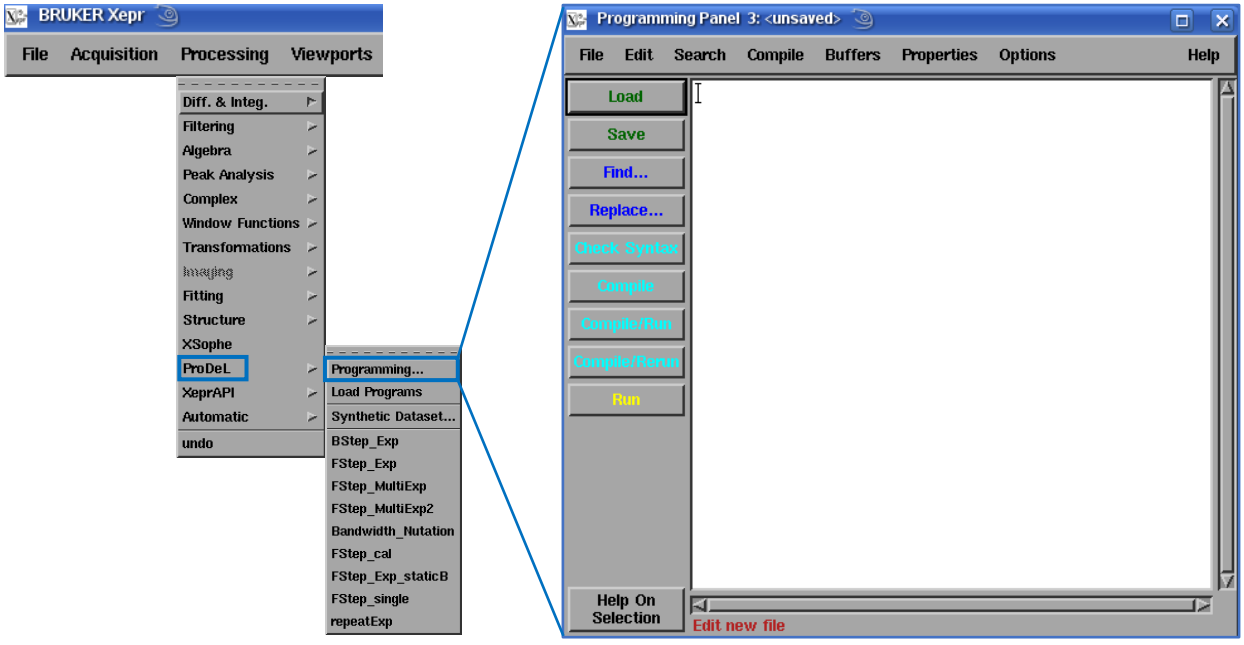
At this point the experiment generated at the beginning of this section must be set as the current dataset of the Primary viewport. This is accomplished by pressing the ▼ button of the experiment list and selecting the name given to the advanced experiment at the beginning of this section.



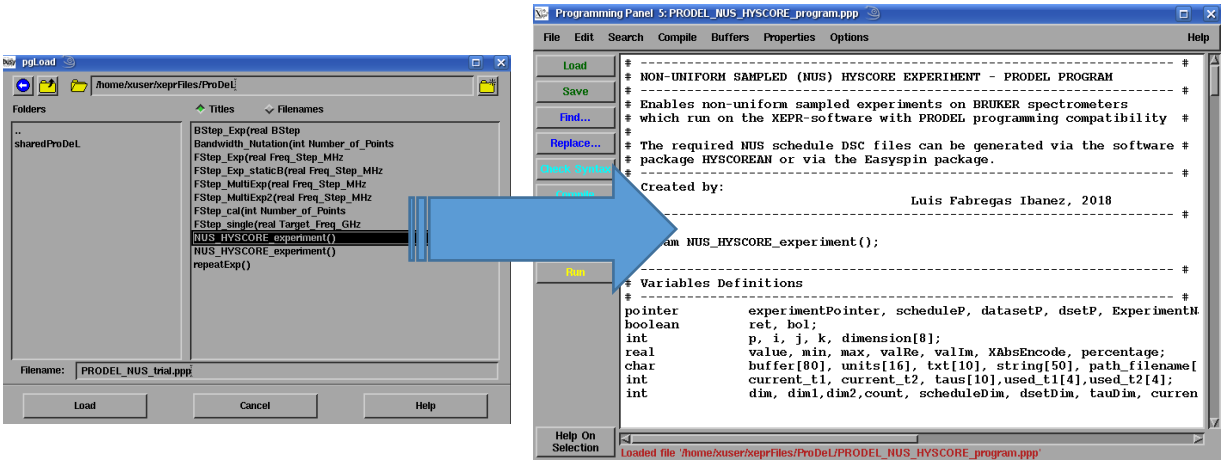
If all steps have been followed carefully the viewport status bar should now look as follows:

