OctoView

The reconstruction and viewer software for MPI, MPS and COMPASS

Document-Version 1.9 01/2026

This software tool is for the management, processing, and visualization of MPI data. The graphical user interface (GUI) and the data structure are designed to help the user manage multiple projects from simulations and/or measurements with different scanner parameters as well as different reconstruction parameters easily in one package.

Additional documentation:

- MPI-Systemmatrices — provides specific information about system matrices: from generation and manipulation to reconstruction
- Datatypes — provides specific information about the datatypes used in OctoView and Magnetic Field Simulator (MFS)
- TWMPI Sequences — provides specific information about encoding schemes, trajectories and sequence adjustments (frequencies, phases, etc.)

Author: Dr. rer. nat. Patrick Vogel

eMail: Patrick.Vogel@phase-vision.com

Table of content

1. The Graphical User Interface (GUI)

The GUI of this software consists of multiple windows for the control of scanner and reconstruction parameters. In addition, several helper windows supporting the data processing and visualization are available.

Main window

The main window is divided into two main areas:

On the left side (Left bar), the projects and sub-categories are listed with access to additional sub-categories:

- **scanners/projects** list of different scanners (real or simulation) or projects with different parameters
- **sequences** for each project, different sequences (e.g. 2D projection or 3D) can be implemented and set up
- **experiments** list and manages all raw data files generated with the selected scanner and sequence
- **recos** manages different types of reconstruction methods on the selected data

On the center, for each category a separated sub-page consists multiple parameters settings, visual boxes and information (Figure 1-1 shows for example the experiment view for a selected scanner [2024_TWMPI_V1] and a selected sequence [type0_exp]).

Optional: On the right (Right bar), the scanner control panel provides rapid overview about scanner parameters in real-time and allows a full scanner control for running experiments.

Figure 1-1: Screenshot of the main window of the software. ***Left:*** *overview of the projects and their sub-categories.* ***Center:*** *For each category, a separate page is provided with additional parameters.* ***Right:*** *scanner control panel with access to scanner parameters and experiment control.*

Project overview

This area manages multiple scanners or scanner-projects.

A ‘known’ scanner-project can be loaded by selection of the drop-down menu at the top.
ADDing, ***DUPlicating***, ***DELeting***, ***IMPORTting*** or ***BACKUP*** scanner projects are the major functions here.

For each scanner project, multiple sequences can be added, e.g., to run the scanner with different trajectories or frequencies, etc.

For each sequence, multiple experiment-folders can be created to store and handle different experiments and data sets.

For each sequence, multiple types of reconstruction methods can be created and managed.

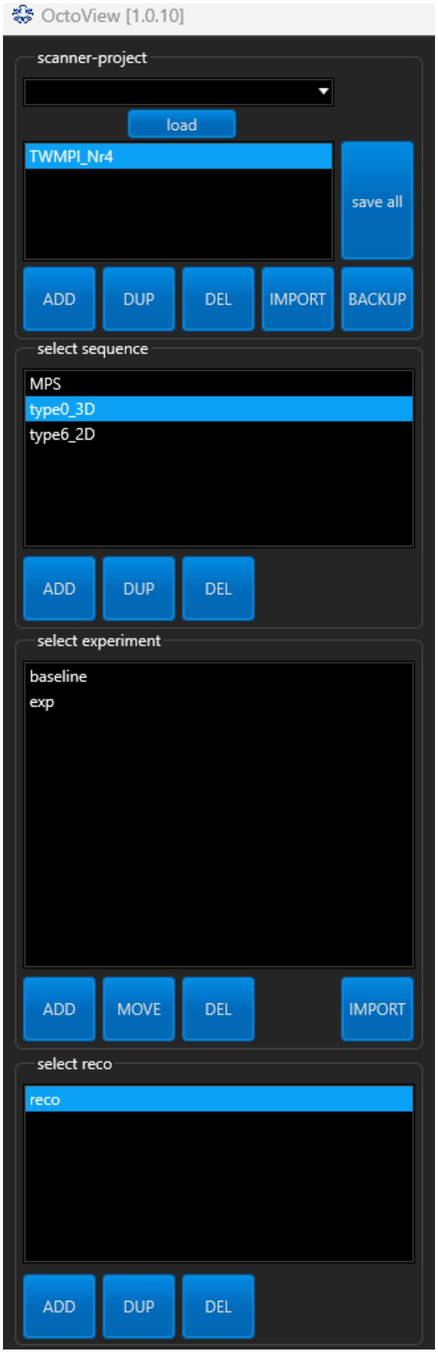


*Figure 1-2**:** Project overview: this area manages multiple types of scanners and their sequences, experiments as well as reconstructions.*

*Hint 1-1: By **double-clicking** on the desired scanner-project or sequence or experiment or reconstruction, the software opens the corresponding area in the main area of the window.*

Main area

The main area consists of multiple tabs with different areas providing additional layers for parameter settings. E.g. the data handling in the scanner-project tab, the frequency setups in the sequence tab, the data set management in the experiment tab, different reconstruction methods in the reconstruction tab, I/O management in the pipeline tab and global settings in the settings tab.

Figure 1-3: Overview of the tab area of the main area of the main Octoview window.

Start-tab

The **"select scanner or project"** area provides a selection of pre-defined setups for scanner/spectrometer types (TWMPI, COMPASS, MPS).
Last projects can be selected and loaded by pressing "GO".

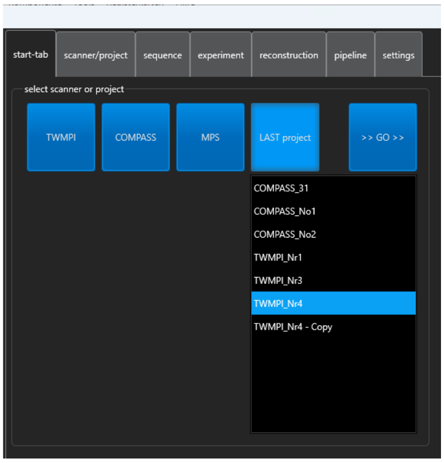


*Figure 1-4: Overview of the start tab: **Select experiment** provides different pre-defined setups for scanners and offers fast loading for last projects.*

Scanner/project tab

The scanner/project tab provides all necessary settings for a scanner (real or simulation).

- Scanner name [project]: sets the scanner/project name
- System name [server]: sets the name for the server: each real scanner comes with pre-calibrated parameters, which are hard-coded on server's site. ATTENTION: be careful! Changing the system's name can destroy the scanners' hardware (ask the support).
- Scanner type: defining the scanner type: ***{not defined; TWMPI; COMPASS; MPS}***
- Scanner directory: the folder, the data can be found (see Project structure and workflow)
- Scanner info: text-file for comments and descriptions regarding the scanner/project
- RCC (receive chain correction): this area manages the RCC data set required for correcting the receive chain (see Receive chain correction (RCC)). The receive chain data can be visualized in the graph below.
- System matrices (**TWMPI**): manages all system matrices for the project (see **System matrices (SM)**).
- FOV parameters (**TWMPI**): for 3D visualization:
  - setting the diameter and length of the scanners field of view [mm]
  - setting the angle {Z-angle} [DEG] and position {X; Y; Z} [mm] of the volume rendering

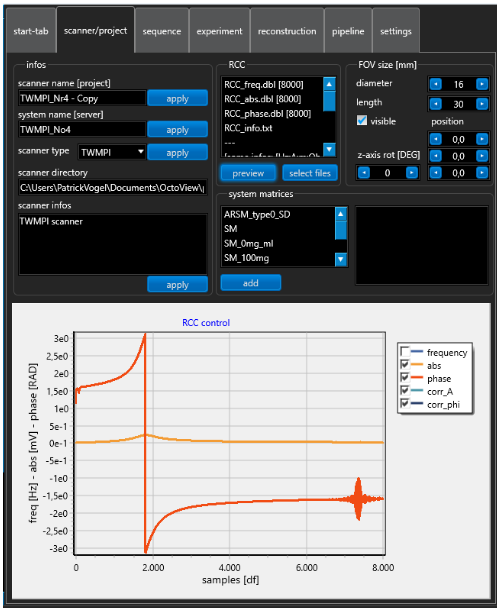


Figure 1-5: Overview of the scanner/project tab.

Sequence tab

The sequence tab handles the parameters for each sequence depending on the scanner type defined in the scanner/project tab. See Figure 1-6 for more details.

Common settings

- Sequence name: set the name of the sequence
- Sequence infos: description and notes for the sequence
- Data settings: sets the global sampling rate (*SR*) and datalength (*dl*) for running the scanner. The corresponding frequency resolution (*df*) is calculated automatically.
  Hint 1-2: these parameters are device specific.
- Scanner setup: -forces a parameter update from the scanner-server.
  -activates preview for data chart (show data) and sequence chart (show sequ) from server-site.

Figure 1-6: Overview of the sequence tab: Left: the TWMPI settings. Right: the COMPASS settings.

TWMPI related settings

- Frequencies, amplitudes and phase: sets the waveform parameters for the TWMPI scanner following the formula (Table 2-1). The frequency resolution is limited through the *df* parameter from (data settings).
  Pressing "apply"-button programs the scanner.
  The "draw"-button visualizes the waveforms in the graph at the bottom.
  Hint 1-3: be careful adjusting these parameters since the scanner can be damaged.
- Sequence type: for the TWMPI scanner, several pre-defined trajectory templates are available, each providing a specific frequency scheme (see Figure 2-5). These schemes are also used to provide sequence specific formulas required for reconstruction (see chapter 3: From MPI signal to the image).
  $f_{rot} = f_1$ $f_{travel} = f_2$ $f_{swipe} = f_3$
  $phi_{rot} = phi_1$ $phi_{travel} = phi_2$ $phi_{swipe} = phi_3$
  The formula is in a software specific format. Multiple independent formulas can be concatenated with ";".
  - Type0: 3D sequence
    $f_{rot} < f_{travel} < f_{swipe}$
    formula f_3*(2*x-1)+f_2*2*y+f_1*2*z;f_3*2*x+f_2*(2*y-1)+f_1*(2*z-1)
  - Type1: 3D sequence
    $f_{rot} < f_{swipe} < f_{travel}$
    formula f_3*(2*x-1)+f_2*2*y+f_1*(2*z-1);f_3*2*x+f_2*(2*y-1)+f_1*2*z
  - Type2: 3D sequence
    $f_{travel} < f_{rot} < f_{swipe}$
    formula f_3*(2*x-1)+f_2*2*y+f_1*2*z;f_3*2*x+f_2*(2*y-1)+f_1*(2*z-1)
  - Type3: 3D sequence
    $f_{travel} < f_{swipe} < f_{rot}$
    formula f_3*(2*x-1)+f_2*(2*y-1)+f_1*2*z;f_3*2*x+f_2*2*y+f_1*(2*z-1)
  - Type4: 3D sequence
    $f_{swipe} < f_{rot} < f_{travel}$
    formula f_3*(2*x-1)+f_2*2*y+f_1*(2*z-1);f_3*2*x+f_2*(2*y-1)+f_1*2*z
  - Type5: 3D sequence
    $f_{swipe} < f_{travel} < f_{rot}$
    formula f_3*(2*x-1)+f_2*(2*y-1)+f_1*2*z;f_3*2*x+f_2*2*y+f_1*(2*z-1)
  - Type6: 2D sequence
    y-z-plane: f_{rot}=0 , $f_{travel} < f_{swipe}$ + phi_1=90
    formula f_3*(2*x-1)+f_2*2*y;f_3*2*x+f_2*(2*y-1)
  - Type7: 2D sequence
    x-z-plane: f_{rot}=0 , $f_{travel} < f_{swipe}$ + phi_1=0
    formula f_3*(2*x-1)+f_2*2*y;f_3*2*x+f_2*(2*y-1)
  - Type8: 2D sequence
    x/y-z-plane: f_{rot}=0 , $f_{travel} < f_{swipe}$ + phi_2
    formula f_3*(2*x-1)+f_2*2*y;f_3*2*x+f_2*(2*y-1)
  - Type9: 2D sequence
    x-y-plane(along z-axis): f_{travel}=0 , $f_{rot} < f_{swipe}$ + phi_1
    formula f_3*(2*x-1)+f_1*2*y;f_3*2*x+f_1*(2*y-1)
  - Type10: MPS sequence
    No encoding gradients used: f_{travel}=0 , f_{rot}=0
    formula f_3*x

  ATTENTION: be careful adjusting these parameters. Not every scanner can handle every sequence (ask the support).

COMPASS related settings

The COMPASS settings define the parameters for the COMPASS sequence as indicated in the graph in Figure 1-7. The following parameters can be used:

DutyCycle	[0..1]	Defines the delay in percent between adjacent AC/DC bursts
Frequency steps		Defines the amount of frequency steps between a min and max frequency. The frequencies are defined by the frequency min and frequency max parameters. E.g. with min=5 kHz and max=10 kHz and 3 frequency steps, the frequencies are calculated to {5.000, 7.500, 10.000} [Hz]. *Hint 1-4**:** when using frequency step is 1, the maximum frequency is taken.*
Frequency min	[Hz]	Defines the minimum frequency.
Frequency max	[Hz]	Defines the maximum frequency.
Periods offset		Defines the waiting periods before data acquisition (ADC open).
Periods (signal)		Defines the periods for data acquisition. *Hint 1-5**:** to increase the SNR, increase the number of signal periods (averaging).*
Periods postset		Defines the number of periods after signal acquisition and before new frequency.
Harmonics		Defines the number of taken harmonics. *Hint 1-6**:** make sure, that the number of harmonics N does not affect the Nyquist theorem. Maximal available higher harmonic h_{max}:* $h_{max} = N \cdot f \leq SR/2 - 1$
DC-steps		Defines the number of steps between DC-min and DC-max parameters.
AC-steps		Defines the number of steps between AC-min and AC-max parameters.
DC [min...max]	[T]	Defines the minimum and maximum field strength of the DC range.
AC [min...max]	[T]	Defines the minimum and maximum field strength of the AC range.
ScanSym		Defines the scan symmetry: 0 means no symmetry, starting at DC-min and AC-min 1 means DC symmetry, starting at -DC-max...+DC-max 2 means AC symmetry, starting at -AC-max...+AC-max 3 means DC & AC symmetry

*Hint 1-7**:** The given parameters influence the length of the COMPASS sequence, especially the dutycycle, number of signal-periods and number of steps (DC, AC, frequencies). Be careful with using high numbers of these parameters since the COMPASS system can overheat during running the sequence.*

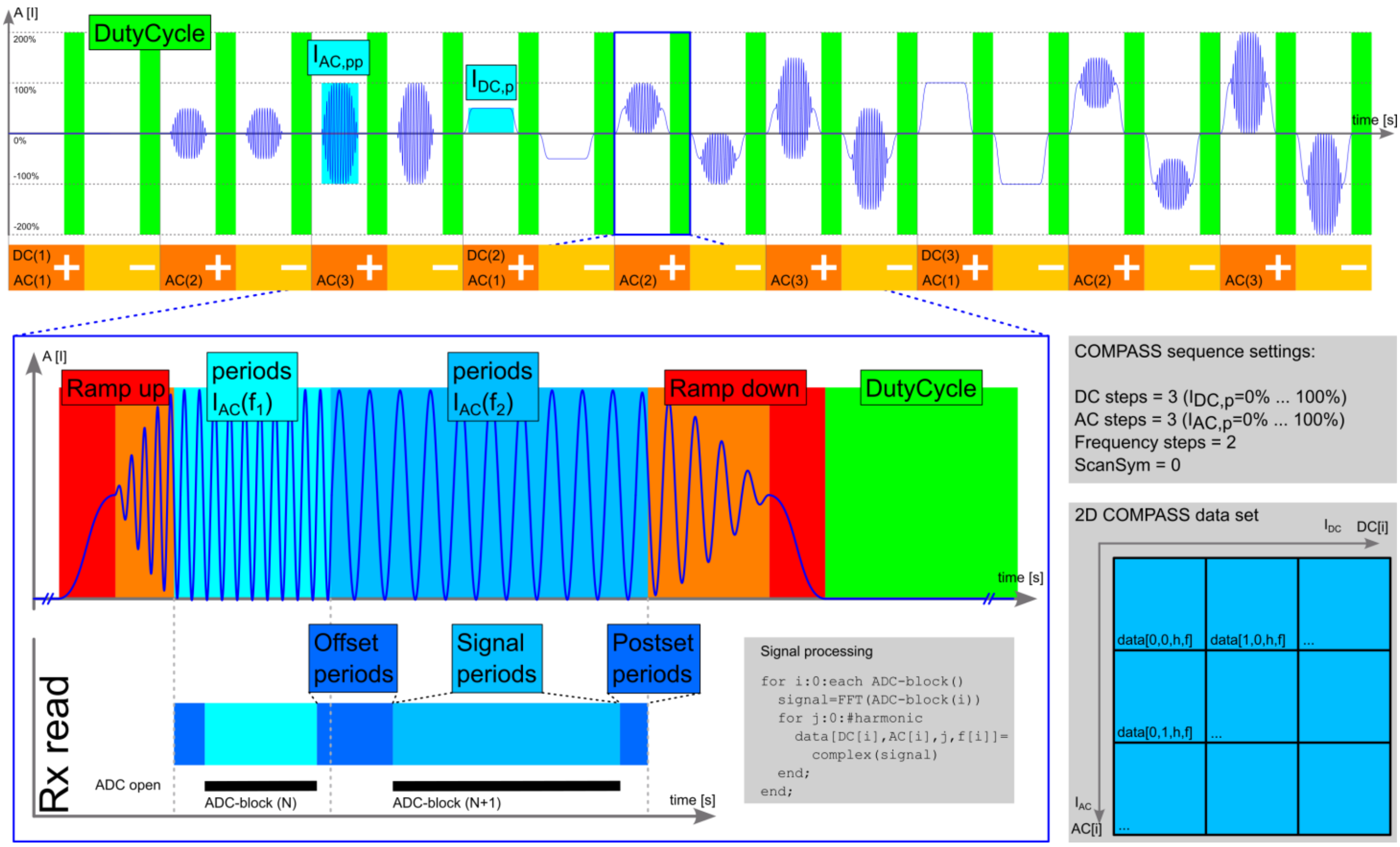


Figure 1-7: Sketch of a COMPASS sequence for the example sequence settings: 3 DC- and AC-steps, 2 frequency steps and no scan-symmetry. For a 3x3 sequence, the system uses 18 measurements (bursts) with positive and negative signs. For each defined frequency f and DC/AC ratio DC & AC, the drive calculates a complex number for all harmonics h `data[DC,AC,h,f]`. This number is used to generate the 2D COMPASS graphic (real, imaginary, abs, phase).

The sketch Figure 1-7 shows the COMPASS sequence for a small scan resulting in a 3x3 2D COMPASS graph. The sequence instead uses 18 measurements (bursts), where 9 of them are used for the positive data set and 9 for the negative data set. As indicated, the sequence uses positive and negative AC/DC values to generate the fields and acquire the signals. The background for this step is the avoidance of an uneven loading of the amplifier. Furthermore, the additional data set can be helpful to investigate the symmetry of the device and measured samples.

Hint 1-8: The negative data set is stored from server-site and is available in the `_backup` folder in the user-specific document directory.

Experiment tab

The experiment tab manages the data sets for each experiment.

- Infos: sets the name of the experiment
- Experiment infos: sets information and description for the experiment
- Data sets: all valid data streams in the experiment folder (see chapter 4) are indicated in the listbox. Selecting a specific data stream visualizes them in the graph at the bottom.

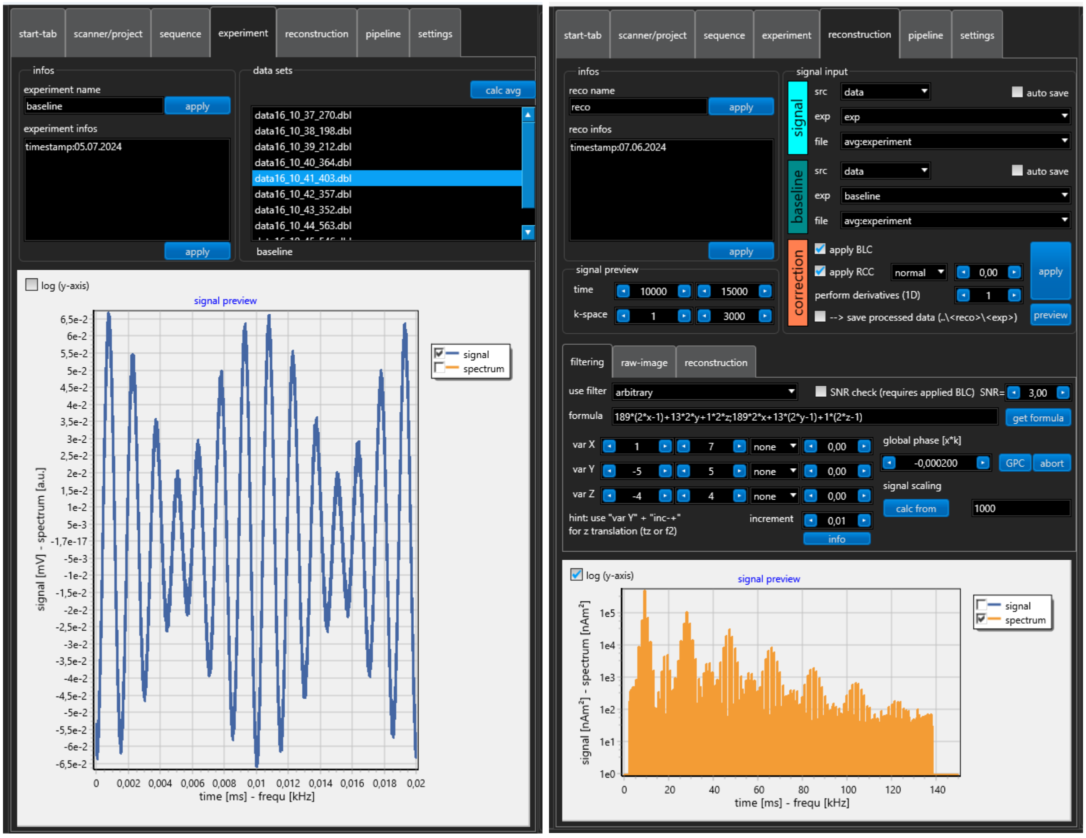


Figure 1-8: Left: overview of experiment tab. Right: overview of reconstruction tab.

Reconstruction tab

The reconstruction tab provides all parameter settings for reconstruction methods, which can be applied on the acquired data.

- Reco name: sets the name of the reconstruction
- Reco infos: sets information and description for the reconstruction
- Signal preview: sets the min/max values for indicating the preview signal in the graph at the bottom.
  Time sets the number of samples (1/SR [s]) showing the time data stream
  k-space sets the number of k-indices (df [Hz]) showing the Fourier spectrum of the data stream.
- Signal input: This area manages the input data used for the reconstruction process. This area is divided into three sub-areas:
  Signal (Signal input)
  src: sets the source for input data
    - data: uses stored data from the selected experiment folder
    - pipeline1/2: pipelines are defined on the pipeline tab

  exp: selects the experiment folder used for this reconstruction
  file: selects the input file
  Hint 1-9: there is an entry: "avg:experiment": this entry is a virtual data set averaging over all data sets within the experiment folder.
  auto save: when working with "pipelines", all incoming data streams will be stored automatically in the selected folder.

Baseline (Baseline input)

src: sets the source for input data

- data: uses stored data from the selected experiment folder
- pipeline1/2: pipelines are defined on the pipeline tab
- pipeline1/2-once: same as before, but after one input data, the source automatically jumps to “hold”. This feature is used for BLC purposes.
- hold: just use the last data set

exp: selects the experiment folder used for this reconstruction

file: selects the input file

Hint 1-10: there is an entry: “avg:experiment”: this entry is a virtual data set averaging over all data sets within the experiment folder.

auto save: when working with “pipelines”, all incoming data streams will be stored automatically in the selected folder.

Correction

Apply BLC: uses BLC (baseline correction) for reco process

Apply RCC: uses RCC (receive chain correction) for reco process

Perform derivatives: performs a derivation on the time signal *N*-times

Save processd data: saves also processed data sets within the reco folder

- Process area (filtering, raw-image, reco): This area provides the parameters for several processing steps:
    - Filtering: see chapter 4: filtering tab
    - Raw-image: see chapter 4: raw-image tab
    - Reconstruction: see chapter 4: reconstruction tab

Pipeline tab

The pipeline tab provides the GUI for managing incoming data streams from simulation software and/or real scanners. For that, different pipelines are used, which can be connected to different sources.

- Pipeline 1/2: a pipeline connects a third-party source with the software. Two different types of pipelines area available:
    - MMF: the MMF (memory mapped files) approach uses a virtual data transfer pipeline.
      Usage: select MMF and type in the name of the virtual MMF-file.
    - LeCroy: this methos uses the proprietary pipeline to communicate with digital oscilloscopes from Teledyne LeCroy.
- Time: sets the polling time for data receive
- TCP/IP client: This area uses a TCP/IP client to connect to the MPI server.

An overview of the provided commands can be found in

Channel Phase Calibration (CPC)

Devices: TWMPI

Special requirements: none

...in preparation

Auto Rx Calibration (ARC)

Devices: TWMPI, COMPASS, MPS

Special requirements: none

The drive console consists of a receive input with an adjustable pre-amplifier. This is used to amplify the incoming signal to a level where the sensitivity or dynamic range of the analog-digital converter (ADC) is used with the maximum available ENOBs (efficient number of bits) to ensure the best signal-to-noise ratio (SNR).

The automatic receive-chain calibration (ARC) is a feature, which uses a short sequence to check the saturation of the ADC and set up a pre-amplifier level, which guarantees an 80% saturation of the ADC.

Depending on the selected device, the software prepares a specific sequence depending on the given parameters:

TWMPI -take the sequence as it is (**ATTENTION**: averaging values are not affected)
-disable gradient fields (A_ch1 = A_ch2 = A_ch3 = A_ch4 = 0)

COMPASS -set AC-steps to 1; DC-steps to 2, frequency-steps to 1 (shortest possible sequence).
-set the maximum frequency (ACFrequencySteps=1)
-set scansym to 0

Hint 5-11: The user should put in the sample with the maximum expected signal before performing the ARC.

Magnetic Field Calibration (MFC)

Devices: TWMPI, COMPASS, MPS

Special requirements: 1×Pick-up loop

Disconnect TX1 and RX cabling from drive console.

Place a small pickup-coil (5 mm NMR glass tube) with the sample holder in the scanner and connect the cable to the RX of the drive console.

Set up the scanner via TCPIP interface with the following commands:

```
Reset_all_params
Set_params_amptx 1
```

The first command reset all parameters to the default settings, among others set the gradient field to zero.
The second command sets the main coil (drive field) to a given value representing an optimal value for 1 T/m (default sequence).
With the given parameters of the pick-up coil, the magnetic field B_{max} can be calculated:

$$B_{max} = \frac{U_{max}}{\pi \cdot N \cdot r^2 \cdot f}.$$

E.g., with the parameters N=10; r=1.9 mm; f=9.450 Hz; U_{max}=57 mV), the maximum magnetic field can be calculated to B_{max}=8.6 mT.

Based on the calculated magnetic field, the scanners' parameters for gradient strength can be adjusted and validated.

Hint 5-12: The U_{max} value is directly available in the uncorrected time signal from the calibrated drive system.

Data I/O interfaces.

Hint 1-11: at the moment, only servers on the same computer are supported.

Hint 1-12: use “reset”-button to clear the command list between client and server. Use this function to restart the communication process.

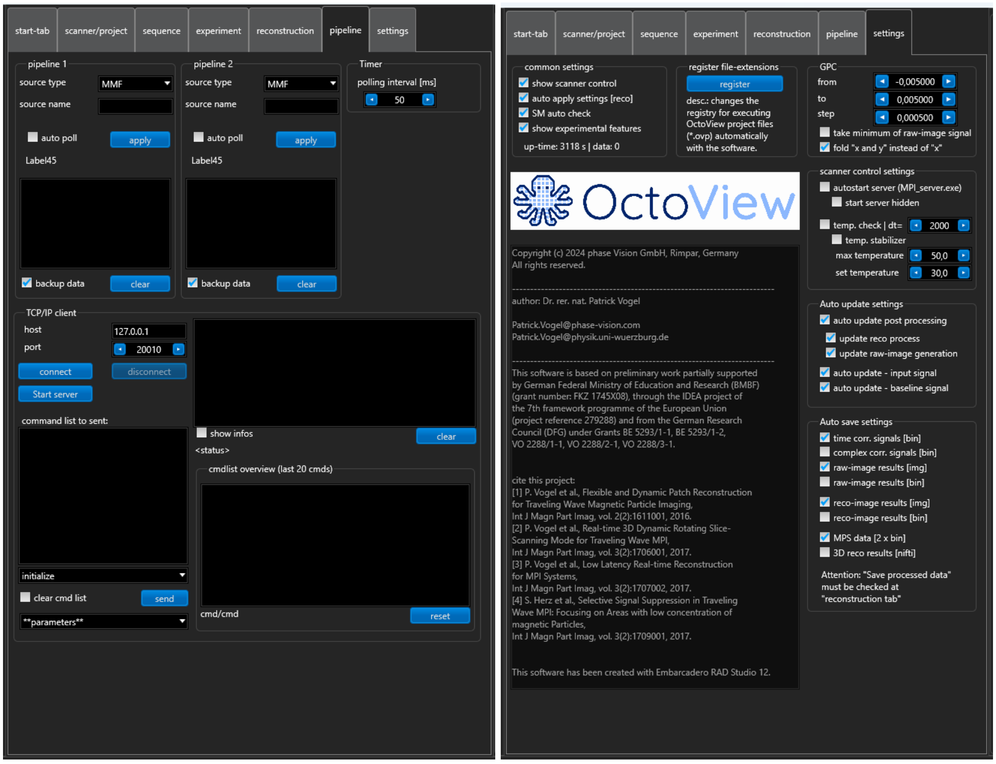


Figure 1-9: Left: overview of the pipeline tab. Right: overview of the settings tab.

Settings tab

The settings tab provides additional information about the software as well as additional global software settings.

- **Common settings:**
  - show scanner control: hide the scanner control bar
  - auto apply settings [reco]: automatic update of parameter changes for reco area
  - SM auto check: automatic check for correct system matrices
  - show experimental features: hide new features
- **Register file-extensions:**
  This registers the *.ovp file extension on the Windows OS. This allows a direct start of a project.
- **GPC:**
  - from/to/step parameters: defines searching parameter range and step-size
  - take minimum: takes the minimum instead of maximum
  - fold perform a 4-times folding
- **Scanner control settings:**
  - temperature check: activates a temperature check every 2 seconds
  - autostart server: autostart server if available. This starts a batch-script starting the server.exe.
    Hint 1-13: when using a Matlab-based server, this server has a higher priority.
    - start server hidden: hide the batch-script

- **Auto update settings:**
  - post processing: automatic parameter updates for processing area
    - reco process: for reco processes
    - raw-image process: for raw-image processes
  - input signal: for input signals
  - baseline signal: for input baseline signals
- **Auto save settings:**
  This section defines the processed data, which are stored within the reco folder.
  - corr signal [bin] saves the processed time data set as *.dbl file
  - raw-image results [img] saves the processed raw-image as 2D image file (*.png)
  - raw-image results [bin] saves the processed raw-image as 2D binary file (*.2df)
  - reco-image results [img] saves the processed reco-image as 2D image file (*.png)
  - reco-image results [bin] saves the processed reco-image as 2D binary file (*.2df)
  - MPS data [2 x bin] saves the processed H-M files (*.dbl)
  - 3D reco results [nifti] saves the 3D volume file (*.nifti)

Control bar (optional)

The **control bar** (Figure 1-10) provides the most required features for controlling an MPI scanner in one view:

← connecting to the MPI scanner via server

← adjust the **gradient strength** for a selected scanner and sequence

← performing **correction** methods as well as single measurements

← abort measurements if necessary

← the status of the scanner is visualized with

> **data status**: green means incoming data are fine, red means incoming data could be erroneous (check ADC saturation).
>
> **ADC**: gives the percentage of ADC saturation with the set "RX Uin MAX" value
>
> **temperature**: for each measurement, a temperature check is performed. The data can be exported.

← applying **averaging** for experiments: set the number of averages and the delta time.

Hint 1-14: be careful by setting up the delta time to small.

← adjust the **sensitivity of the receive chain**: the default value is 0.3 V. The range can be set between 0.010 V and 0.500 V. Check the status of the incoming signal.

← Set up a **multi-run** experiment: set the number of experiments with a given delta time. This experiment run is controlled by the software itself.

← Set up a **multi-run** experiment: set the number of experiments with a given delta time. This experiment run is managed by the server.

Figure 1-10: Overview of the control bar.

Helper windows

The GUI provides additional helper windows (Figure 1-11) to provide the user different visualizations such as

- "raw-image preview" window: Based on a frequency selection of f_1, f_2 & f_3, from each acquired signal a raw-image can be generated through drawing the signal values on a 2D image using the ratio of two frequencies (default: f_2 & f_3) line by line, where each horizontal line represents a full period (0..2π) of the second selected frequency. Along the z-axis, the raw-image represents a full period (0..2π) of the first selected frequency.

- “reco-image preview” window: Applying a reconstruction method on the acquired data results in a final reconstructed image showing the particle distribution, e.g. in 2D. Applying a 3D reconstruction method results in a sophisticated raw-image representing the 3D voxel structure as 2D projection (→ 3D visualization).
- “3D visualization” window: This visualization uses additional information of the system matrix to structure the raw-image results in a 3D space.

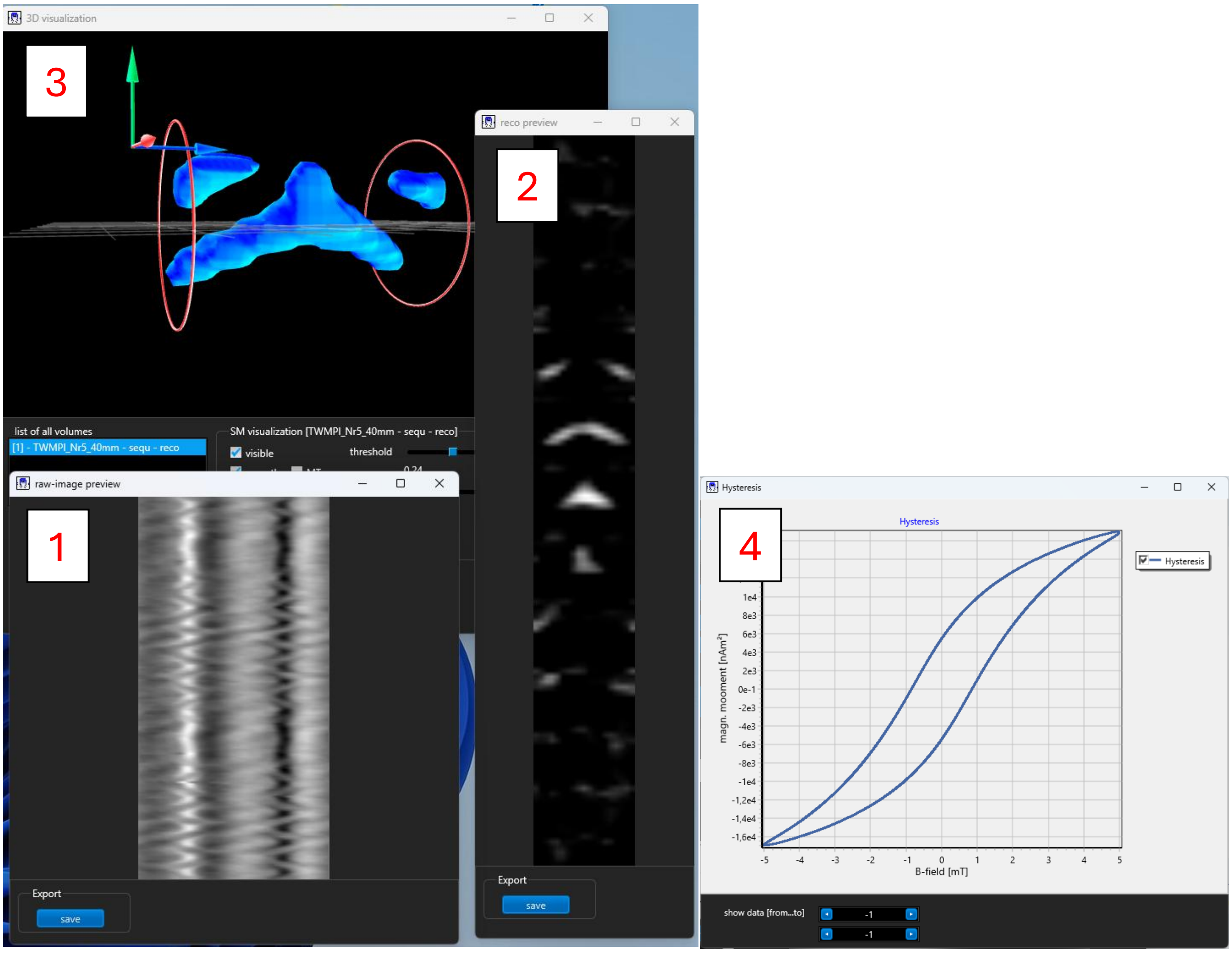


*Figure 1-11: Example screenshot of additional helper windows: **(1)** raw-image preview, **(2)** reco preview, **(3)** 3D visualization window, and **(4)** Hysteresis preview.*

2. Device principles

MPI – a short introduction

This chapter is used to provide some background information for this technology and helps to understand the philosophy of this software package.

MPI basics

Magnetic Particle Imaging (MPI) is a tomographic method for the visualization of magnetic materials such as magnetic nanoparticles (MNPs). MPI utilizes the non-linear magnetization response of the MNPs to time-varying magnetic fields with sufficient amplitude to generate higher harmonics of a given excitation frequency f_1 [1]. This non-linear magnetization behavior can be described by the Langevin-function indicated in the H/M plot in Figure 2-1.

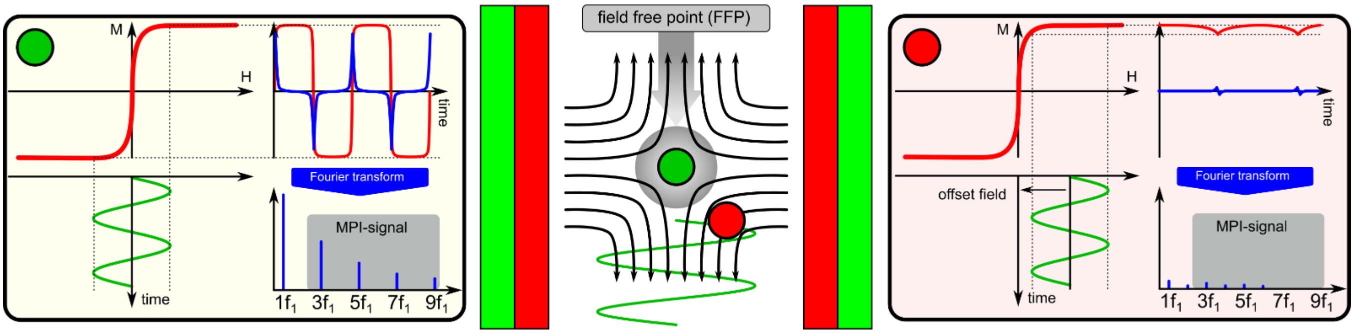


Figure 2-1: MPI basics: The Langevin function (red curve) describes the magnetization response of superpara-magnetic material (MNPs) to time-varying magnetic fields. As a result, higher harmonics ($n \cdot f_1$) can be occurred in an inductively measured signal (green side). Applying an offset field, the MNPs can be saturated, which suppresses the generation of higher harmonics (red side). For spatial encoding, a string gradient is used to reduce the area of possible signal generation, here in form of a field-free point (FFP). Through point-by-point scanning of the field-of-view (FOV), an image of the MNP distribution can be generated.

For spatial encoding, a strong gradient can be used, which can be generated e.g. by two permanent magnets facing each other with the same polarization. As a result, in the center of both magnets an area with reduced magnetic field in comparison to areas near the magnets can be occurred (magnetic field gradient). This area is called a field-free region (FFR) or in this case a field-free point (FFP). To separate for example two ensembles of MNPs (~10^{16} particles) indicated as red and green dot in Figure 2-1, where the green MNPs lies in the vicinity of the FFP with the reduced magnetic field and the red MNPs sees a higher magnetic field. Through the higher magnetic field, the red MNPs are saturated in such a way that their magnetization response to a time-varying magnetic field is almost zero. The green MNPs on the other side shows a high magnetization response.
By moving the FFPs along a given trajectory through the area of interest, an image (or 3D space) can be scanned point-wise to generate an MNP distribution.

Different encoding schemes

The above-mentioned encoding scheme is one possible way to determine the MNP distribution within the scanners' FOV. It turns out, that an encoding scheme based on a field free line (FFL) is more efficient and sensitive in comparison to the FFP design [2]. The reason is the coverage of a larger volume for signal acquisition [3]. There are several configurations that produce FFLs in 2D. To cover a full 3D FOV, additional mechanical parts have to be introduced resulting in a reduction of acquisition speed [4-6].

Traveling Wave Magnetic Particle Imaging using a Field-Free Line (TWMPI-FFL)

For a fully electrically driven 3D FFL MPI scanner, the Traveling Wave concept [7,8] is used in combination with a novel gradient coil setup generating the desired time-dependent field gradients [9,10].

The TWMPI FFL scanner works with so-called double helix coils and can rotate and move the FFL in 3D. To understand the idea of those coils the evolution from a single-wound solenoid to a cross coil is shown in Figure 2-2.

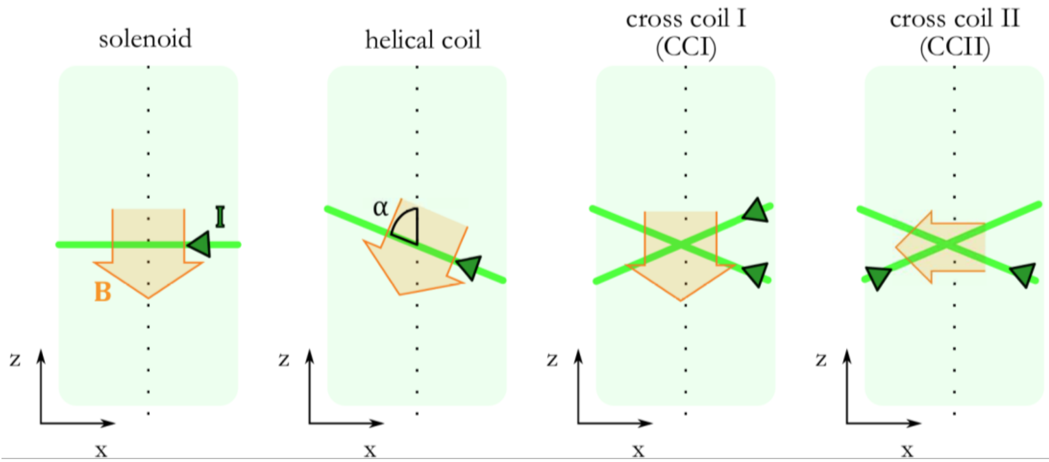


Figure 2-2: Evolution from a solenoid to a cross coil. First the solenoid is rotated (helical coil) and a second solenoid is added across the first one. There are two different outcomes of magnetic field direction depending on the direction of the currents flowing through the solenoids (CCI and CCII).

When placing two cross coils (CCII) next to each other with opposite current and therefore also magnetic field configuration, a static FFL in y-direction is generated in between as displayed in Figure 2-3 (a).

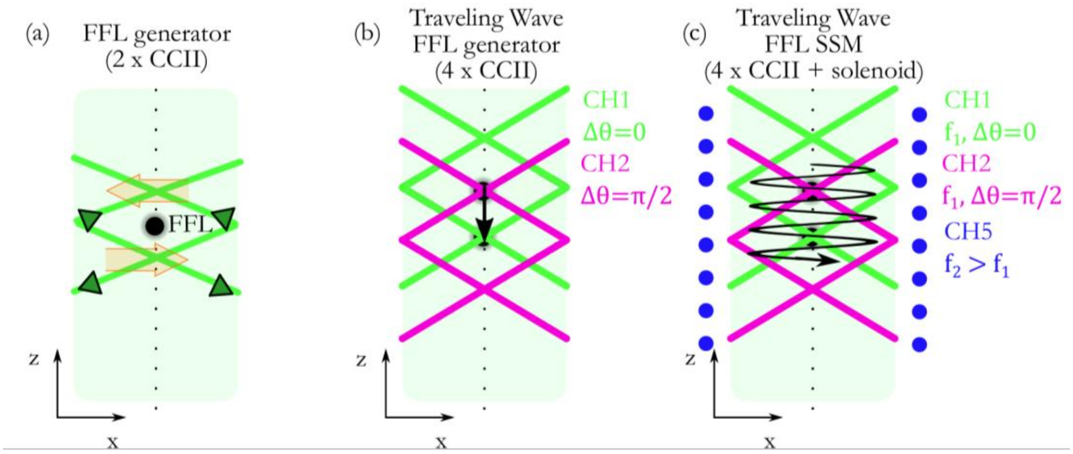


Figure 2-3: Evolution from a cross coil to the FFL TWMPI scanner with three channels. In (a) two cross coil pairs with opposite current direction produce an FFL in between them. By copying this coils configuration and paste it shifted in z-direction (b), there are two FFLs. If an alternating current is applied with a phase shift of 90°, the FFL is moving. In (c) a solenoid is added which has a larger frequency than the cross coils. Hence, the FFL is moving on a sinusoidal trajectory.

For moving FFLs for imaging, a second pair of CCII coils is assembled next to the first pair. Applying an alternating current to both pairs with a phase difference of π/2, the FFL is moved along the symmetry axis through the scanner (Figure 2-3 (b)) – **Traveling Wave concept**. Those two coil pairs are denominated as channel 1 and 2 (CH1 and CH2). To displace the FFL in x-direction, an additional solenoid coil encompasses the cross coils, which is denoted as channel 5 (CH5). This results in a 2D projection image (x-z plane) along the y-axis (Figure 2-3 (c)). The final step to fully enable 3D imaging requires two additional pairs of cross coils sitting at the same spot as the first two but rotated 90° around the z-axis. That alone would generate an FFL pointing in x-direction but also moving in z-direction. All coils together allow a full rotation and displacement in x-y plane and traveling along the z-axis of the FFL. Those last two pairs of cross coils are defined as channels 3 and 4 (CH3 and CH4).