NOTE:
It is critical that the viewports are not modified at this point until the ProDeL program has been executed. If the datasets on the Primary or Secondary viewports are changed, the NUS HYSCORE measurement will not be successful and an error will be prompted by ProDeL.

The next step is to open the ProDeL program which will perform the NUS HYSCORE measurement. The Programming Panel can be opened by selecting the Processing>ProDeL>Programming… button.



The program source code can then be loaded via the Load button and then selecting the ProDeL program file from its location on the computer either via its file name PRODEL_NUS_HYSCORE_program.ppp or via its title NUS_HYSCORE_experiment (by default the file can be located at the Hyscorean's location under the prodel directory).



NOTE:
It is recommended to copy the ProDeL file from its original location to the xeprFiles/ProDeL directory since this will always be opened as a default directory for ProDeL files, making the loading more comfortable.

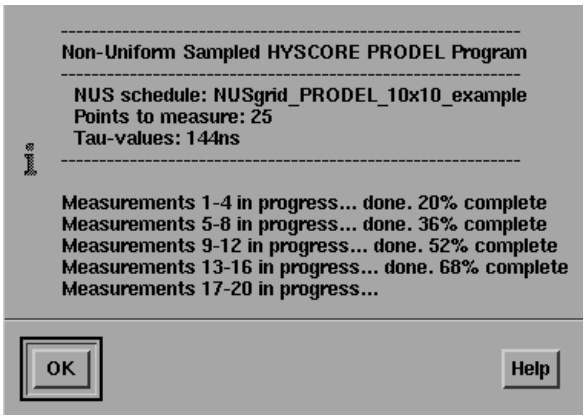
The user now only needs to set a last parameter in the ProDeL program under the User Input section.

```
# ------------------------------------------------------------------------#
# User Input                                                              #
# ------------------------------------------------------------------------#
# Tau values to be measured   [ns] #
# (NOTE: In PRODEL language arrays have to be filled element by element #
#        starting with index 0. Ex/ taus[0] = 100; taus[1] = 120;...  ) #
taus[0] = 144;
taus[1] = 155;
# Number of taus to be measured #
tauDim = 2;
# ------------------------------------------------------------------------#
```

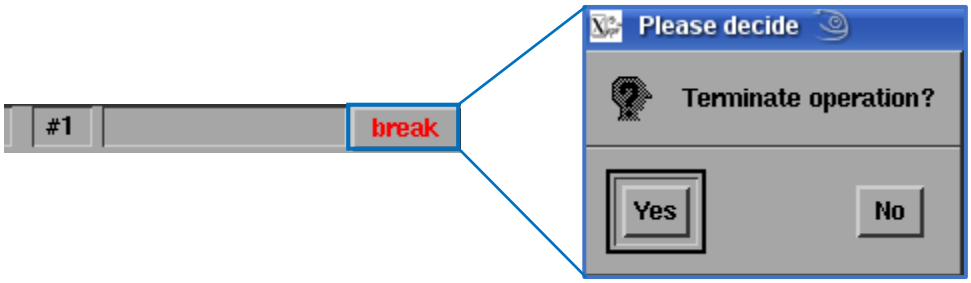
There the user can introduce the different τ-values (in ns) at which the HYSCORE experiments must be measured. The taus array elements must be defined one by one as allowed by ProDeL. At the end the user needs to specify how many tauDim of these τ-values need to be measured and the program will take the tauDim first τ-values in the array. After this no more input is required by the user and otherwise nothing else in the program should be modified. Any issued arising from modifications will be under the user's responsibility.