In total the scanner concept has five channels. The first four channels belong to the 4 pairs of cross coils and the last one belongs to the outer solenoid. For generating a moving FFL in z-direction, an alternating current with frequency f_{travel}, amplitude A and a phase shift of π/2 between the cross coils has to be applied. This holds for all four cross coil pairs. Modulating the current with another frequency f_{rot} and this time a phase shift of π/2 between the channels 1 & 2 and channels 3 & 4 leads to a rotation of the FFL in the x-y plane. Ultimately, channel 5 is driven by an alternating current with amplitude B and a frequency f_{swipe}. Table 2-1 presents the control of all channels summed up.

Table 2-1: The configuration of five independent channels allows the scanner to generate and move the FFLs through the FOV on multiple trajectories. The common sequence uses a frequency combination of $f_{rot} < f_{travel} < f_{swipe}$.

channel	driving current
1	$A \cdot \sin(2\pi \cdot f_{travel} \cdot t) \cdot \sin(2\pi \cdot f_{rot} \cdot t)$
2	$A \cdot \sin\left(2\pi \cdot f_{travel} \cdot t + {}^{\pi}/_{2}\right) \cdot \sin(2\pi \cdot f_{rot} \cdot t)$
3	$A \cdot \sin(2\pi \cdot f_{travel} \cdot t) \cdot \sin\left(2\pi \cdot f_{rot} \cdot t + {}^{\pi}/_{2}\right)$
4	$A \cdot \sin\left(2\pi \cdot f_{travel} \cdot t + {}^{\pi}/_{2}\right) \cdot \sin\left(2\pi \cdot f_{rot} \cdot t + {}^{\pi}/_{2}\right)$
5	$B \cdot \sin\left(2\pi \cdot f_{swipe} \cdot t\right)$

It is important to know that channels 1 & 3 are not only two cross-coils (CCII) but three cross coils (CCCII) (see Figure 2-4). The third one has the same configuration as the first one and therefore CH1 and CH3 generate two FFLs to establish a larger FOV along the z-axis. CH2 and CH4 are correspondingly placed in a way that the FFL generated from CH2 and CH4 sits directly in the center of the other two FFLs generated by CH1 and CH3.

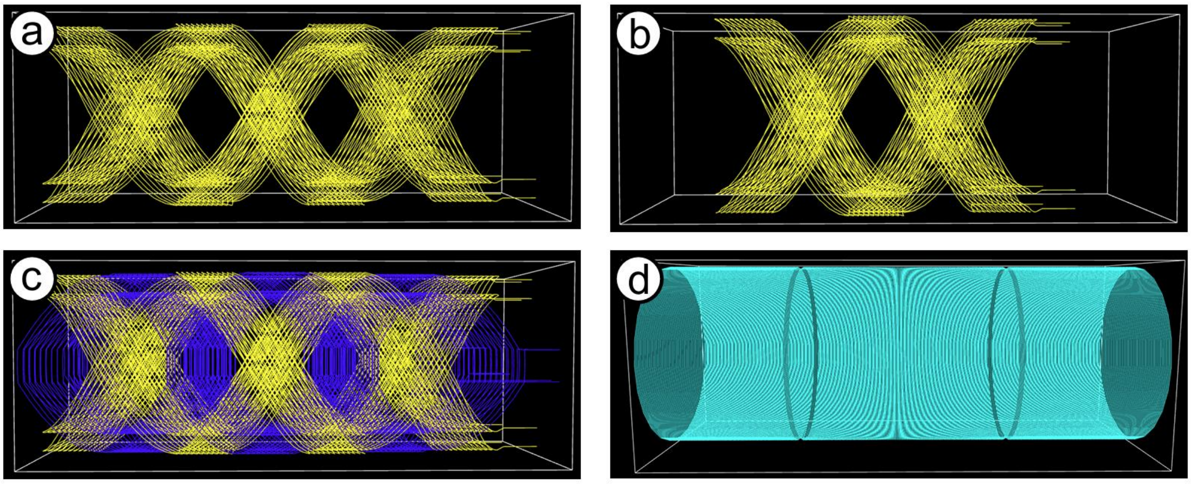


Figure 2-4: Coil configuration of FFL TWMPI scanner: (a) and (b) show CH1 and CH2, respectively. (c) display the combination of CH1 and CH3 and (d) receive coil.

The inner layer of the scanner is the receive coil (see Figure 2-4 (d)), which is a solenoid coil in a gradiometer configuration. The latter was chosen to decouple the transmitting and receiving system of the scanner. Therefore, the coil is divided into three parts. The middle part covers the FOV in the center of the scanner and the outer parts are wound in opposite direction. If a channel applies an alternating current and thus a magnetic field, the induced voltage in the middle part has the opposite sign compared to the outer parts. Assuming a constant magnetic field strength in the scanner and the total length of the outer parts is equal to the middle part, the induced voltage by the coils of the scanner vanishes.

As the scanner works with five different channels, several different scan modes are possible for 2D and 3D imaging as indicated Figure 2-5. Further details can be found in [11, 12] and in the additional documentation: ***TWMPI_sequences***.

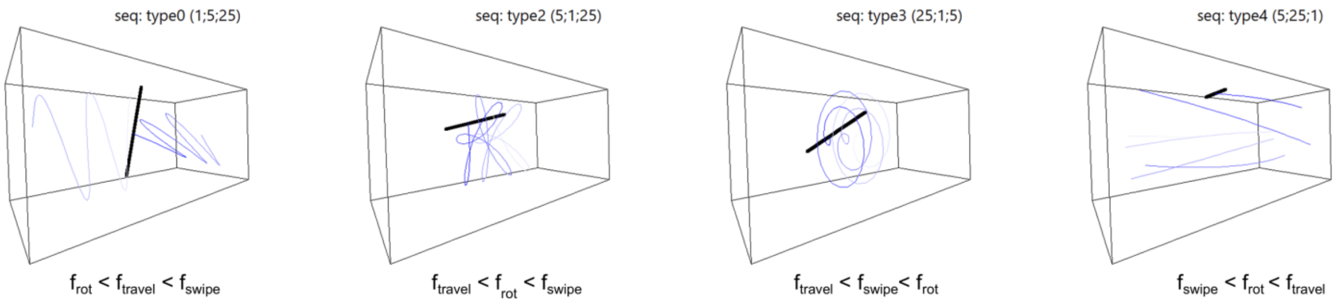


Figure 2-5: Examples of different scanning modes. The black line represents the FFL, the blue vanishing line the trajectory through the scanner. The default scanning mode is type0.

MPI scanner hardware and corrections

This section will give a short insight into the specifications of hardware parts that are essential for imaging. Regarding the combined cross coils channels (CH1- CH4), as said before CH1 and CH3 consist of triple cross coils as CH2 and CH4 consist of double cross coils. To produce a stronger magnetic field, there are another two triple and another two double cross coils that are part of the corresponding channel but have larger radii. 3D example structure of one triple cross-coil and one double cross coil is shown in Figure 2-6. All four channels together are just those two cross-coils copied with different radii and rotated. This scanner part (CH1-CH4) consists of four coil layers. Hence, there are four layers containing altogether eight cross coil configurations. As one notices, the cross coils are not single-wound to ensure a stronger gradient and therefore better defined FFL.

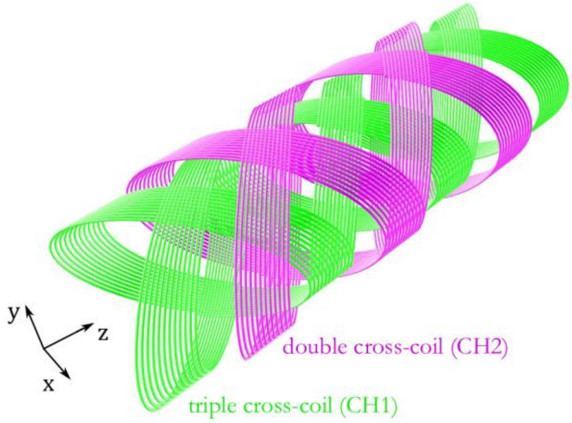


Figure 2-6: Rendering of one triple cross-coil of CH1 and one double cross-coil of CH2. The other triple and double cross-coils from CH1 and CH2 sit on top with a larger radius. The coils for CH3 and CH4 are similar but rotated 90° about the z-axis.

Transmit chain

The transmit chain consists of electrical components which are essential to control the five channels. It starts with a function generator that controls the channels with the desired current configuration depending on the desired scanning mode. There is need for an amplifier to create adequate current strengths to form strong gradients and therefore defined FFLs. Amplifiers are typically not able to amplify the input signal without adding higher harmonics of the input frequency. Since the higher harmonics arising from the particles are crucial for MPI, the ones created by the amplifier disturb the receive signal. Hence, a low pass filter is placed between the amplifiers and the channels to damp the higher harmonics before entering the coils.

Receive chain

The receive chain components are schematically shown in Figure 2-7. The magnetic particles and the excitation coils induce a voltage in the receive coil. The gradiometer configuration should in the first step damp the signal originating from the channels, i.e. the three frequencies f_{travel}, f_{swipe}, f_{rot}. Additionally, high-pass filters are involved to highlight the higher harmonics and damp the fundamental frequencies even more. Since the signal of the higher harmonics is quite small, a low-noise amplifier is typically introduced after filtering. At the end a digitization unit converts the voltage to a digital signal.

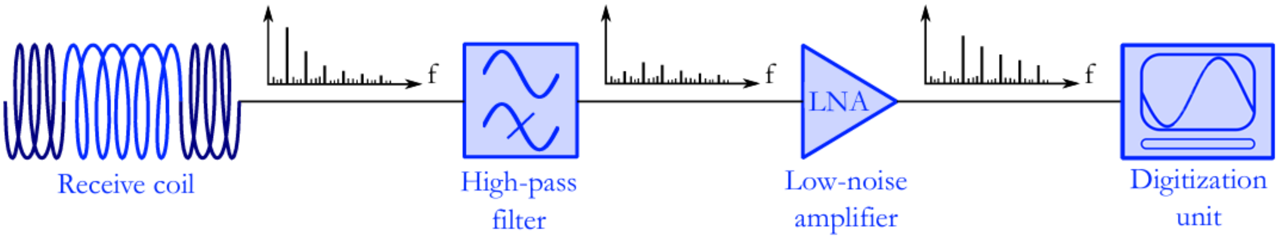


Figure 2-7: Sketch of the receive chain and the signal course. Despite the gradiometer configuration, the decoupling of the transmit and receive coils is not perfect. Hence, a high-pass filter additionally filters the excitation frequencies. After that, a low-noise amplifier enhances the higher harmonics of the signal before it gets to a digitization unit.

From MPI signal to the image

At the end of the receive chain, the digitization unit (ADC – analog-digital converter) converts the analog signal into a digital one. Of importance is the sampling rate (SR) of the digitization unit as well as the efficient number of bits (ENOB). The bandwidth defines the resolvable frequency range of the higher harmonics of the signal. Since the different scanning modes are working with at least two different frequencies, one needs to know how these two frequencies reflect in the particle signal. Figure 2-8 shows the impact of two excitation frequencies on the particle signal. On the left-hand side the Fourier transform of the excitation signal is shown for an exemplary frequency choice of f_1=1 kHz and f_2=10 kHz. In the results, there are two frequency components at 9 & 11 kHz ($f_2 \pm f_1$ kHz).

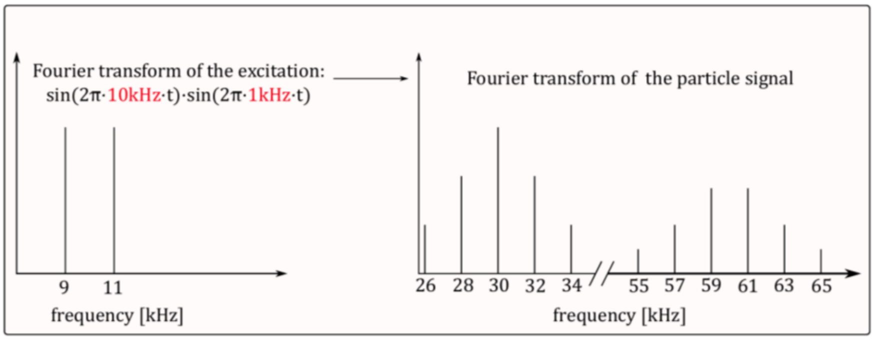


Figure 2-8: Frequency mixing in the xz- or xy-projection mode. Whenever using two excitation frequencies, the particle signal shows different behavior for odd and even higher harmonics of it. The right-hand side shows the mixing at two exemplary multiples of the excitation frequency.

The right-hand side shows two exemplary higher harmonics of the particle signals' Fourier spectrum. Regarding odd higher harmonics, in this example 30 kHz, the frequency components show at $(30 \pm 2n_1 \cdot f_1)$ kHz ($n_1 = \{0,\ldots,N_1\}, N_1 \in \mathrm{N}$), where N_1 defines the number of sidebands that are still resolvable. Considering higher harmonics, in this example an even higher harmonic of 60 kHz, the frequency components show at $(60 \pm (2n_1 - 1) \cdot f_1)$ kHz ($n_1 = \{1,\ldots,N_1\}, N_1 \in \mathrm{N}$). This phenomenon is known as frequency mixing [13]. In general, when dealing with two frequencies in the scanning mode for 2D, where the two frequencies are f_{travel} and f_{swipe} with $f_{\text{swipe}} >> f_{\text{travel}}$ the so-called frequency sidebands are located at

$$n_{swipe} = odd: f_{SB} = \left(2n_{swipe} - 1\right) \cdot f_{swipe} \pm 2n_{travel} \cdot f_{travel}, \\ n_{travel} = \{0, \ldots, N_{travel}\}, N_{travel} \in \mathbb{N}$$

$$n_{swipe} = even: f_{SB} = 2n_{swipe} \cdot f_{swipe} \pm (2n_{travel} - 1) \cdot f_{travel}, \\ n_{travel} = \{0, \ldots, N_{travel}\}, N_{travel} \in \mathbb{N}$$

where $n_{swipe} = \{0, \ldots, N_{swipe}\}, N_{swipe} \in \mathbb{N}$.

This gets even more complicated if three frequencies are involved, e.g., in the scanning modes for 3D. There, the generalized formula to calculate the frequency sidebands reads

$$n_{swipe} = odd: f_{SB} = (2n_{swipe} - 1) \cdot f_{swipe} \pm 2n_{travel} \cdot f_{travel} \pm 2n_{rot} \cdot f_{rot},$$
$$n_{travel}, n_{rot} = \{0, \dots, N\}, N \in \mathbb{N}$$

$$n_{swipe} = even: f_{SB} = 2n_{swipe} \cdot f_{swipe} \pm (2n_{travel} - 1) \cdot f_{travel} \pm (2n_{rot} - 1) \cdot f_{rot},$$
$$n_{travel}, n_{rot} = \{0, \dots, N\}, N \in \mathbb{N}$$

where $n_{swipe} = \{0, \dots, N_{swipe}\}, N_{swipe} \in \mathbb{N}$. To calculate the bandwidth, which is necessary for adequately mapping the received signal, the Nyquist–Shannon sampling theorem must be considered. Therefore, the required bandwidth BW for mapping the receive signal reads

$$BW_{2D} = 2 \cdot [(2 \cdot N_{swipe} - 1) \cdot f_{swipe} + 2 \cdot N_{travel} \cdot f_{travel}]$$
$$BW_{3D} = 2 \cdot [(2 \cdot N_{swipe} - 1) \cdot f_{swipe} + 2 \cdot N_{travel} \cdot f_{travel} + 2 \cdot N_{rot} \cdot f_{rot}],$$

where the 2 at the beginning of the right-hand side is due to the Nyquist–Shannon sampling theorem, N_{swipe} gives the number of odd higher harmonics and N_{travel},N_{rot} give the number of even higher harmonics of the corresponding frequency. The drive has a default sampling rate of 2.5 MS/s, which results in a maximum frequency of 1.25 MHz that is resolvable. With the sampling rate and the time of signal acquisition (T_{acq}), one is able to calculate the number of data points (*dl*, data length)

$$dl = SR \cdot T_{aqu}$$

With an acquisition time of 20 ms, the number of data points is *dl*=50,000 samples.

Data processing

Before the data is handed over to the reconstruction step, there are several processing steps to reduce the effects of the hardware components on the signal.

Baseline correction (BLC)

To eliminate disturbing peaks resulting from the excitation signal itself, the first step is to take a measurement without a sample inside the scanner. The measured signal ($S_{baseline}$) can then be subtracted from the measured signal with sample (S_{sample}) which gives an empty measurement corrected signal (S_{BLC}):

$$S_{BLC} = S_{sample} - S_{baseline}$$

Receive chain correction (RCC)

The hardware components of the receive chain (receive coil, high-pass filter and low-noise amplifier) distort the pure particle signal regarding its amplitude and phase across the whole frequency range. Since amplitude and phase of the particle signal are crucial for spatial encoding, it is necessary to specify how these particular hardware components influence them.

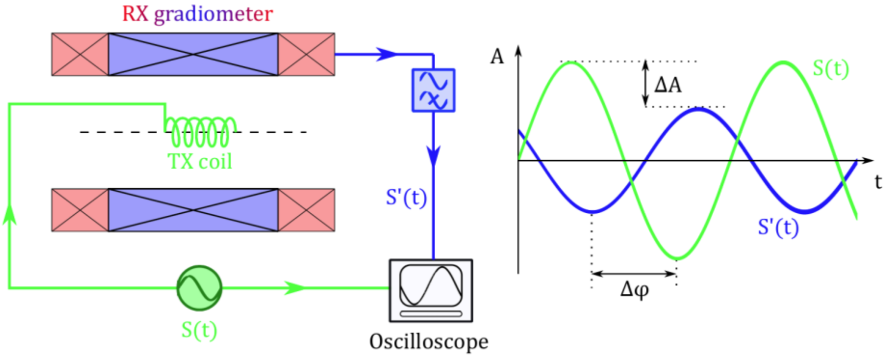


Figure 2-9: On the left: Sketch of the experimental setup for the receive chain correction. A small transmit coil (TX) is placed in the middle of the receive coil (RX gradiometer). A signal S(t) drives the transmit coil and is also measured by an oscilloscope. The signal received from the RX gradiometer S'(t) is also measured by an oscilloscope. On the right: An

exemplary measurement. From evaluating both signals, one is able to determine the amplitude and phase shift at a certain frequency due to the distortions from the receive chain.

Mathematically spoken: There is a transfer function *T*(*f*), which describes the change of the amplitude and phase of the signal in frequency domain. Hence, the measured signal *S*'(*t*) is a convolution of the transfer function in time domain with the time-derivative of the particle signal *S*(*t*):

$$S'^{(t)} = \dot{S}(t) * T(t).$$

It is the time-derivative, because the measurement is done inductively with the receive coil. A measurement setup to derive the transfer function is shown in Figure 2-9. A small transmit coil is placed in the center of the receive coil. A transmit signal is generated and the generated signal as well as the received signal is recorded. Doing this for one particular frequency, the result is shown on the right-hand side. Since one needs to know the amplitude and phase change of the whole frequency domain one could step-wise increase the frequency and do several measurements at different frequencies.
Another way to find *T*(*t*) is to do one measurement where the transmit signal is a short pulse which is like a Dirac delta function. Since it corresponds to a constant in the frequency domain, one has information about the whole frequency range with just one measurement. With the data from the transmit and receive signal, one is now able to construct the transfer function *T*(*f*). Hence, one is able to correct the data with the help of the Fourier transform by simply weighting the recorded signal with the transfer function:

$$\begin{aligned} &\mathcal{F}\big(S'(t)\big) = \mathcal{F}\left(\dot{S}(t)\right) \cdot \mathcal{F}\big(T(f)\big) \\ &\Rightarrow \hat{S}'(f) = 2\pi i \cdot \hat{S}(f) \cdot \hat{T}(f) \\ &\Rightarrow \hat{S}(f) = -\frac{1}{2\pi} i \cdot \hat{S}'(f) \cdot \hat{T}(f)^{-1}. \end{aligned}$$

Regarding the signal's phase, a time shift results in a rotation in the frequency domain after the Fourier transform, i.e.

$$x(t - t_0) \overset{\mathcal{F}}{\rightarrow} e^{-i\omega t_0} \cdot \hat{x}(\omega).$$

Therefore, the measured phase from the transfer function is multiplied with the signal to correct the signal's phase.

Software filtering

After the amplitude and phase correction, one is able to perform a software filtering of the data. When dealing with data in the Fourier space, filtering is done by erasing the unwanted frequency components. Therefore, low-pass, high-pass and band-pass filters are easy to implement on the software side. Also, since one in principle knows every peak location due to the frequency mixing, one is also able to erase frequencies that are not part of the particle's signal. This can be accomplished by firstly letting a peak picking algorithm decide what peaks are detectable and then choose which to erase.

2D Gridding

The gridding of the data onto a 2D plane is essential for image-based reconstruction approaches. The data points are gridded corresponding to the FFL trajectory and create a raw image. Therefore, the parameters needed are data length *dl*, sampling rate *SR* and excitation frequencies f_1, f_2, f_3.

Pixel density increase

The pixel density of the raw images is an important factor for reconstruction of the images. One way to increase the pixel density was already shown in Figure 2-10 (b,c). The gridding process creates an image that has four parts originating from the two FFLs and the forward- and backward-scan of the FFL. These four parts depict the same FOV so it is possible to fold them together like one would do it with a piece of paper but with additionally inverting the intensity so that the signal gets larger in sum. By this, the pixel density increases by a factor of four and the interpolation is reliable. Another already discussed option is to choose the operation frequencies in a way that with every FOV scan, the trajectory is slightly shifted. Hence, more different pixels are scanned, and the pixel density increases. Both mentioned approaches are shown in Figure 2-11 [14].

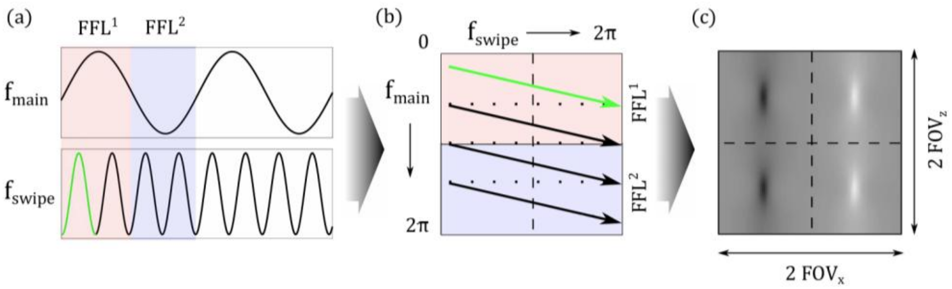


Figure 2-10: Visualization of the gridding principle of a 2D sequence. (a) The two frequencies f_{travel} and f_{swipe} determine which trajectory the FFL takes. The FFL travels on a sinusoidal path where the ratio between the frequencies decides how many half periods the FFL takes per FOV passing. (b) The FFL path is linearized displayed within 2π: At the top, the first FFL and on the bottom the second one. In total, two FFLs pass the FOV one by one within one T_{travel}. The result is a 2D raw image with four signals, in this example from a point-like sample. (c) The horizontal line divides the first from the second FFL and the vertical line divides the scanning direction of the FFL (back and forth).

2D visualization (x-z projection – type7-9)

In the xz-2D-sequence the FFL passes the FOV on a sinusoidal trajectory along the z-direction. The frequency f_{travel} determines the time (T_{travel} = $(f_{travel})^{-1}$) in which the two FFLs pass the FOV one by one. The ratio between the second frequency f_{swipe} and f_{travel} gives the number of half periods of the sinusoidal trajectory within one FOV passing. The trajectory from one T_{main} is then projected onto a 2D image where one period of f_{swipe} resembles the horizontal range and one period of f_{main} the vertical one. For simplicity, the sinusoidal path of f_{swipe} is linearized. The process is shown in Figure 2-10 (a,b). The raw image is divided into four sections. The horizontal split is a result of the two opposite sign FFLs (FFL1 and FFL2) during one T_{travel} and the vertical split comes from the different scanning directions of the FFL due to the sinusoidal path. An exemplary raw image of a point-like sample in the center of the scanner is shown on Figure 2-10 (c). As one can see, there are no blank spots in the raw image, which would have been consistent with the gridding process. It is not possible to fill every pixel of the raw image with data, since the trajectory of the FFL is not scanning every pixel of the FOV. The filling depends on the number of samples and the chosen frequencies. To get a complete image, the missed pixels are interpolated. It is obvious, that the better the pixel density of the trajectory, the better the interpolation and the reconstruction works. Ways to increase the pixel density are shown in Figure 2-11. Another important task is to match the starting points of the gridding algorithm and the data recording, otherwise the image is shifted. One way to accomplish this is to simulate the measurement process and look where the FFL is generated at the start of the sequence. If it is not in the top corner of the FOV, one could simply adjust the sequence with adequate phase shifts. Another possibility is to generate the raw image of e.g. a measured, centered point-like sample and correct the phase manually by shifting the data points in a way that the point-like sample on the raw image is located realistically. An important decision to make is which frequencies to pick to get an optimal raw image. One can estimate after how many periods of f_{travel} the trajectory of the FFL exactly repeats [15]

$$N_r = \begin{cases} 1, & f_{swipe} \bmod f_{travel} = 0 \\ \dfrac{LCM(f_{travel}, f_{swipe})}{f_{swipe}}, & f_{swipe} \bmod f_{travel} \neq 0 \end{cases}$$

where LCM(*a*,*b*) means the least common multiple of *a* and *b* and the *mod* means the modulo operator that returns the remainder of a division. Imagine f_{swipe}=10 kHz and f_{travel}=1 kHz, there are exactly five periods of f_{swipe} per FFL. Since the second FFL within one T_{main} starts always with the negative x-value in comparison to the first one, the trajectory exactly repeats after two FOV passings, which is one period of f_{main}. After the trajectory is repeating the signal is averaged, since every spatial coordinate was scanned before. If the trajectory is not instantly repeating, the FFL travels on shifted trajectories trough the FOV. This leads to a higher pixel density because the FFL scans several new spots that are slightly shifted in comparison to the last FOV passing.

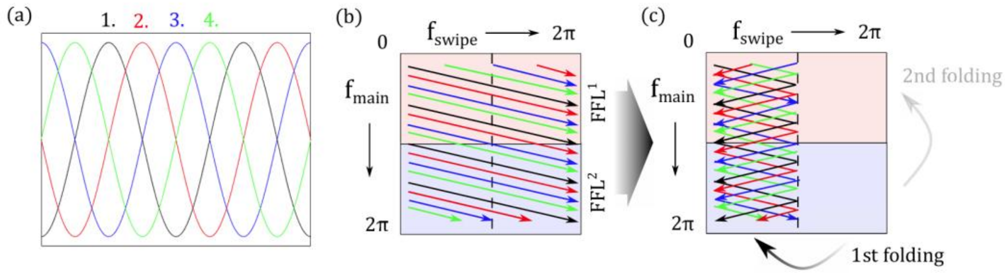


Figure 2-11: Approaches for a pixel density increase. (a) The frequencies are chosen such that the trajectory is shifted with every T_{travel}. The result is shown in (b): The pixel density is increased. (c) The first folding brings the forward and backward scan together. The second folding would additionally unite both FFLs' data and leads in total to a four-times higher pixel density than before folding.

Image reconstruction

In this section, reconstruction methods for the scanning modes are explained. The creation of raw images is crucial for reconstruction of images-based reconstruction approaches since they are directly using the raw images for further processing. The *k*-space based reconstruction methods on the other side do not require the raw images but also needs some corrections steps that are illustrated in detail in its subsection. At the end, one general approach is discussed whereby it is possible to reconstruct the image of all scanning modes.

2D reconstruction (x-z projection – type7-9)

The raw image of the xz-2D-sequence shows not the original image but its convolution with the point spread function (PSF) of the scanner. As the raw image is 2D, so is the PSF. A detailed mathematical description of the multidimensional theory of MPI is found in [16]. As the name suggests, the PSF is the response of a system to a point-like sample. Thus, the intuitive way to find the PSF for the scanner is to measure or simulate a point-like sample. Since there is no point-like sample in reality, one has to approximate it by a small sample of superparamagnetic particles. The simulated PSF for the FFL TWMPI scanner in the 2D-xz mode is visualized in Figure 2-12.

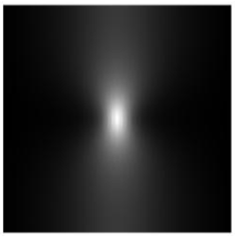


Figure 2-12: Simulated PSF of the FFL TWMPI scanner for a 2D xz-sequence (PSF type 1) [28]. The simulation frequencies were f_{travel}=950 Hz and f_{swipe}=9.8 kHz.

If the PSF is known, one is able to reconstruct the image by a deconvolution algorithm [14,15].

One often-used algorithm is described by the so-called Wiener deconvolution [17]. It aims at minimizing the noise during deconvolution and works in the Fourier space. Mathematically spoken, the raw image $i'(x,y)$ is the convolution of the original image $i(x,y)$ with the 2D PSF of the system in addition to an unknown noise $n(x,y)$

$$i'(x,y) = (i * PSF)(x,y) + n(x,y).$$

The algorithm can find an approximation $\hat{\imath}(x,y)$ to $i(x,y)$, if $i'(x,y)$ and the 2D PSF is known. The Fourier transform $F(I(u,v))$ of $\hat{\imath}(x,y)$ can then be calculated by

$$\mathcal{F}\big(I(u,v)\big) = \underbrace{\left[\frac{\left|\widehat{PSF}(u,v)\right|^2}{\left|\widehat{PSF}(u,v)\right|^2 + NSR(u,v)}\right]}_{\mathcal{F}(PSF)} \cdot \frac{I'(u,v)}{\widehat{PSF}(u,v)},$$

where *NSR*(*u*,*v*) is the noise-to-signal ratio in the frequency domain and *I'*(*u*,*v*) is the Fourier transform of the raw image *i'*(*x*,*y*). Applying an inverse Fourier transform, the approximation of the original image *i*(*x*,*y*) can be calculated. An exemplary convolution of the original image with the PSF is found in Figure 2-13. One can clearly see the impact of the PSF onto the raw image.

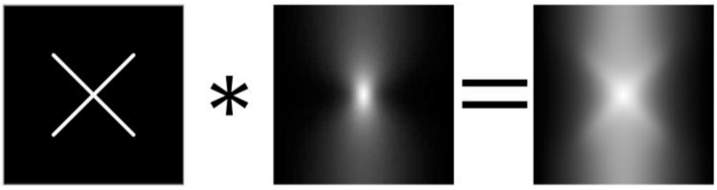


Figure 2-13: Visualization of the convolution of the original image (left) with the PSF (middle) resulting in the raw image (right).

k-space base reconstruction (system matrix reconstruction)

The system matrix reconstruction approach is more independent of the scanning mode. It does not matter how the FFL exactly moves through the FOV, it should be same for each measurement. Sometimes it is very hard to determine every detail the scanners hardware does, this approach can be important to compensate magnetic field inhomogeneities, different distortions and other impacts on the signal.
In general, one can describe the measurement data as a vector $\boldsymbol{m}$ which results from the multiplication of the system matrix S with the vector of the particle concentration $\boldsymbol{c}$ [18, p. 127f], i.e.

$$S \cdot \boldsymbol{c} = \boldsymbol{m}.$$

This means that the two steps to find $\boldsymbol{c}$ are determining the system matrix and inverting it. The system matrix contains information on how particles are mapped by the scanner depending on their position in the scanner. Hence, to construct the system matrix of a scanner, for each position the corresponding PSF inside the desired FOV is measured. Of course, the finer the sampling, the larger but also more precise the system matrix will be.
System matrices built out of a real sample: Producing the system matrix by using a real point-like sample of the particle system, which is used in the upcoming experiments, e.g. using a robot arm is the most correct way to generate an exact reconstruction kernel. However, it is the most time-consuming approach.
Model-based system matrix: To overcome this issue, the generation of the system matrix can be performed by simulation using a digital-twin of the MPI scanner. This is faster but not as exact as real measurements since effects, which cannot be calculated, are not taken into account. The magnetization model is a difficult point: the default way is to use the Langevin function, which is the fastest way for calculation, but this model does not consider any relaxation effects. However, the way for system matrix generation is similar: a point-like sample is positioned at every point in space, and a unique data set is generated (Figure 2-14). When using this model-based system matrix (using Langevin function) for reconstructing a simulated sample, there is one significant difficulty, but reconstructing a measured signal, this can be more difficult. Due to the electrical components in the transmit chain, the trigger of the FFL trajectory is unknown and in general not the same as in the simulation of the system matrix. This mainly affects the phase of the received signal and leads during image reconstruction to a spatial shift of the object (**hardware-phase**) (in best case: when using complex sequences, this effect can produce more interesting effects, e.g., when scanning with a 3D type0 sequence within a TWMPI scanner, a point-like sample can appear as a donut-like structure). Furthermore, the real particle system itself influences the phase of the signal (**sample-phase/MNP-relaxation**).

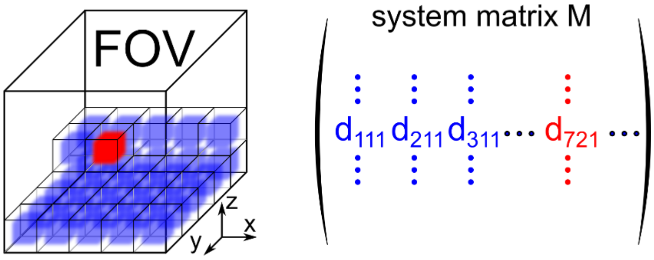


*Figure 2-14**:** Sketch of building a system matrix from a point-like sample positioned at each point in space in the FOV. The information written in the appropriate column can be taken from k-space (peak-picking) or raw image information.*

To solve the hardware-phase/trigger issue, the hardware's delay time to the simulation has to be determined. This can be performed in three different ways: **1.** step-wise adaption of the trigger-point within the simulation, generating the system matrix and reconstructing a known sample. **2.** Changing the trigger point of the hardware: all TWMPI scanner have the possibility to change the phases of the transmit channels, which is equivalent to change the trigger points. **3.** For TWMPI scanners, there is a third way to correct the phase/trigger of the reconstruction: the GPC is a global phase, which is applied during the signal processing to virtually adapt the phase of the measured signal to the system matrix (**see global phase correction**).
At the end all approaches will come the same result. One so-to-say adapts the data to the system matrix. It has to be mentioned that this approach is not usable without performing an RCC (receive-chain correction).
A second possibility would be an image-based system matrix [19]. There, one builds the raw image for every point-like sample and corrects the phase shift manually by e.g. shifting the distorted point-like sample in the center of the scanner back to its real position. This necessary offset is then applied to all of the other raw images. The system matrix will then be assembled with the corrected raw images.
When the system matrix is constructed, there are plenty of possibilities to invert it. The singular value decomposition (SVD) method for example factorizes the matrix into a product of a unitary matrix, a diagonal matrix and another unitary matrix, which makes it easy to generate the pseudoinverse. For more details on the SVD approach, see [20]. Another option is using algorithms that iteratively try to construct the inverse, like Kaczmarz's algorithm, also called algebraic reconstruction technique (ART) [21].

COMPASS – a short introduction

Short introduction
Critical Offset Magnetic PArticle SpectroScopy (COMPASS) [31] is a novel measurement method expanding the Magnetic Particle Spectroscopy (MPS) [26] method by an additional offset magnetic field (DC) in addition to a time varying magnetic field (AC). The combination of both AC and DC fields allows the investigation of multidimensional pattern of the signal behavior for the higher harmonic response of magnetic material. Furthermore, at specific ratios of AC and DC fields nodes (zero-crossings) appear in the pattern of real and imaginary parts offering critical points (CPs) with steep phase jumps. The position of these CPs highly depends on the effective viscosity combining several parameters of the particle and their surrounding media.

Fingerprint routine
The magnetic nanoparticle answer is measured at different combinations of AC- and DC-magnetic fields within field ranges of about 0 mT to 20 mT each.
The measured data is rearranged in 4D datasets with the following dimensions:

- Dimension 1: DC-field strength
- Dimension 2: AC-field strength
- Dimension 3: higher harmonic
- Dimension 4: variable parameter (e.g. hydrodynamic diameter, viscosity, frequency…)

To do so, each measured dataset is Fourier transformed and the complex value for a selected range of higher harmonics is extracted and rearranged in the 4D dataset (Figure 2-15). The fourth dimension generally results from different samples, e.g. particles with different hydrodynamic diameters. Thus, it is useful to join the 3D dataset of each sample in a separate step to create the 4th dimension.

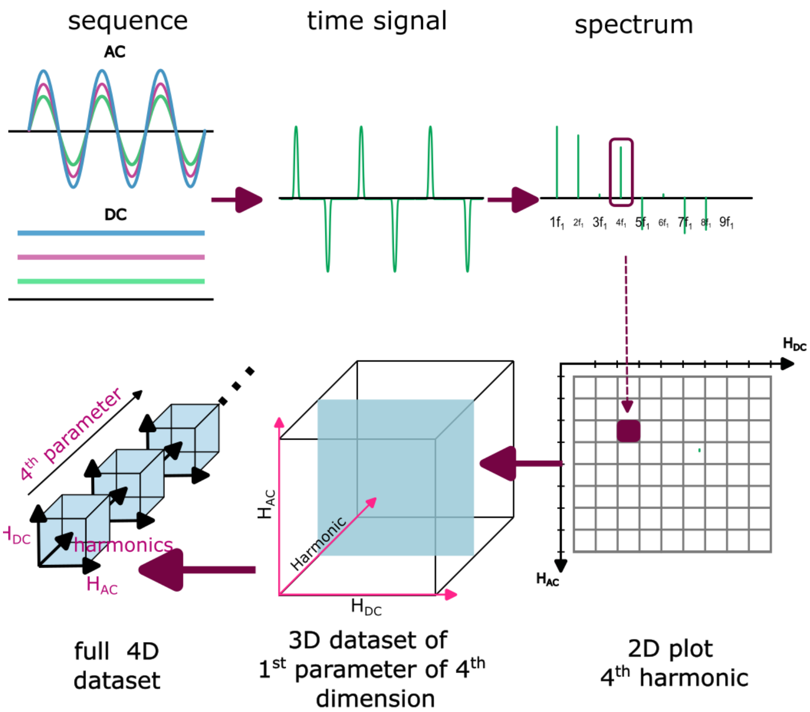


Figure 2-15: Sketch of the parameter extraction and value assembling into a 3D space as preparation for full 4D COMPASS dataset.

With this arrangement it is possible to extract information about characteristic critical points (CPs) for all harmonics and field configurations (Figure 2-16 right).

Critical points are steep changes in the complex phase occurring for certain combinations of AC and DC field strength values (Figure 2-16 left). The higher the observed harmonic the more critical points occur. These combinations are characteristic for each particle system and its properties.

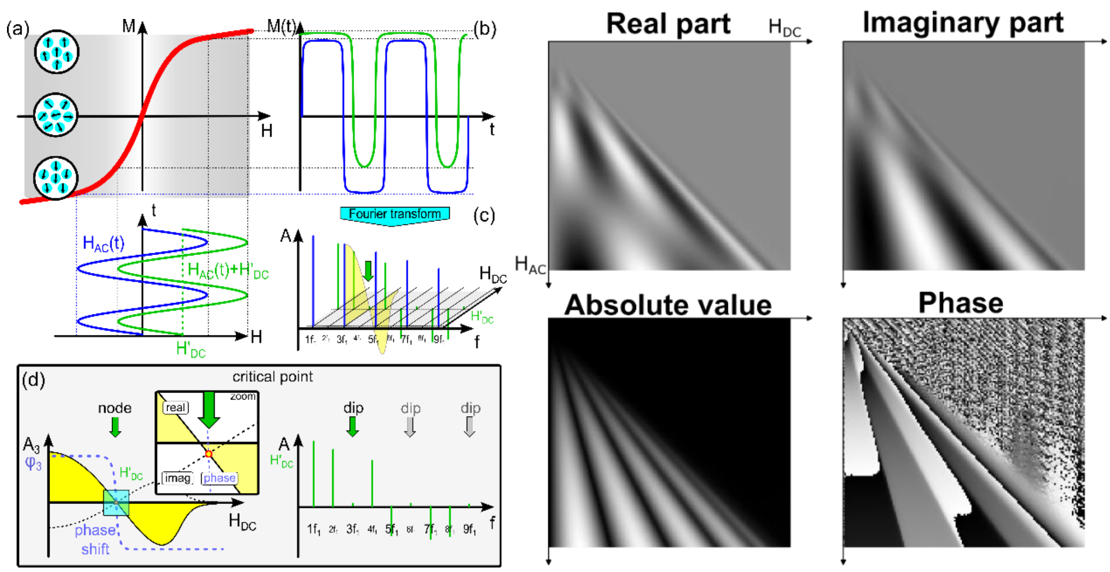


*Figure 2-16: **Left**: **a, b** The behavior of the magnetization M of MNPs in dependency of external magnetic fields H can be described by the non-linear Langevin function (red curve). Exposing an MNP ensemble to a sinusoidal magnetic field $H_{AC}(t)$ with frequency f_1 and sufficient amplitude, the magnetization response M(t) consists not only of the fundamental frequency but also odd (and even) higher harmonics ($f_n = n \cdot f_1$) depending on the presence of an offset magnetic field H_{DC}, which can be visualized in the Fourier spectrum. **c** Visualizing the dependency of the harmonic A_n for the n-th higher harmonic for increasing offset magnetic field H_{DC} (with $H_{DC} < H_{AC}$). The specific shape for varying H_{DC} with nodes (green arrow) depends on the harmonic number n. As an example, the real part of the third harmonic of simulated data is indicated to show the connection between a "dip" in the Fourier spectrum and a "node" in the $A_n(H_{DC})$ plot: this point is called the critical point (CP). **d** In the vicinity of a CP, the position, which is most susceptible to the sample parameters, the phase $\phi(H_{DC})$ of the signal shows an approx. 180° degree shift with a strong slope. Thus, even minimal changes in the*

sample parameters and thus in the shape of the $A_n(H_{DC})$ and $\phi(H_{DC})$ curves result in high changes in the phase $\phi(H_{DC})$ and thus in the resulting signal. The sixth and ninth harmonic are integer multiples of three and therefore happen to also vanish in this case (gray arrows). ***Right:*** *Example of a typical pattern of the 4th higher harmonic for a simulated COMPASS data set.*

To find the optimal CP position, the derivative of the phase is calculated. Optionally, the derivative can be performed using the Savitzky-Golay filter to smooth the curve. Due to the shape of the phase change, the derivative exhibits peak with the maximum at the steepest position of the occurring phase change. To extract the maxima with a higher accuracy than the sampling of the AC and DC field strength range, a gaussian fit is performed with selected sampling points in the vicinity of a maximum (Figure 2-17). The sampling points are selected by a variable threshold. This ensures that just enough points are included to create a gaussian fit. This justifies the use of a gaussian fit, since it does not underly any physical model in this case.

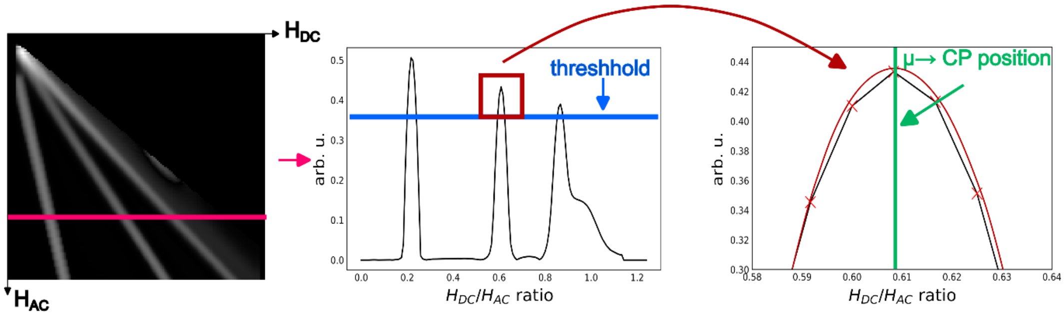


Figure 2-17: Accurate extraction of CP position using multiple steps: derivation, smoothing, thresholding and curve fitting.

The maximum of the determined peak position corresponds to the critical point position. Additionally, other peak parameters like the maximum height or FWHM can be extracted with this method.

Alternatively, the dataset can be interpolated and the maxima of the derivative by a simple peak picking algorithm.

Since the measured data is summarized in a 4D Array it is possible to extract the positions of one specific critical point in the 4th dimension. This allows the visualization of the course of the CP position for different sample parameters, e.g., here the hydrodynamic diameter for different samples (Figure 2-18).

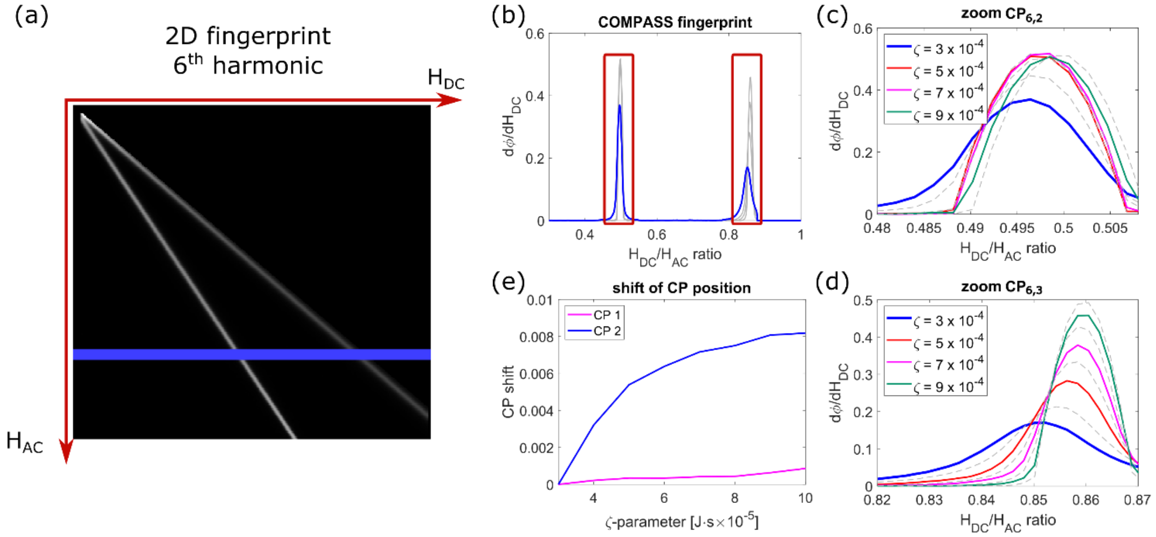


Figure 2-18: COMPASS fingerprint for MNP-samples with different hydrodynamic diameters.