The program can then be compiled via the Compile button and then start the program via the Run button. Alternatively, both steps can be executed via the Compile/Run button.

Once the ProDeL program starts a message window will prompt where the output of the program will be display. The program will start informing the user which NUS schedule was loaded into the program, how many HYSCORE points are to be measured from the NUS grid and the tau-values to be employed.



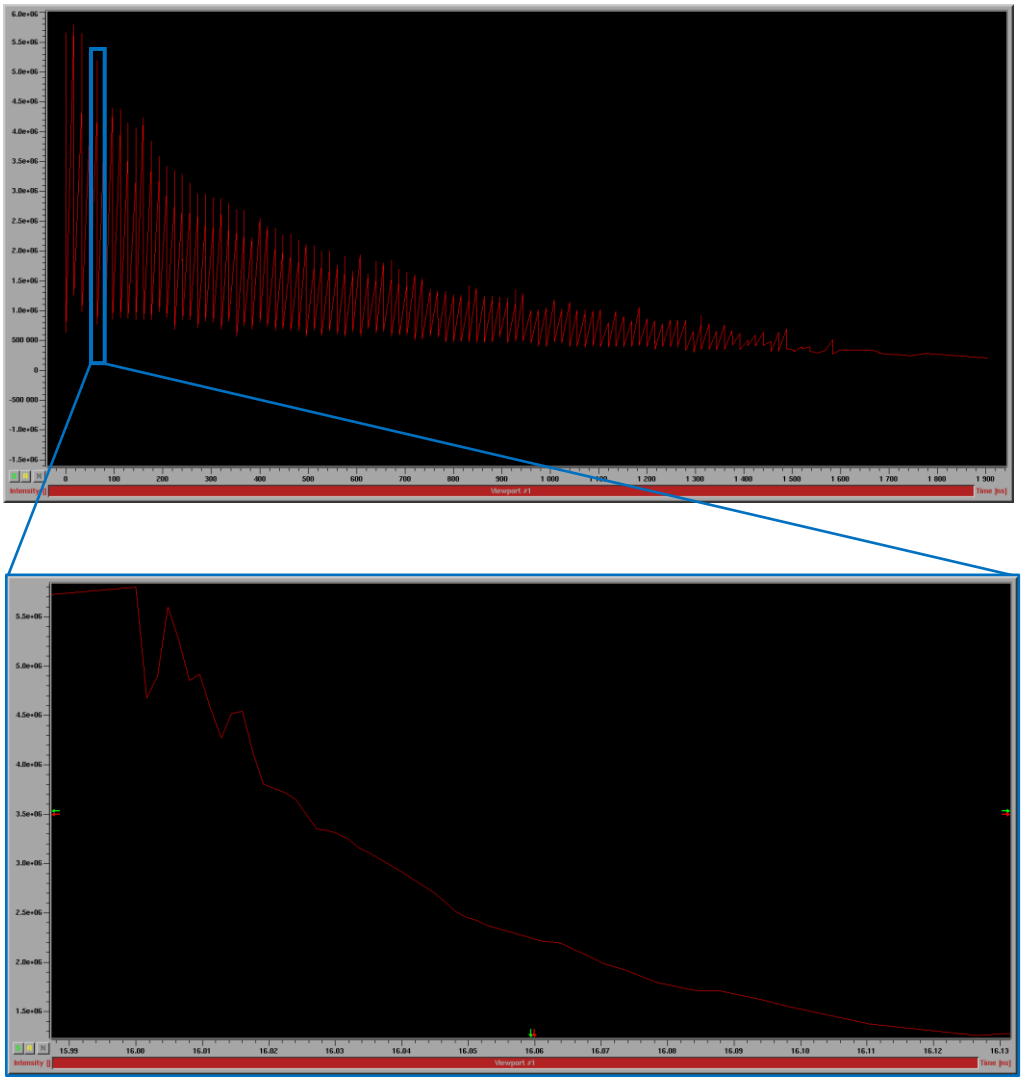
During the execution of ProDeL programs, the XEPR GUI is completely blocked from any user input until the program finishes execution. Nonetheless, the program can be killed at any moment via the break button on the bottom-right side of the XEPR main window. This button will pause the execution of the ProDeL program and ask the user whether the execution is to be terminated.