References

[1] B. Gleich & J. Weizenecker, Tomographic imaging using the nonlinear response of magnetic particles, Nature 435:7046, 1214–17, 2005.

[2] J. Weizenecker et al., Magnetic particle imaging using a field free line, J Phys D: Appl Phys 41(10): 105009, 2008.
[3] D.L. Parker & G.T. Gullberg, Signal-to-noise efficiency in magnetic resonance imaging, Med Phys 17(2):250-7, 1990.
[4] M. Weber et al., Novel field geometry using two halbach cylinders for FFL-mpi, Int J Magn Part Imag 4(2), 2018.
[5] K. Bente et al., Electronic field free line rotation and relaxation deconvolution in magnetic particle imaging, IEEE TMI 34(2):644–51, 2014.
[6] P.W. Goodwill et al., Projection x-space magnetic particle imaging, IEEE TMI 31(5):1076-85, 2012.
[7] P. Vogel et al., Traveling Wave Magnetic Particle Imaging, IEEE TMI 33(2), 400-7, 2014.
[8] P. Vogel & P. Klauer, et al., Dynamic Linear Gradient Array for Traveling Wave Magnetic Particle Imaging, IEEE Trans Magn 54(2): 5300109, 2018.
[9] C. Greiner et al., Traveling Wave MPI utilizing a field-free line, Int J Magn Part Imag 8(1):2203027, 2022.
[10] P. Vogel, M.A. Rückert, V.C. Behr, System and method for generating a traveling field free line, EP3545835A1, US20210137407A1, WO2019185292A1, 2018, 2019.
[11] C. Greiner, Traveling Wave Magnetic Particle Imaging using a Field-Free Line, master thesis, Julius-Maximilians University Würzburg, 2021.
[12] P. Vogel et al., iMPI: portable human-sized magnetic particle imaging scanner for real-time endovascular interventions, Scientific Reports 13:10472, 2023.
[13] P.W. Goodwill et al., Narrowband magnetic particle imaging, IEEE TMI 28(8):1231–1237, 2009.
[14] P. Vogel et al., First in-vivo Traveling Wave Magnetic Imaging of a beating mouse heart, Phys. Med. Biol. 61(18): 6620-6634, 2016.
[15] P. Vogel et al., Superspeed traveling wave magnetic particle imaging, IEEE Trans Magn 51(2):1-3, 2015.
[16] P.W. Goodwill & S.M. Conolly, Multidimensional x-space magnetic particle imaging, IEEE TMI 30(9):15811590, 2011.
[17] A.P. Dhawan et al., Image restoration by wiener deconvolution in limited-view computed tomography, Applied optics 24(23):4013–4020, 1985.
[18] T. Knopp & T.M. Buzug, Magnetic particle imaging: an introduction to imaging principles and scanner instrumentation, Springer Science & Business Media, 2012.
[19] P. Vogel et al., Flexible and dynamic patch reconstruction for traveling wave magnetic particle imaging, Int J Magn Part Imag 2(2):1–11, 2016.
[20] G.W. Stewart, On the early history of the singular value decomposition, SIAM review 35(4):551–566, 1993.
[21] R. Gordon et al., Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography, Journal of theoretical Biology 29(3):471–481, 1970.
[22] L. Li et al., Transverse MNP Signal-Based Isotropic Imaging for Magnetic Particle Imaging, IEEE TIM 73:4501413, 2023.
[23] P. Vogel et al., Low Latency Real-time Reconstruction for MPI Systems, Int J Magn Part Imag 3(2):1707002, 2017.
[24] W.E. Lorensen & H.E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, ACM Computer Graphics, 21(4): 163–169, 1987.
[25] R.W. Cox et al., A (sort of) new image data format standard: NiFTI-1, OHBM (Budapest) #10, 2004.
[26] S. Biederer et al., Magnetization response spectroscopy of superparamagnetic nanoparticles for magnetic particle imaging, J. Phys D: Appl Phys 42:205007, 2009.
[27] J. Tafur et al., Development and validation of a 10kHz-1MHz magnetic susceptometer with constant excitation field, J. Appl Phys 111:07E349, 2012.
[28] J. Rahmer et al., Signal encoding in magnetic particle imaging: properties of the system function, BMC Med Imag 9:4, 2009.
[29] P. Vogel et al., Highly Flexible and Modular Simulation Framework for Magnetic Particle Imaging arXiv:2208.13835, 2022.
[30] MFS5 website: mfs5.avicula.net
[31] P. Vogel et al., Critical Offset Magnetic PArticle SpectroScopy for rapid and highly sensitive medical point-of-care diagnostics, Nature Comm, 13:7230, 2022.

3. Example projects

In this section, a short example is provided to demonstrate the workflow with OctoView and MFS5.

Connect your MPI project to the simulation framework MFS5

Requirements

MFS framework with a prepared virtual scanner project

The OctoView software also can be used for working directly with external simulation software. This example shows a step-by-step tutorial for the creation and connection with the simulation framework: Magnetic Field Simulator (MFS) [29,30].

Step 1: Start the software and perform the initial settings as indicated in Figure 3-1. The left sidebar is now available for further setups.

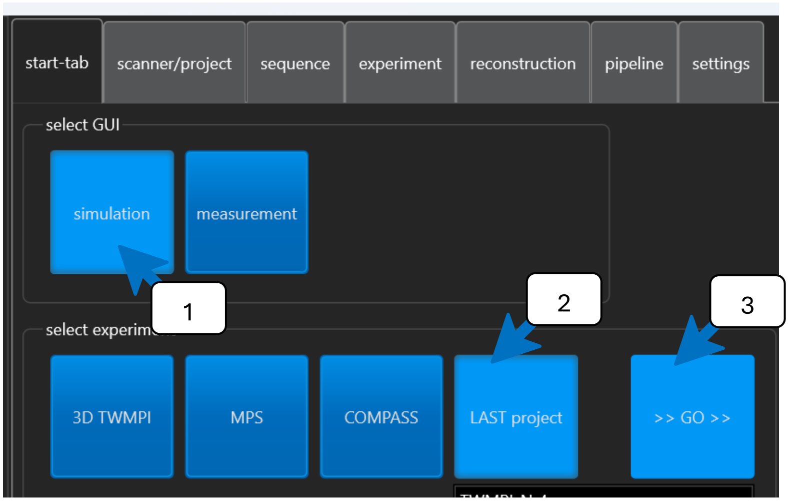


Figure 3-1: Start-tab configuration.

Step 2: In the left sidebar, "add" a new scanner-project (Figure 3-2). The software asks you for the place the project has to be stored: confirm this with "select folder".

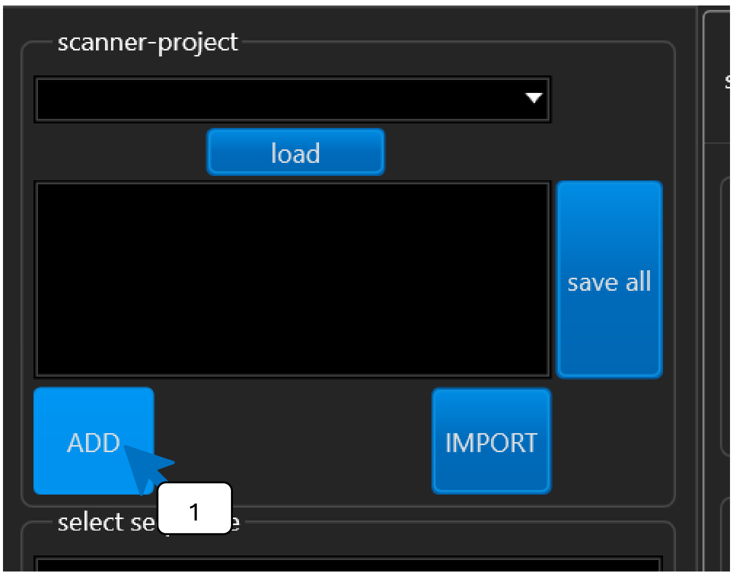


Figure 3-2: Add a new scanner-project.

Step 3: A new scanner-project (default-name: TWMPI_XX) is now available (Figure 3-3). Depending on the scanner the project is connected, adjust the name and FOV-size. Furthermore, additional notes can be made in the "scanner infos" area.
Select a pre-defined systemmatrix via "add" button in the system matrices area.

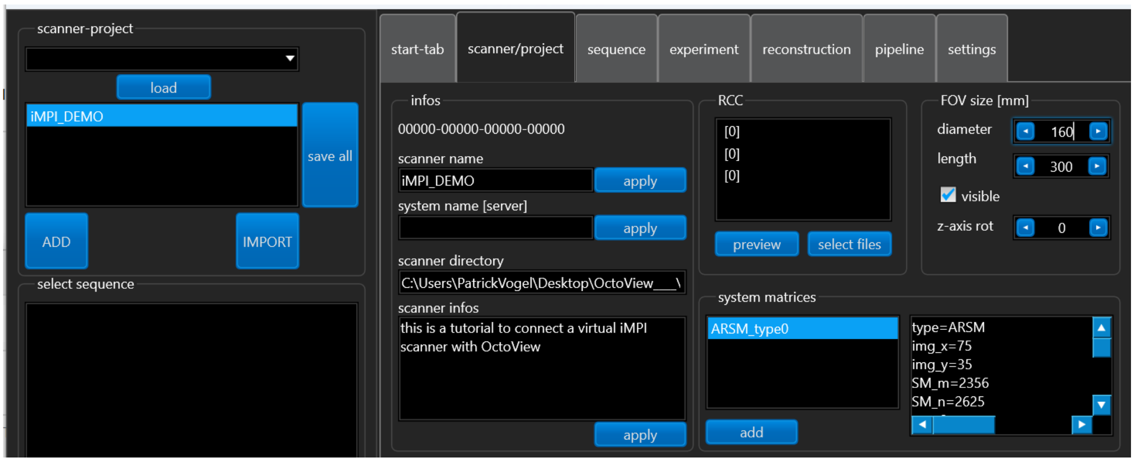


Figure 3-3: Adjust project-name and FOV size of the new scanner-project, here "iMPI_DEMO". Use the scanner infos field for adding specific notes to the project.

Step 4: Add a new sequence, rename it to "type6_2D", add additional notes and setup the desired frequencies of the scanner (Figure 3-4).
Since the iMPI scanner provides a 2D projection scanning mode, only two frequencies have to be set here: f_2 (travel) and f_3 (swipe). The frequencies are f_2=60 Hz and f_3=2,420 Hz. Furthermore, the data length is 125,000 samples, which corresponds to an acquisition time of 50 ms. The sequence is a standard 2D sequence, here type6 (2D), which uses 2 independent frequencies.
Hint 3-1: ***Confirm all changes with the corresponding "apply" button.***

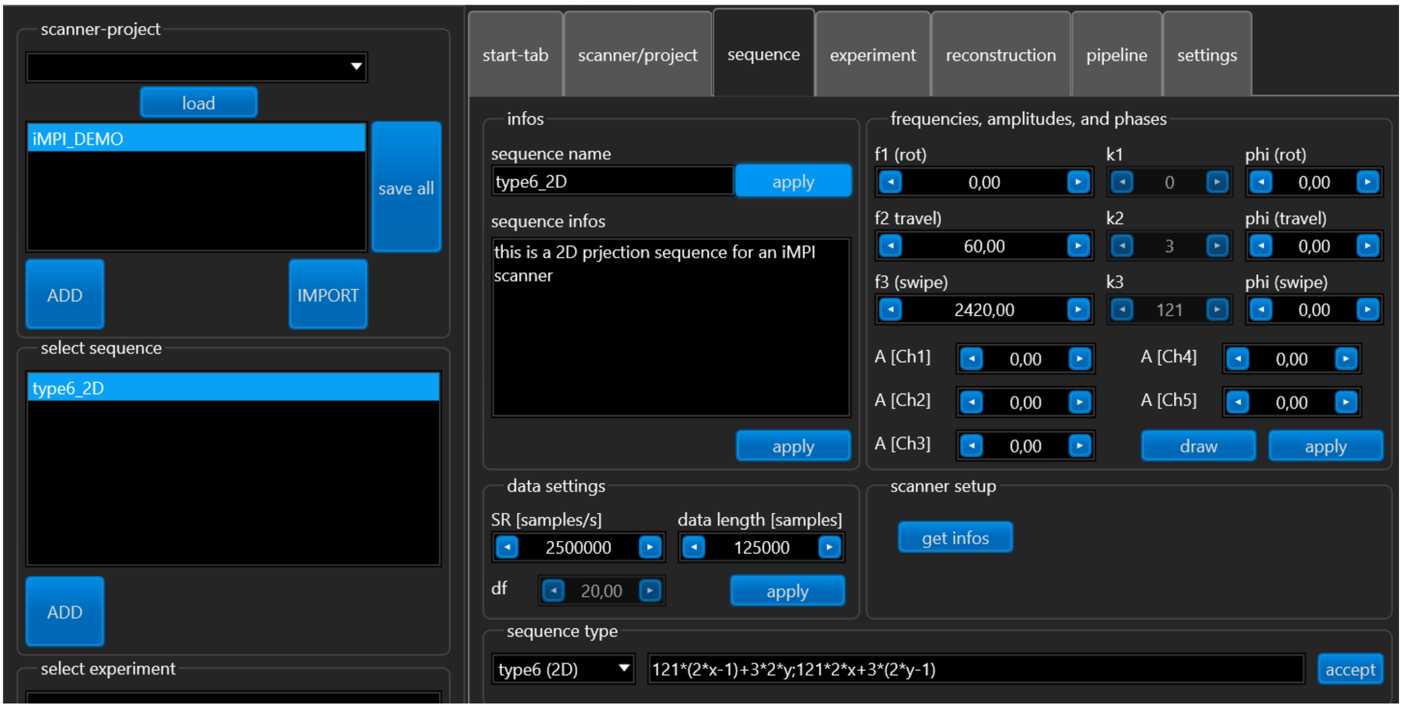


Figure 3-4: Setup the sequence of the virtual scanner, here a2D projection sequence for an iMPI scanner with a data length of 50 ms (125.000 samples) and frequencies of f_2=60 Hz and f_3=2,420 Hz.

Step 5: Add a new experiment "test" and a new reco "SM_type6". Set up the signal input to "test" and "pipeline1".

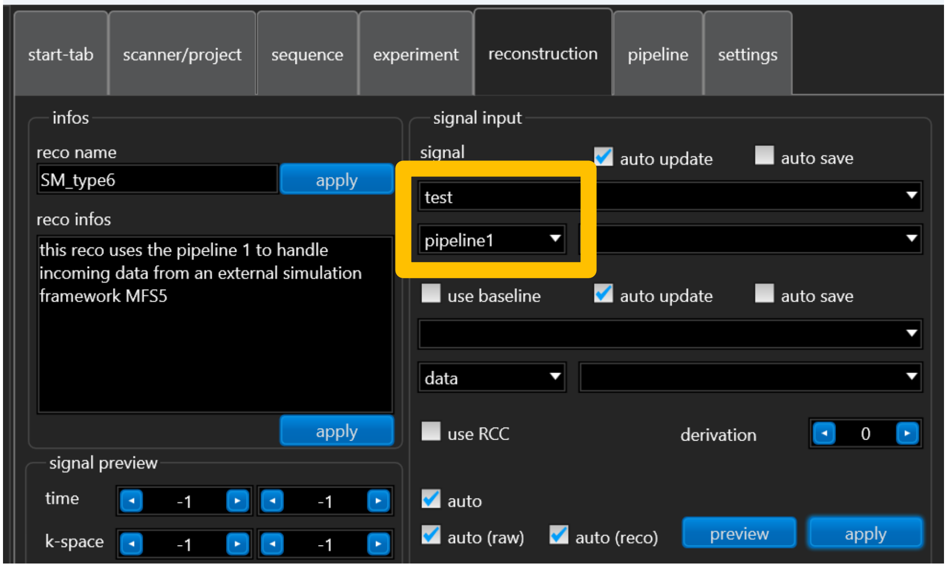


Figure 3-5: Set up the signal input area.

Step 6: Set up the interface connection via the pipeline tab (Figure 3-6). For the MMF name use here "MMFSim". Activate "auto poll" and confirm with "apply" button.

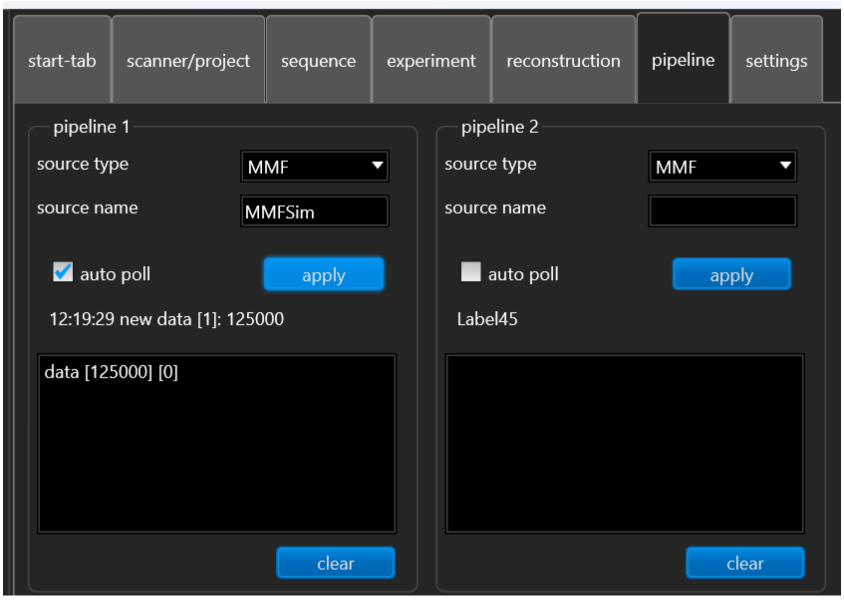


Figure 3-6: Set up the MMF interface.

Step 7: Start MFS and open the desired project (iMPI.mfs5). Go to the "Cond" tab and set up the MMF connection. Select a receive coil (e.g. rx_static_big) and press "generate data set". Figure 3-7 shows a screenshot of the MFS software.

In the OctoView software now a confirmation of transferring a data set should appear in the pipeline 1 area.

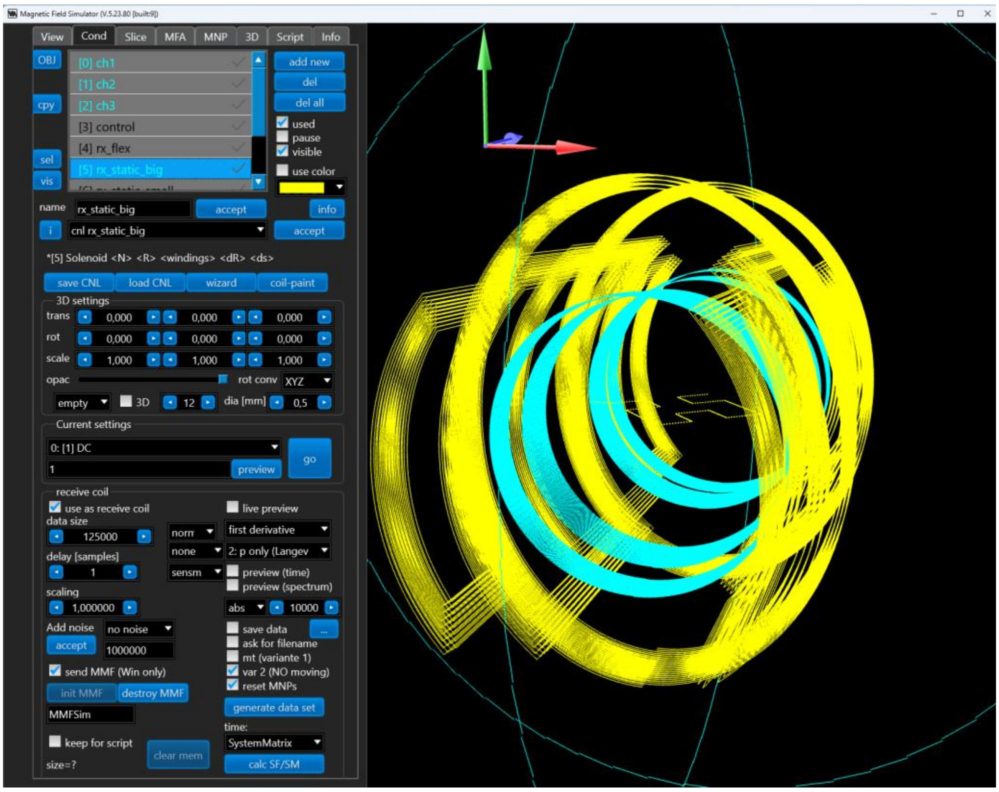


Figure 3-7: MFS simulation environment for emulating MPI scanner systems: here an iMPI scanner is pre-loaded.

Step 8: Set up "raw-image" parameters in the OctoView software (Figure 3-8 left). In the result, the raw-image preview helper window pops up and shows the raw-image of a complex sample convolved with the PSF type 1 (Figure 3-8 right).

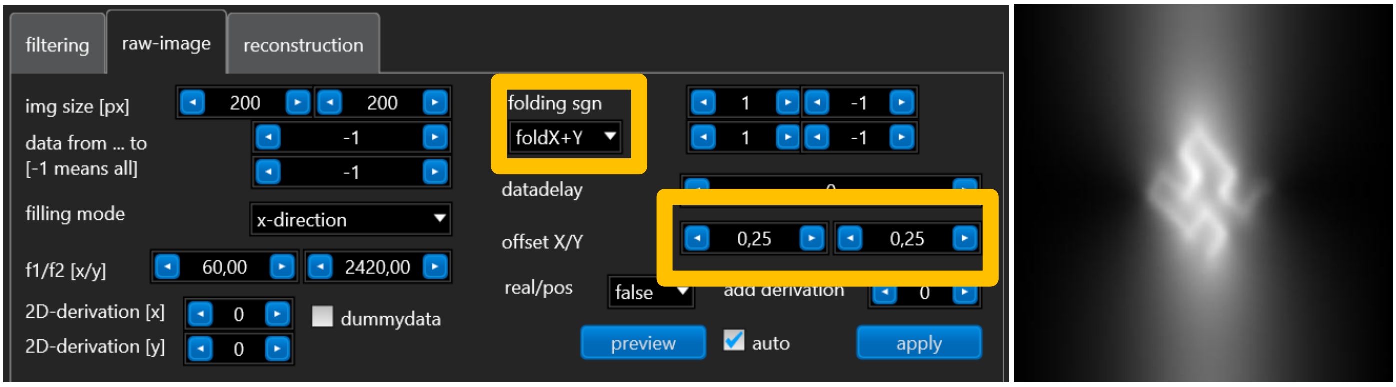


Figure 3-8: Left: parameter settings for generating the raw-image. Right: Screenshot of the raw-image preview result showing a complex sample convolved with PSF type 1.

Step 9: Set up the system matrix parameters in the "reconstruction" tab (Figure 3-9 left). Chose the reconstruction type (select type) "SM" for system matrix. Confirm with "apply" button. Select a pre-loaded system matrix, here "ARSM_type0", set iterations to 20, lambda to 0.05, threshold to 2,000 and check "force real". In the result two helper windows pop up and show the reco preview and also the result in 3D (Figure 3-9 right).

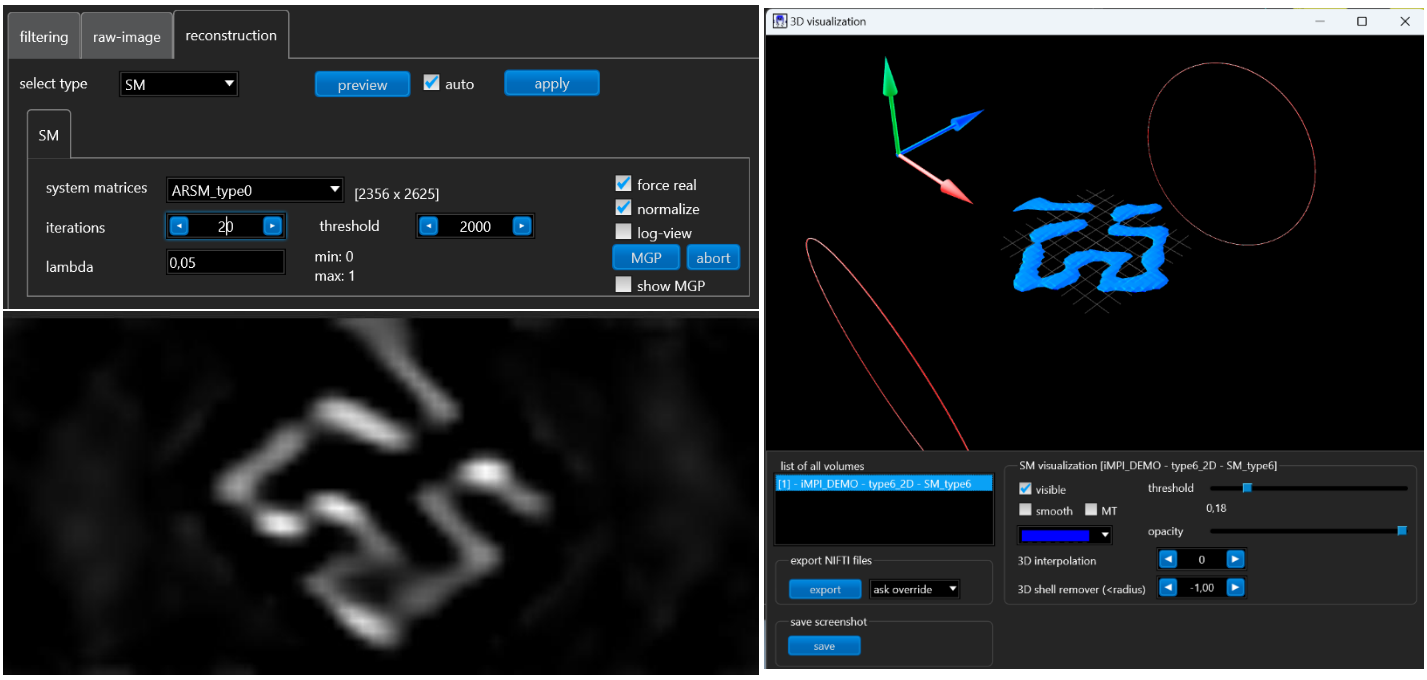


Figure 3-9: Left: "reconstruction" tab shows the parameters and settings for using the system matrix "ARSM_type0". The reco preview helper window pops up and shows the reconstruction results. Right: In addition, a 3D preview is available.

4. Project structure and workflow

The management of a project follows a strict structure, which allows the software to work with many datasets, system-matrices, sequences, etc. in a highly flexible way.

File structure

After installation, the main software is located in the installation path, e.g., **..\Program Files\PD OctoView**, with the folder structure:

```
\Documentation
\driver
\icons
\MFS5
\MFS5_projects
\projects\templates\         consists of templates for different devices
\Win64\Release\              consists of executables
```

The log-files can be found at two different locations:

```
..\AppData\Local\Pure Devices\MPI\logs          …high level log
..\Program Files\PD OctoView\Win64\Release      …low level log
```

Hint 4-1: It has to be mentioned that the latter log file cannot be created when the user does not have administrator rights.

At the first start of OctoView, the software creates a new project structure in the users "**My documents" folder**:

```
..\my documents\OctoView\projects\
```

with pre-defined template-projects for specific devices. This folder is the default project path for OctoView.

Hint 4-2: A OctoView project can be located outside of this path. The sub-structure has to be kept.

The project sub-folder structure of the selected scanner-project is visualized in Table 4-1.

Table 4-1: Overview of the directory and sub-directory structure of an Octoview scanner-project.

```
- project-scanner folder <scannername1>\
      - sub-folder RCC\
            - <RCC>_freq.dbl          → frequency info
            - <RCC>_abs.dbl           → absolute values [frequ]
            - <RCC>_phase.dbl         → phase values [frequ]
            - rcc.txt                 → additional information
      - sub-folder SM\
            - sub-subd-folder <SM1>\
                  - <SMname>.sm       → system matrix values
                  - <SMname>.3dt      → 3D mapping info
                  - <SMname>.int      → system value ordering
                  - <SMname>_order.int      → orig. system value ordering
                  - <SMname>.txt      → system matrix info
            - sub-sub-folder <SM…>\
      - sub-folder <sequence1>\
            - sub-sub-folder <experiment1>\
                  - <dataset1>.dbl    → time data values [samples]
                  - <dataset1>.txt    → time data info
                  - <dataset2>.dbl
                  - <dataset2>.txt
                  - …
```

```
                - exp.txt            → experiment info
                - expinfo.txt        → experiment notes
         - sub-sub-folder <experiment…>\
         - sub-sub-folder <reco1>\
                - reco.txt           → reco settings info
                - recoinfo.txt       → reco notes
                - optional: MPSdata-files
         - sub-sub-folder <reco…>\
                - sequence.txt       → sequence settings info
                - sequenceinfo.txt   → sequence notes
  - sub-folder <sequence…>\
  - scanner.ovp                      → project info
  - scannerinfo.txt                  → project notes

- <next scannername…>\
```

File types

There are different types of proprietary files used within this software (see specific information in **_datatypes.pdf**):

- **Data set file** (*.dbl) – ***Binary file***, no header
  Format for 1D data array: consists of *N* double values $N \times$`double(8Byte)`
- **Scanner file** (OctoView project file: *.ovp) – ***ASCII format***
  consists of several settings
- **System matrix files** (*.sm; *.3dt; *.int)
  For managing the system matrix data, different types are used:
  - Info-file (*.txt) – ***ASCII format***
  - System matrix values (*.sm) – ***Binary file***, no header $M \times N \times$`double(8Byte)`
  - Index file (*.int) – ***Binary file***, no header $N \times$`integer(4Byte)`
  - 3D mapping file (*.3dt) – **Binary file**, no header $M \times 3 \times$`double(3×8Byte)`
- **COMPASS data set** (*.c4da) – ***Binary file***
  4D complex array with 4 Byte Integer header

Config file

After the first session with the OctoView software, a `config.ini` file is created in the project folder (users "**My documents" folder**: `..\my documents\OctoView\projects\`) of this software consisting of several setup parameters for the software:

```
OctoView config file
version=1.0.5
[parameters]
temperature-control=0
server-autostart=1
autosave-reco=1
timer-interval=50
server-hidden=0
server-starting-attempts=10
show_experimental_features=0
```

Temperature-control (Bool[0/1]):	If `true` (=1) and connected to a device, the software asks every 2 seconds for the current temperature of the system.
Server-autostart (Bool[0/1]):	If `true` (=1) then try to start the MPI_Server.exe automatically.
Autosave-reco (Bool[0/1]):	If `true` (=1) then every change of reco settings will be saved automatically.

Timer-interval (Integer): Sets the polling interval of the internal timer to a specific value in milliseconds.

Server-hidden (Bool[0/1]): If `true` (=1) then try to start the MPI_server.exe in hidden mode.

Server-starting-attempts (Integer): Number of attempts to connect to a server (Matlab or executable based).

Show experimental features (Integer): If `true` (=1) then experimental features are available within the software.

Add new scanner-project

To start with a completely new scanner-project from scratch, first open Octoview. The start tab in figure shows initial information about different pre-defined projects, which can be selected for a quick start.

As visualized on Figure 4-1 left, for an empty project, select the "measurement" button (***see*** Hint 4-3) and the "LAST projects" button and confirm with the ">> GO >>" button.

With that, an empty project is created, and the left bar is now available for performing additional settings.

In the next step, by pressing the "ADD" button in the scanner-project area on the left bar, the software asks for a folder to store the new project. The default folder is the subfolder "projects" in the main directory of this software. By confirming "select folder", a generic scanner-project is created, by default "TWMPI_XX".

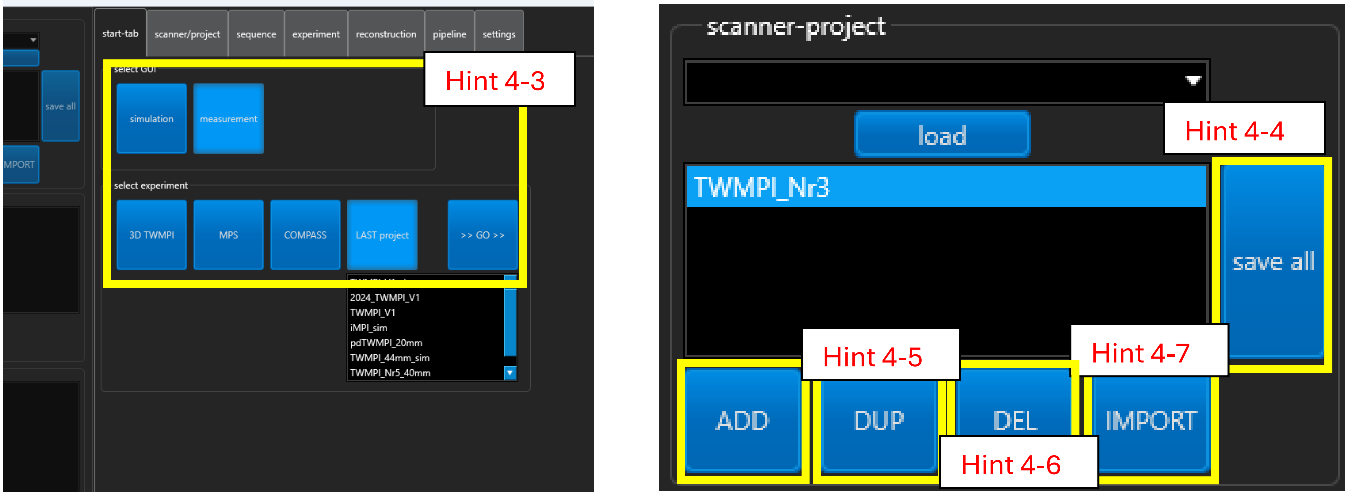


*Figure 4-1: **Left:** first, select measurement and LAST project to create an empty scanner-project. **Right:** overview of "scanner-project" selections.*

In the selected folder, a new subfolder is created with the same name as the new scanner-project (Figure 4-2 left). Additional options for the description and re-naming of the new scanner-project are now available (Figure 4-2 right).

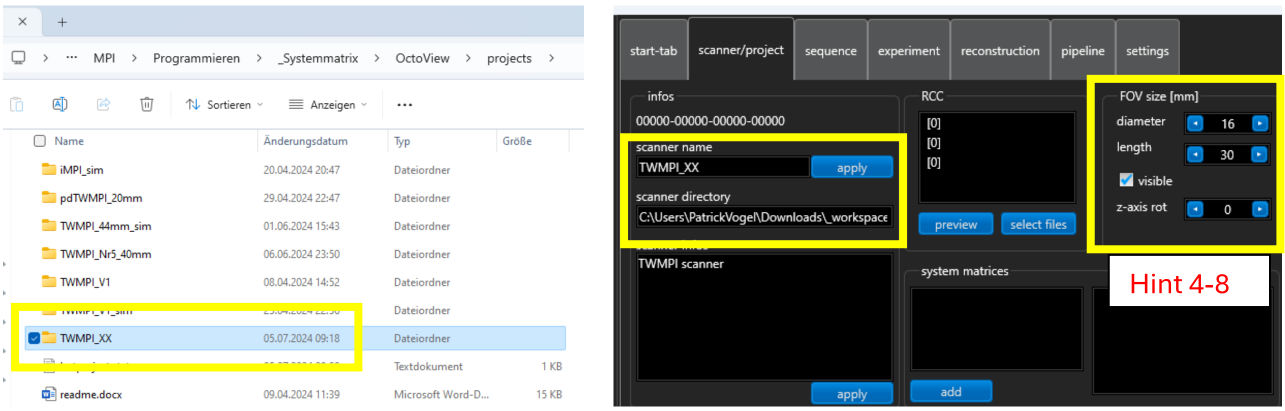


*Figure 4-2: **Left:** Viewing into the selected folder of the scanner-project shows a new subfolder for the new scanner-project. **Right:** The new scanner-project can be renamed (scanner name) and supported with additional information (scanner info). By pressing "apply", the new settings can be confirmed.*

*Hint 4-3 – **Hide scanner-control bar**: In addition to the left bar, the right bar (scanner control bar) also appears. When selecting the "simulation" button instead, the bar keeps hidden. This setting can be changed afterwards on the settings tab (Figure 4-3).*

*Hint 4-4 – **Saving all data:** The software creates and saves new sequences, recos, and experiments automatically in the subfolders of the main directory of the project. However, specific settings, e.g., new* *parameters for a reconstruction, are not saved automatically. For that, the user can use the "save all" button to store the latest settings for a selected project.*

*Hint 4-5 **– Duplicate an existing scanner-project:** By pressing "DUP", the selected scanner-project is duplicated. This means, all scanner settings as well as all sequences and reconstructions are copied **WITHOUT** data sets.*

*Hint 4-6 **– Deleting an existing scanner-project:** By pressing "DEL", the selected scanner-project is removed from the list. By default, all scanner-project information is backuped within a zip-archive in the project folder.*

*Hint 4-7 **– Import an existing scanner-project:** By pressing "IMPORT", the software asks for the folder of an existing scanner-project. By selection, the entire folder (and substructure) is copied into the default projects folder of this software.*

*Hint 4-8 **– Size of the scanners' field of view (FOV):** For each scanner, the bore size, size of the field of view (FOV), and the rotation angle of the visualization can be set for the 3D visualization.*

*Figure 4-3**:** In the settings tab area, the right bar (scanner control) can be enabled and disabled.*

*Hint 4-9 **– register OctoView files**: The OctoView specific project files (*.ovp) can be registered in the operating system (OS). This allows a direct execute of *.ovp files, which opens the software.*
**This feature is only available for Windows platforms.*

Add RCC files

Requires

acquired RCC files

To load a new receive chain correction (RCC) for a given scanner into the software, go to the **scanner-project** tab and press the "select files" button (Figure 4-4). In the following, four openfile dialogs opens to select the acquired files in the following order:

1. file consisting of frequency information, e.g., `RCC_frequ.dbl`
2. file consisting of absolute value information, e.g., `RCC_abs.dbl`
3. file consisting of phase value information, e.g., `RCC_phase.dbl`
4. optional: file consisting of additional pick-up loop information, e.g., `RCC_info.txt`

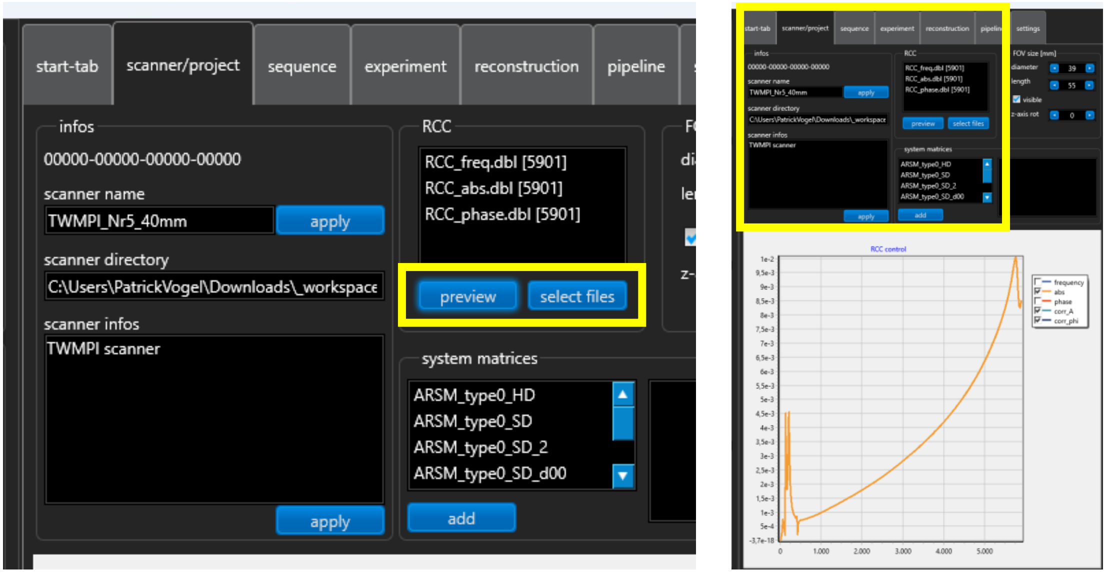


*Figure 4-4: **Left:** zoom into the area for RCC and system matrix selection. **Right:** After loading an RCC data set, a preview of the transfer function (abs value) can be visualized.*

To check the correctness of the RCC files and the transfer function, press the "preview" button to visualize each of the data (frequency, abs-values, phase-values, transfer function) as visualized in Figure 4-4 right.

Add System-Matrix (SM) files

System matrices (SM) are required for the reconstruction process to calculate from the acquired data set the MNP distribution. Since these SMs are scanner specific, the management is performed under the scanner-project tab. To include a new system matrix, press the "add" button in the "system matrices" area. The software asks you to select a valid folder consisting of the required system matrix data. When the selection is valid, the system matrix data are copied into the sub-folder "SM" of the selected scanner-project with the name of the selected folder. To get more information about the system matrix, select the desired SM (Figure 4-5).

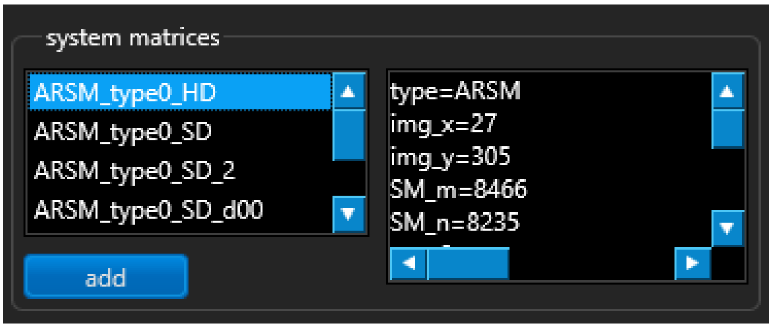
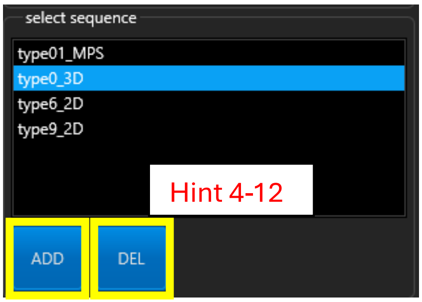


Figure 4-5: When selecting an imported system matrix, the corresponding information can be found in a listbox on the right side.

Add new sequence

For each scanner-project, multiple sequences can be defined. By adding a new sequence for a selected scanner-project, a new sub-folder in the scanner-project structure is created (see Figure 4-6).



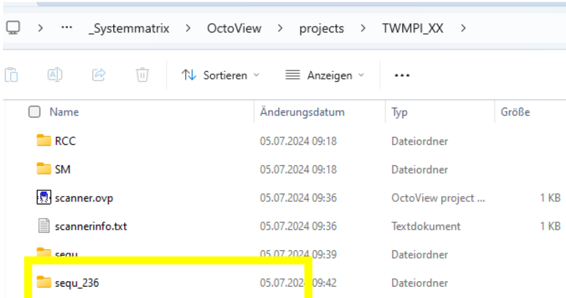
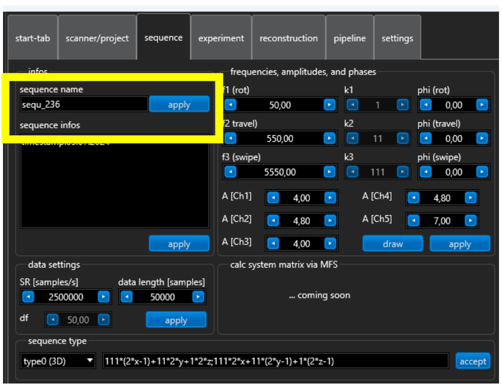


*Figure 4-6: **Left:** Overview of sequence selection. **Right:** For each sequence, a new sub-folder is created.*

The generically created sequence name can be changed on the "sequence tab" for a selected sequence (Figure 4-7).

In addition, several other parameters can be given for the selected sequence:

- **Sequence infos:** For each sequence, specific notes can be given to provide more information.
- **Data settings:** The sampling rate (default: $SR=2.5\cdot10^6$ samples per second) and data length (default: dl=50,000 samples) can be adjusted here. The resulting frequency resolution df can be calculated by $df=SR/dl$.
  ATTENTION: Be careful by setting these parameters since it will have an effect on the entire data-I/O pipeline and reconstruction process.
- **Sequence type:** The TWMPI scanner comes with a high flexibility of using different types of sequences. Thus, in the drop-down menu, multiple sequences can be pre-selected, e.g., type0 (default). For each sequence, a specific formula is generated, which is required for the frequency selection (peak-picking – in section Device principles) for system matrix reconstruction.
  Hint 4-10: Not all parameters are useful or available for real scanner hardware. Thus, in measurement mode, only available parameter settings are possible depending on the hardware.
- **Frequencies, amplitudes, and phases:** For each sequence, multiple waveform parameters can be set. Each TWMPI scanner comes with 5 independent transmit channels (Ch1..Ch5). For each channel, the amplitude in ampere can be set. In addition, the available frequencies can be set (f_{rot}, f_{travel}, f_{swipe}).
  Hint 4-11: Not all parameters are useful or available for real scanner hardware. Thus, in measurement mode, only available parameter settings are possible depending on the hardware.



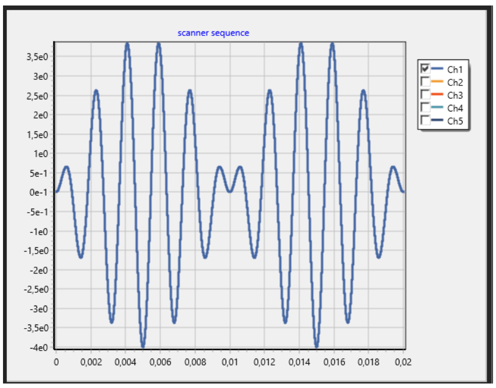


*Figure 4-7: **Left:** Overview of the parameter settings for a selected sequence. **Right:** Example of a waveform preview for channel 1.*

*Hint 4-12 – **Deleting selected sequence:** By using the "DEL" button, the sequence is removed from the scanner-project and backup in the scanner folder as zip-archive.*

Add new experiment

To add a new experiment to a selected scanner-project and sequence, press the “ADD” button in the experiment selection area (Figure 4-8 left). For each new experiment, sub-folders are created consisting of the raw-signals collected from simulation or real measurements.