NOTE:
Terminating the program will stop its execution. However, if a measurement is running, terminating the program will not stop the current measurement. The user can wait for the measurement to finish or to stop if via the now active Stop button.

Once the NUS HYSCORE measurement finished and the ProDeL execution reached the end, the Primary and Secondary viewports will be set with the complete NUS HYSCORE dataset generated by the ProDeL program under the name NUS HYSCORE Experiment. The data are saved in both viewports to ensure that the data is not lost should one viewport be overwritten by accident.

In the data as displayed in the viewports represent the t2-timings measured during the NUS experiment, which may look like vertical lines. If one zooms-in into one of these vertical lines, the measured t1-timings corresponding to that t2-timing can be seen.



The user now must save/export the NUS HYSCORE dataset via the File>Save… or File>Export ASCII… buttons on the main toolbar from either the Primary or Secondary viewports. The files can then be loaded into Hyscorean and processed as described in section 4.

10. GNU LGPL 3.0 License

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function

or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and

Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

11. References

[1] G. Jeschke, *Signal processing in spectroscopy,* ETH Zurich: Lecture notes, 2017.

[2] R. R. Ernst, "Advances in Magnetic Resonance," *Academic Press,* vol. 2, p. 1, 1966.

[3] A. Ferige and J. Lindon, "Resolution Enhancement in FT NMR Through the Use of a Double Exponential Function," *Journal Of Magnetic Resonance,* vol. 31, pp. 337-340, 1978.

[4] C. S. Lessard, "Window Functions and Spectral Leakage," in *Signal Processing of Random Physiological Signals*, 2006, pp. 175-193.

[5] S. Pribitzer, L. Fábregas Ibáñez, C. Gemeiner, I. Ritsch, D. Klose and G. Jeschke, "Two-Dimensional Distance Correlation Maps from Pulsed Triple Electron Resonance (TRIER) on Proteins with Three Paramagnetic Centers," *Applied Magnetic Resonance,* vol. 49, no. 11, pp. 1253-1279, 2018.

[6] Stoll et al. , *Journal Magnetic Resonance,* vol. 177, pp. 91-101, 2009.

[7] Maciejewski et al., "Nonuniform Sampling and Spectral Aliasing," *Journal of Magnetic Resonance,* vol. 199, no. 1, pp. 88-93, 2009.

[8] Shchukina et al., "Pitfalls in compressed sensing reconstruction and how to avoid them," *J. Biomol. NMR,* vol. 68, pp. 79-98, 2017.

[9] B. Worley, "Convex accelerated maximum entropy reconstruction," *Journal of Magnetic Resonance,* vol. 265, pp. 90-98, 2016.

[10] Balsgart et al., "Fast Forward Maximum entropy reconstruction of sparsely sampled data," *Journal of Magnetic Resonance,* vol. 223, pp. 164-169, 2012.

[11] Wikipedia, "68–95–99.7 rule," [Online]. [Accessed 31 1 2019].

[12] A. S. Stefan Stoll, "EasySpin, a comprehensive software package for spectral simulation and analysis in EPR," *Journal of Magnetic Resonance 178 (2006) 42–55,* vol. 178, pp. 42-55, 2006.

[13] Stoll et al., " EasySpin Documentation: Fitting EPR spectra," [Online]. [Accessed 12 2018].