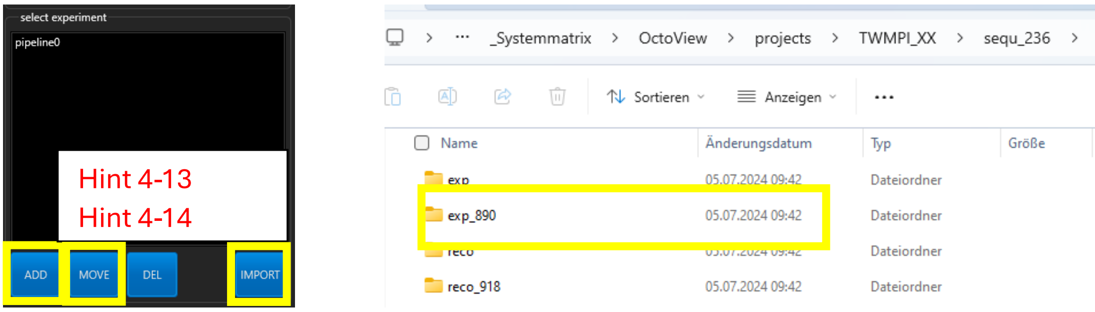


*Figure 4-8: **Left:** Overview of experiment selection. **Right:** For each experiment a sub-folder is created.*

Figure 4-9 shows the experiment tab with additional information about the selected experiment. A rename function as well as a text field for specific notes about the performed experiment are available. On the right side, all collected signal data are presented.

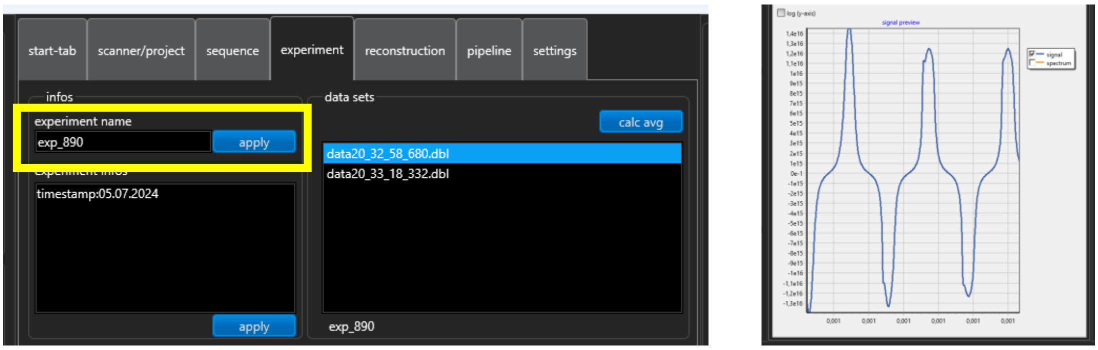


*Figure 4-9: **Left:** Overview of the experiment section. **Right:** visualization of an example data set stored in the selected experiment.*

*Hint 4-13 – **moving or importing an existing experiment:** Use the “MOVE” feature to integrate a complete experiment folder to the sub-structure of the selected scanner-project, where the original data is deleted. Use the “IMPORT” feature to duplicate the selected structure.*

*Hint 4-14 – **deleting an existing experiment:** Use the “DEL” feature removes the experiment from the scanner-project. By default, the data is backup in a zip-archive in the sequence folder.*

Add new reconstruction type

To add a new reconstruction area to a selected scanner-project and sequence, press the “ADD” button in the reco selection area (Figure 4-10 left). For each new reco, sub-folders are created consisting of the information files of the reconstruction process.

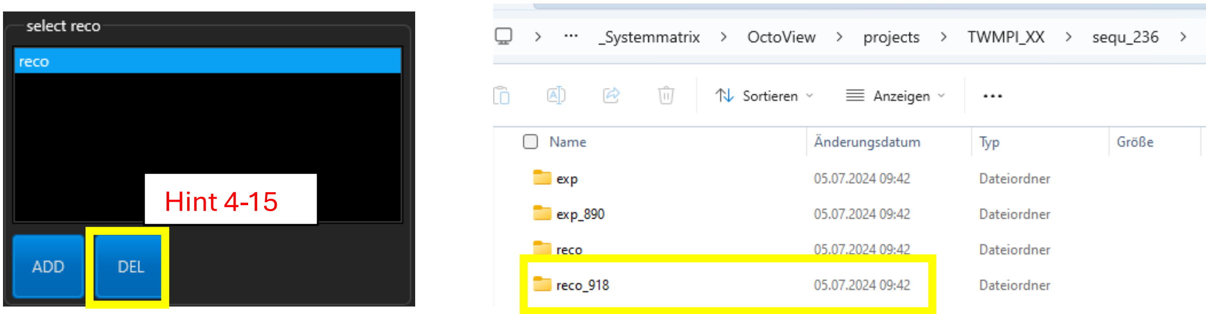


*Figure 4-10: **Left:** Overview of reconstruction selection. **Right:** For each reconstruction a sub-folder is created.*

Selecting a new reco shows the corresponding area for setting up all parameters required for different reconstruction types (see Figure 4-11).

*Hint 4-15 – **deleting an existing reconstruction:** Use the “DEL” feature removes the reconstruction from the scanner-project. By default, the data is backup in a zip-archive in the sequence folder.*

The **“infos area”** can be used to change the generically generated name of the selected reconstruction as well as provide specific information about the data processing.

The **“signal preview”** area defines the from…to values for the visualization of a processed time signal or Fourier signal in the preview area.

The **“signal input”** area manages the data used for reconstruction. Furthermore, some correction steps (baseline correction – BS; receive chain correction – RCC) can be selected and used.

The area on the bottom consists of several tabs:

- **Filtering tab:** The incoming data can be pre-processed using different digital filters. Since most of the filters are based on the specific structure of TWMPI systems, the specific formula of the system matrix can be helpful (see sec. sequence selection).

- **Raw-image tab:** In this area, the setting for creating a raw-image can be set. A raw-image can be used for different reconstruction types, e.g., deconvolution (DeR) or image-based system matrix reconstruction (iSM). Furthermore, it is helpful for calibrating the global phase using a point sample of a given particle system (see section Global Phase Calibration (GPC)).

- **Reconstruction tab:** This area manages the different types of reconstructions:
  - **Deconvolution Reconstruction (DeR):** Direct deconvolution approach uses one deconvolution kernel for reconstruction of the image. This mode requires a calculated raw-image [14].
  - **Transverse Image Reconstruction (TSI):** An image-based reconstruction mode [22].
  - **System matrix reconstruction (SM/iSM):** The system matrix reconstruction mode of this software provides a wide range of system matrix types such as **classic SM approach**, which is based on the peak-picking of specific frequencies (sequence formula) in the Fourier space [18], or image-based system matrix reconstruction (iSM), which uses raw-images for the calculation [19], as well as customized SM methods, which are defined by the naming of the system matrix itself, e.g., GWSM, TWSM, FTSM, FT1F, ARSM, ASM2, TWI1, TWI2, TWI3.
  - **Magnetic Particle Spectroscopy (MPS):** TWMPI scanner can also be used as MPS systems, which means imaging without a selection field (spatial encoding). For that, a specific

visualization environment has been implemented in the software presenting hysteresis curves and more.

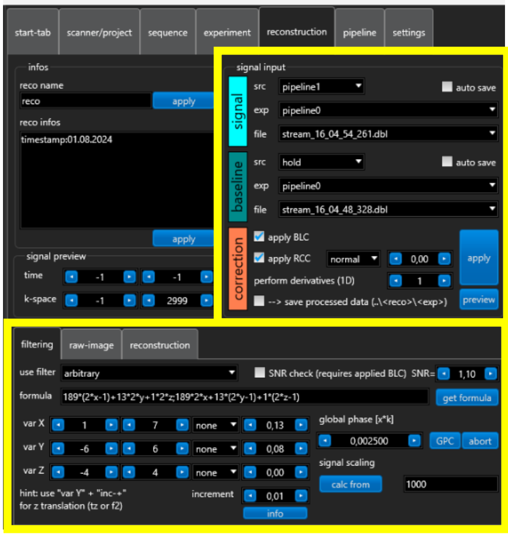


Figure 4-11: Overview of the reconstruction tab providing multiple areas for adjusting parameters for different reconstruction types. The area on the ***bottom*** *provides additional tabs for data processing, raw-image generation, and reconstruction. The area on the* ***top right*** *shows the signal input area defining the I/O pipelines and BLC and RCC processing.*

Signal input area and pipeline tab

The signal input area manages the data input from different sources. The signal part as well as the baseline part come with almost the same features (see Figure 4-12 left):

1. Data source selection: This drop-down menu consists of different source (Figure 4-12 right) such as
    a. **Data**: uses a data set stored in the selected experiment.
    b. **Pipeline 1/2**: uses incoming data from an external source (see pipeline section below).
    c. **Pipeline 1/2 -once** (*baseline only*): takes only the next incoming data from an external source.
    d. **Hold** (*baseline only*): uses the internally stored data set from **pipeline 1/2 -once**.
2. Experiment selection: This drop-down menu consists of all experiments of the selected sequence.
3. Data set selection: This drop-down menu consists of all data sets of the selected experiment.

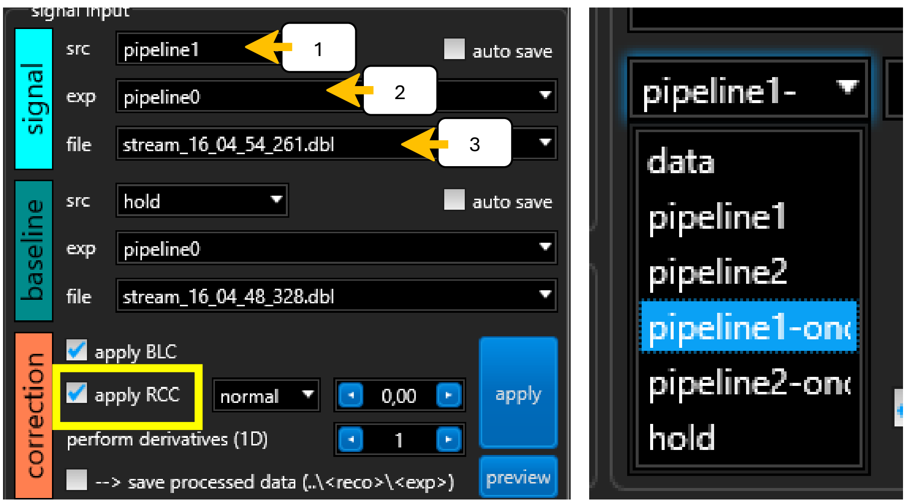


Figure 4-12: Left: Overview of the signal input area: the sources for signal and baseline data can be managed. Furthermore, the RCC correction and signal derivation can be set. ***Right****: The drop-down menu for baseline input consists of different sources for incoming data.*

When selecting a pipeline for incoming data, an additional pipeline tab is available managing I/O data (see Figure 4-13). The areas pipeline 1 and pipeline 2 are used to connect different external sources to acquire data (via drop-down menu):

- **LeCroy**: handling incoming data structures from Teledyne LeCroy digital oscilloscopes via activeDSO library.
- **MMF**: The memory mapped file (MMF) is and interface for the easy and rapid sharing of data between software packages on the same operating system (OS).
  Here, the MMF pipeline is a **client-only**.

Hint 4-16: the source name for MMF files is unique and case sensitive. For each MMF server only one MMF client can be connected.

In addition, there is a TCP/IP client available, which handles communication between the scanner interface (via Matlab) and this software.

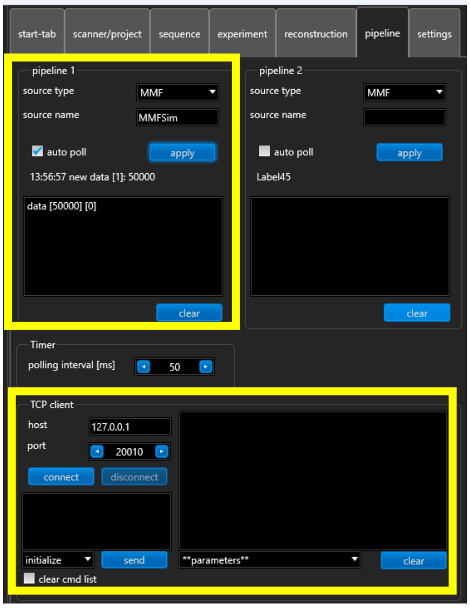


Figure 4-13: Overview of the pipeline area: two separate pipelines can be managed with different sources such as MMF. In addition, a TCP/IP client is available for communication with a real scanner interface.

Examples:

Scenario 1: handling data from a simulation environment.

When using a simulation environment such as MFS5, the generated data sets from the simulation environment can be transferred via a MMF pipeline. In the case of MFS5 usage, select "**pipeline**" and define an MMF interface (pipeline tab). The default MMF name is "MMFSim".
Select also an experiment, which can be used as storage here.
Normally, the baseline area, the RCC, and derivation section are not required here.

Scenario 2: handling stored data sets.

To work with stored data, select "**data**" for the signal area, a desired experiment, and a specific data set. If required, repeat these steps for the baseline area. By checking the "use baseline", the baseline correction (BLC) is activated using the selected data set in the baseline area.
For real scanner data, normally a receive-chain correction (RCC) must be performed. By selecting "use RCC", the stored RCC information of the scanner (see scanner/project tab) is used for correction.

Hint 4-17: When using the RCC, please also keep in mind that at least one derivation must be applied on the incoming data (see section Receive Chain Correction (RCC)).

*Hint 4-18: All parameter changes must be confirmed with the "**apply**" button.*
*Hint 4-19: Use the "**preview**" button to force further processing steps such as filterin, raw-image generation, and reconstruction.*

Data processing structure

The data processing system provides a highly flexible environment for data management, data processing, and visualization. Figure 4-14. show a flow chart about the data processing structure [23].

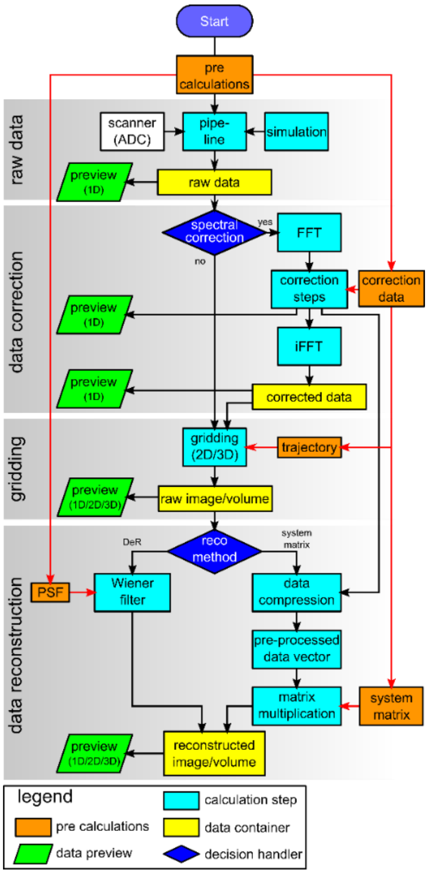


Figure 4-14: Overview of the data I/O processing structure: the data processing can be separated into four sections: raw data, data correction, gridding, and data reconstruction.

Multiple data structures can be pre-calculated to ensure a rapid data processing and visualization with low latency [23].

Raw data section: *The pipeline is the entry point for handling incoming data from different sources.*

Data correction: *Depending on the requirements, the raw data is corrected, filtered, and prepared for further processing.*

Gridding: *When using image-based reconstruction methods, raw images have to be prepared.*

Data reconstruction: *This section shows the reco steps required before data visualization.*

Filtering tab

This tab allows the selection and set up for different digital filtering and data manipulation. For this, several filter templates area available in the drop-down menu:

Arbitrary: This filter uses the a-priory knowledge of the scanners' frequency to 'pick' exactly the higher harmonics of the expected received signal (see section From MPI signal to the image).
Use the **formula** area to define the picking process. This is the same formula, which can be find in the sequence tab (also available through the "**get formula**" button).
Each formula can consist of multiple calculations separated by ";". Each calculation can consist of up to three independent variables: x, y, z. For each variable, a specific value range can be defined the variable runs through.

Hint 4-20: With the additional drop-down menu and the spinbox, different ways of phase manipulation can be applied on each harmonic. E.g., changing the phase on the Y parameter with the "inc+-" selection, the data can be shifted along the z-axis.

Hint 4-21: The numbers given for the frequencies in the calculation parts are the k-parameters (indices) in the corresponding k-space of the data set. It can be calculated by

$$k_f = \frac{f}{df}, \qquad with\ df = \frac{samplingrate}{datalength}$$

Arbitrary & remove kf3 harmonics: This filter template works the same way as the "Arbitrary" filter template but with an additional filtering of all higher harmonics of the swipe frequency 3 (f3 (swipe) – *n**kf3).

Global phase area:

Here, the global phase correction can be applied. By changing this parameter, the data set is manipulated in a way that the raw-image is moved along the x-axis. In other words, with that parameter, the phase of a real data set can be adjusted to a model-based system matrix reconstruction.

Hint 4-22: Use "GPC" button to find automatically the best phase setting for a given sample. Keep in mind, that for this correction-test, point-like samples are required (see chapter 6).
In addition, on the settings tab, the searching parameters can be adjusted.

Signal scaling area:

Sometimes, the data scaling is quite low, which can have an influence on the visualization and preview of the raw data.

Hint 4-23: Scale up the data also have influence on the 3D visualization when the "normalize" checkbox on the reconstruction tab is deselected.

SNR check:

The signal-to-noise (SNR) is a filtering features, which provides the possibility to filter out signal with an SNR factor lower than set up. This feature requires an applied filter as well as an acquired BLC.

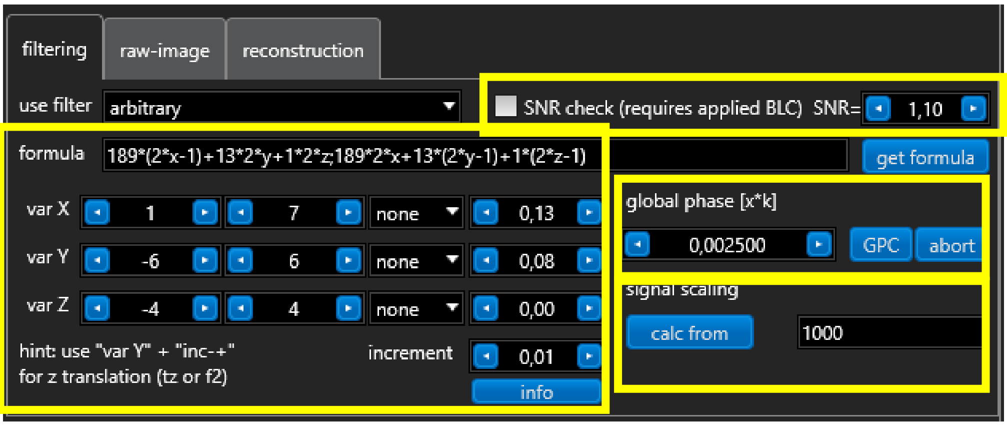


Figure 4-15: Overview of the filtering tab. Different filter templates are available, which can be adjusted by multiple variables. ***Right****: In addition, a global phase and scaling can be applied on the data.*

Raw-image tab

This step is an optional feature, which provides a generation and 2D visualization of raw-images. For that, the acquired data were gridded along the sequence specific trajectory (see chapter 3). In Figure 4-16, an overview of the raw-image tab is given. It provides multiple parameters from the definition of the raw-image size (**img size**) in pixel, the range of the input data (**data from…to**) in samples, the **filling mode**, which defines the type of interpolation (*see pixel density increase*), the frequencies used for gridding (**f1/f2**), and optional image-based derivation features along x- and y-direction. The "**dummydata**" can be used to check the pixel density of a given frequency ratio.

Hint 4-24: Use dummydata and no filling mode to visualize the gridding process on the raw-image.

On the right side of the raw-image tab, the "**folding sgn**" feature provides three types of data fusion: none, foldX, foldX+Y. With given signs for each quarter of the raw-image, the parts are folded together to increase the pixel density and also the SNR. The "**datadelay**" moves the starting point of the beginning of the gridding process (default is top left corner). The "**offset X/Y**" values are defined between 0 and 1 and displace the position of the raw-image in x- and y-direction. The feature "**real/pos**" forces the values for visualization in the

range between 0 to max(). The "**add derivation**" value-box defines additional derivations on the data before raw-image generation (this does not have effect on the k-space data).
The raw-image can be used as an input signal for specific system matrix reconstruction methods (image-based system matrix) as well as for direct deconvolution approaches such as DeR or TIR.

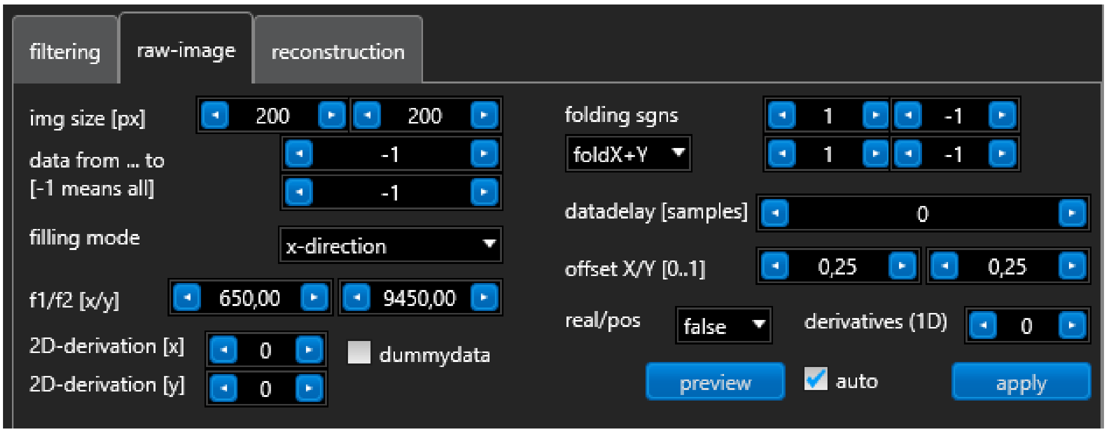


Figure 4-16: Overview of the raw-image tab. This area provides several parameters for the specific generation of raw-images for control purposes or image-based reconstruction methods.

Reconstruction types

In this area, different types of reconstruction and visualization methods can be selected and adjusted. Depending on the selected type, different helper windows are used for visualization. In addition, each type requires different pre-processed raw-data.

System matrices (SM)

System matrix (SM) based reconstruction uses a pre-defined (model-based or measured) system matrix for the calculation of the MNP derivation. For approximation of the inverse of the system matrix, the SVD approach for calculating the Moore-Penrose pseudo-inverse, an iterative Kaczmarz algorithm as well as an implementation of a fast non-negative leasts squares (NNLS) approach is available.
In the SM area the desired system matrix can be selected, which is pre-loaded for each scanner/project (see scanner/project tab). Multiple parameters can be adjusted for reconstruction process:

- iterations sets the amount of iteration steps.
Hint 4-25: be careful to use a high number depending on the size of the used system matrix.

- lambda sets the regularization parameter.
Hint 4-26: Normally, this parameter can be set between 10 down to 1e^{-5}. Since the system matrices normally by pre-sorted by their eigenvalues, a high number results in a smoothed image with less information, where a low number provides highly resolved images but with a noise level.

- threshold sets the amount of used system functions of the system matrix.
Hint 4-27: The system matrix normally is pre-ordered by the strength of the eigenvalues. Thus, withincreasing numbers, more system functions are used for the reconstruction process, but also more possible noise is added (default number is -1, which means using all available system functions). To increase calculation speed, reduce the number.

- force real forces the algorithm to avoid negative (non-physical) values
Hint 4-28: This feature can be helpful to increase the contrast for a given reconstruction.

- normalize normalize the output data
*Hint 4-29: the default setting is CHECKED. To compare different measurements, this feature can be disabled, which can result in an unbalanced visualization. To solve this issue, in the “**filteringtab**”, the scale value can be adjusted.*

- log-view weights the output data in a log-scale
*Hint 4-30: When the data looks like a step-function, increase the scale value on the “**filtering tab**”.*

Two different types of system matrices are available for reconstruction:

- Image-based system matrix (TWI1, TWI2, TWI3)
- k-space-based system matrix (ARSM)

Depending on the selected type of system matrix, the raw-image has to be adjusted in an appropriate way in a prior step.

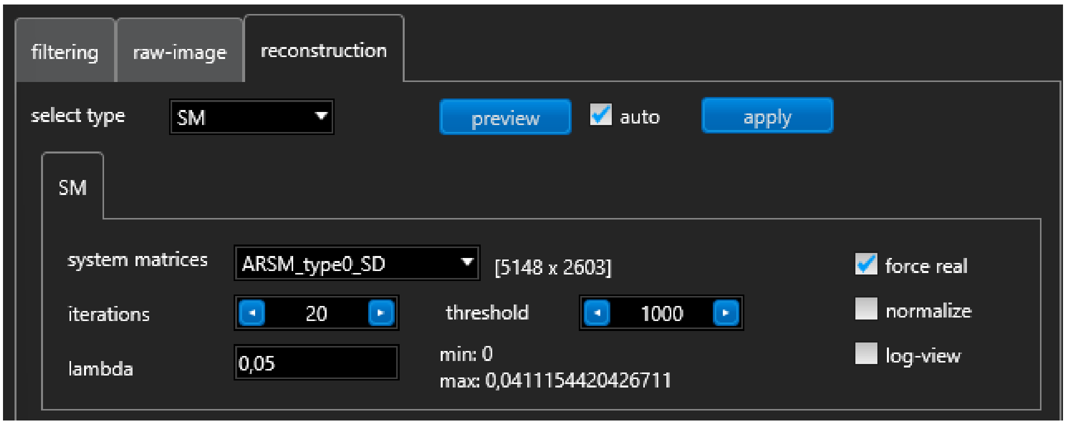


Figure 4-17: Overview of the reconstruction area for system matrix (SM) reconstruction and visualization.

The preview window “3D visualization” pops up when all information is available for visualizing the 3D reconstruction (see Figure 4-18). The helper window provides a fully 3D preview, which can be directly manipulated by mouse gestures:

- hold left mouse button for rotation
- hold right mouse button for translation
- scroll mouse wheel (or two-fingers gesture on touchscreens or mousepads) for scaling

The helper window allows the visualization of multiple project volumes. To get rapid access to the volume select the desired volume from the listbox. The 3D data are visualized by using a marching cube algorithm (MCA) to convert the volume data into a 3D surface rendering model [24].
For each selection, several parameters are available for set up the visualization:

- **visible** Turn on/off the 3D surface rendering of the volume.
- **smooth/MT** Use a smoothing algorithm for recalculating the surface model.
  MT=use multithreaded calculation
- **threshold** Sets the threshold for MCA.
- **color picker** Sets the color for the selected volume.
- **opacity** Sets the opacity for rendering.
- **3D interpolation** Applies an interpolation algorithm to increase the 3D points.
  ATTENTION: the number of voxels increase by n^3.
- **3D shell remover** Clip information beyond the radius.

Furthermore, the possibility to save screenshots as well as 3D NIfTi (Neuroimaging Informatics Technology Initiative) files [25] is possible. For the latter, several options are available:

- **ask override** Ask when selecting an existing file.
- **just override** Saves the new file without asking.

- **append single** Append the selected volume as new 4th dimensional entry.
- **export array** Export a 4D array of the entire selected experiment list.

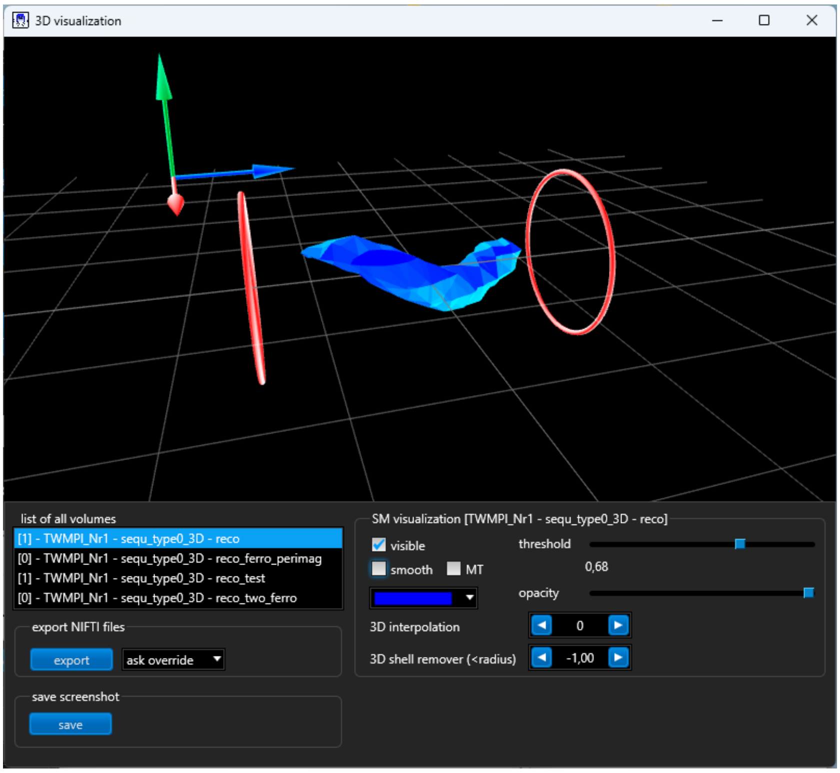


Figure 4-18: Overview of the helper window 3D visualization: the listbox on the left allows the rapid selection of all volume visualizations in this project. On the right, several parameters for 3D visualization can be adjusted.

Magnetic Particle Spectroscopy (MPS) – Hysteresis

Magnetic Particle Spectroscopy (MPS) can be seen as a 0-dimensional imaging experiment. MPS is used to characterize MNPs [26]. As mentioned in the basics section of this manuscript, without a selection field (gradient), the specific information of the MNP can be acquired directly in form of odd higher harmonics of the excitation signal.
In addition, the hysteresis of the MNP can be calculated and visualized. The hysteresis provides the magnetization $M(t)$ of the MNP against the external magnetic field $B(t)$. For that, a ground truth magnetic field data set with correct phase has been acquired in a prior step, e.g., by using a Dysprosium sample.
The selected magnetic field signal is normalized and with the knowledge of the calibrated scanner, the amplitude of the magnetic field can be calculated (x-axis).

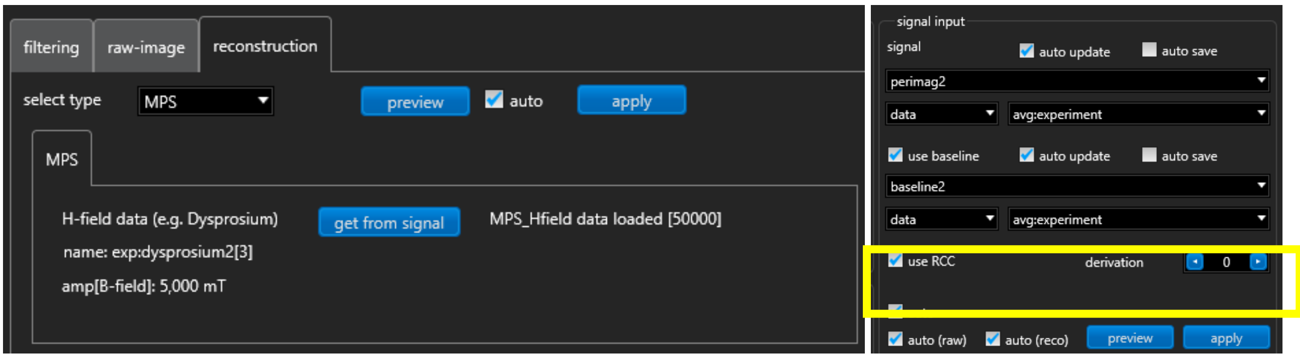


Figure 4-19: Left: Overview of the MPS tab. For a selected Magnetic field data set (e.g. Dysprosium sample), the Hysteresis can be calculated and visualized. Right: The RCC provides the magnetization of the signal, which is here used for Hysteresis.

The helper window “Hysteresis” provides a preview of the calculated hysteresis of the sample (see Figure 4-20).

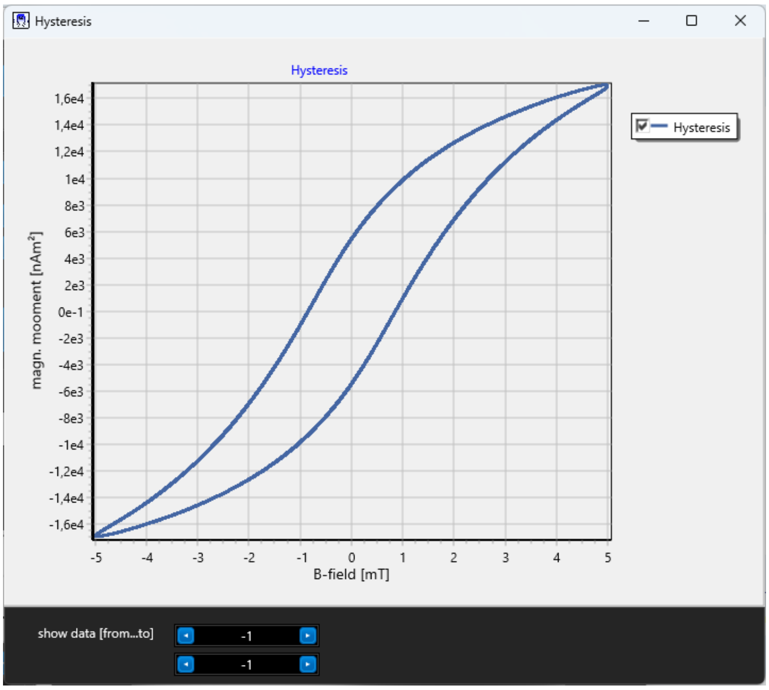


Figure 4-20: Overview of the helper window providing a visualization of the hysteresis.

Hint 4-31: It would be recommended to use a Dysprosium sample, which provides a highly linear signal [27].

Hint 4-32: Since the RCC provides after correction directly the magnetization, no additional derivation is required (see Figure 4-19 right).

Transverse Signal-based isotropic Imaging (TSI)

This reconstruction type is based on a threshold-like method to visualize the distribution of MNPs in 2D [22]. The TSI tab provides several parameters for pre-process the incoming raw-images (Figure 4-21).

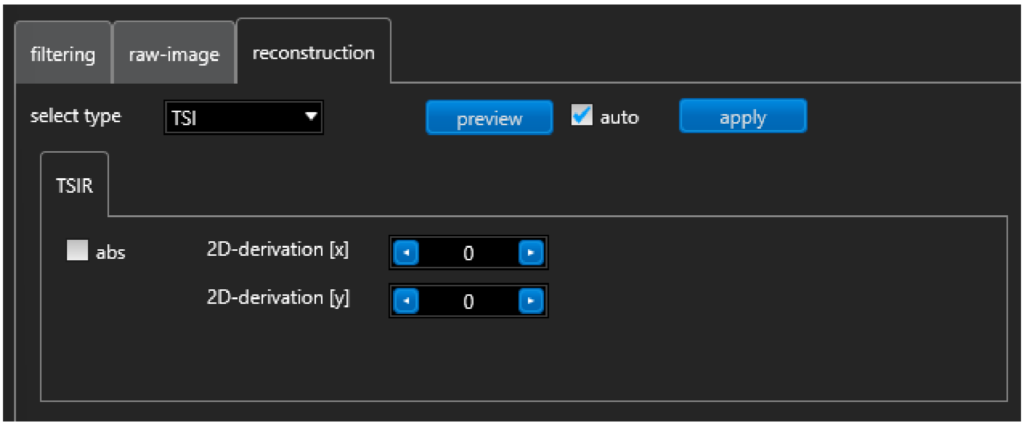


Figure 4-21: Overview of the TSI reconstruction method.

As indicated in Figure 4-22, the idea is based on the second PSF type [28]. For that, after generating the raw-image, an integration followed by derivation along the x- and y-direction is performed.

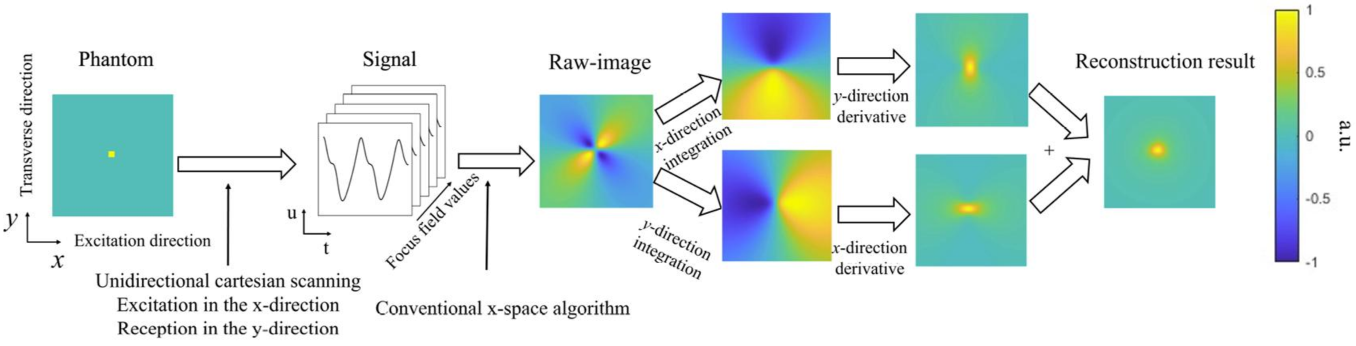


Figure 4-22: TSI workflow taken from [22].

In Figure 4-23, an example for PSF type 1 (top) and PSF type 2 (bottom) is shown. For the latter, a prior derivation along the x- and y-direction is performed to 'generate' a similar PSF type 2 raw-image.

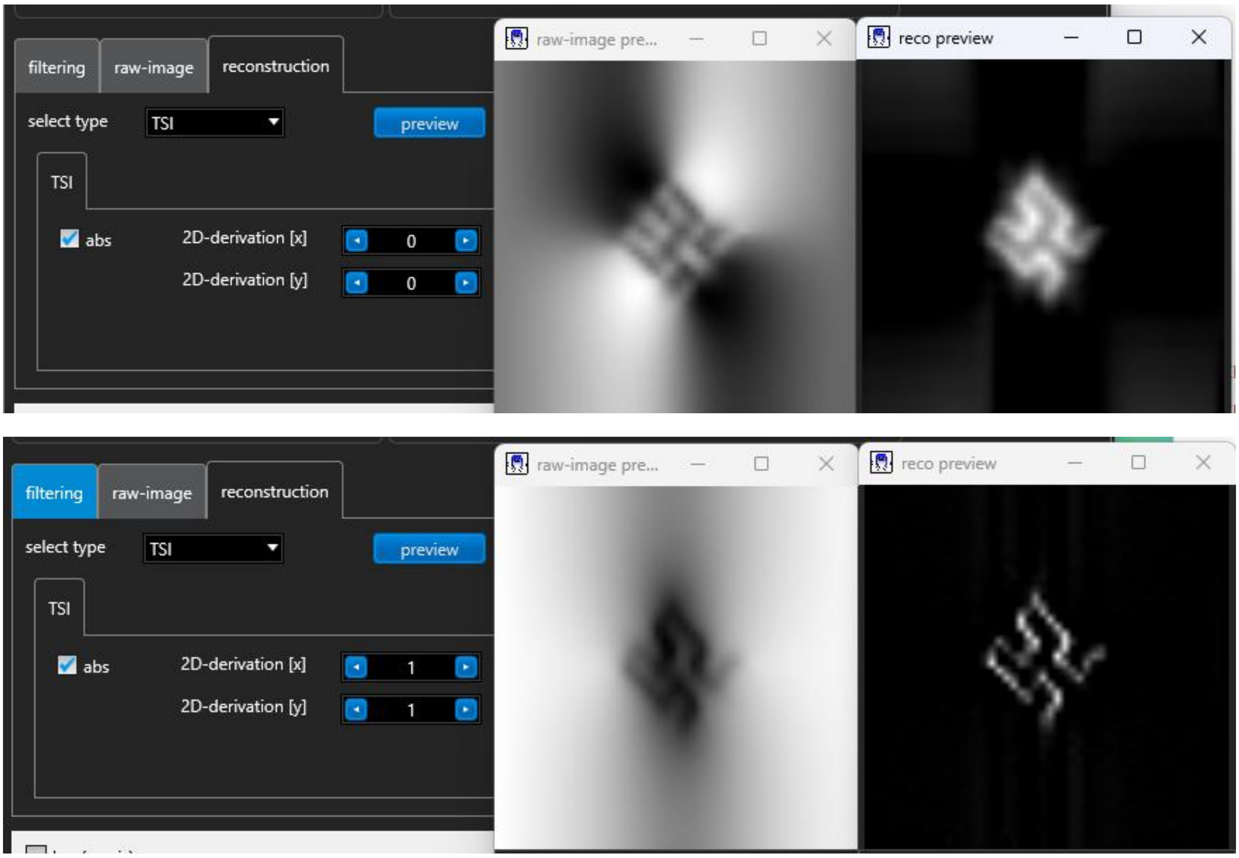


Figure 4-23: Top: TSI reconstruction using a PSF type 2 raw-image. Bottom: TSI reconstruction using a PSF type 1 raw-image.

Deconvolution Reconstruction (DeR)

This reconstruction type is not available in this software version.

5. System-calibrations and -corrections

Often it is necessary to re-calibrate the MPS or MPI systems to overcome distortions within the transmit- and/or receive-chains resulting from environmental parameters, e.g. temperature change. Furthermore, the characterization of a system is desired to calibrate e.g. the transmit/receive part to a specific magnetization.

The OctoView software provides a variety of different calibrations and correction methods to ensure full control of the scanner's parameter.

Correction protocols:

- **BLC**: **B**ase**L**ine **C**orrection — useful before each new measurement
- **DPC**: **D**ysprosium **P**hase **C**orrection — useful before each MPS experiment
- **AFC**: **A**ctive **F**eedthrough **C**orrection — useful before each new measurement day
- **GPC**: **G**lobal **P**hase **C**orrection — useful before each experiment with new MNP system
- **RCC**: **R**eceive **C**hain **C**orrection — once a year or when hardware has changed
- **SMC**: **S**ystem **M**atrix **C**alculation — required after sequence parameter has changed (frequency, etc)
- **SFC**: **S**mart **F**requency **S**election — useful after frequency has changed
- **DPC**: **D**ot **P**roduct **C**orrection — phase correction for COMPASS graphs

Calibration protocols:

- **CPC**: **C**hannel **P**hase **C**alibration — useful before new measurement day
- **ARC**: **A**utomatic **R**x **C**alibration — useful before long sequences
- **MFC**: **M**agnetic **F**ield **C**alibration — once a year or when hardware has changed (RCC validation)

Baseline Correction (BLC)

Devices: TWMPI, COMPASS, MPS

Special requirements: none

The baseline correction is the easiest way to get rid of distortion signal of environmental sources, such as additional devices in the lab, as well as internal sources such as distortions from the used amplifiers. Only synchronized non-stochastic signals can be corrected with a BLC.
The correction can be performed by simply pressing the BLC button in the scanner control area (see Figure 5-1 left). Before BLC all samples should be removed from the scanner.

Step 1: remove all samples from the scanner and perform the BLC → baseline-signal.

Step 2: in a postprocessing step, this baseline-signal can be subtracted from upcoming experiment-signals to remove environmental signals (see Figure 5-1 right).

*Hint 5-1: This correction step is directly available via the scanner control area. When the software is connected to the scanner, the system automatically uses "**pipeline1-once**" for the acquisition of the next baseline signal. Make sure that no sample is placed into the scanner.*

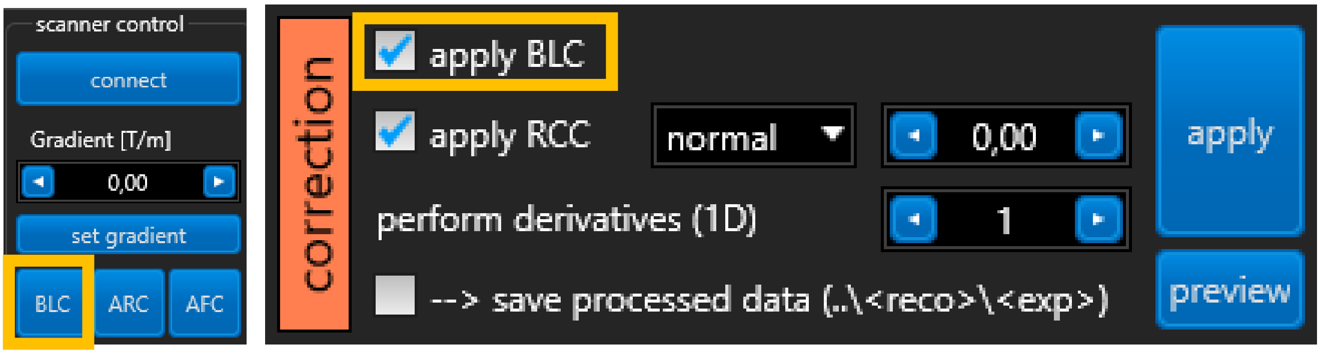


Figure 5-1: Left: The BLC button can be found in the scanner control area. Right: By applying BLC, the selected baseline signal is subtracted from the experiment signals.

Active Feedthrough Correction (AFC)

Devices: TWMPI (COMPASS & MPS: coming soon)

Special requirements: none

The signal generated by the sample is multiple decades smaller than the excitation frequency in an MPI scanner. To get rid of the feedthrough signal from the transmit coil direct into the receive coil, several approaches are available such as passive filtering (high-pass) or gradiometric coil designs. The latter uses at least two receive coils, which are oriented in a way that the induced signal from the transmit coils cancels out each other. A sample positioned near one of the coils now induces their signal in only this coil. With that design, a suppression of one excitation frequency can be realized.
However, gradiometric coil designs can be influenced by the loading effect of the sample and/or temperature effects, which increases the passing of the signal from the transmit coil. To avoid a mechanical adjustment before a new experiment, an electrical way is implemented in the TWMPI scanners.
For that, a second transmit channel sends an inverted signal, which is acquired in a prior step, onto the receive chain to suppress the excitation signal within the analog-digital-converter (ADC).
The AFC protocol can be easily applied by pressing the AFC button in the scanner control area (see Figure 5-2 left).

Hint 5-2: This feature is available via the scanner control area. By pressing the "AFC" button, the system starts the automatic measurement of the feedthrough of the receive chain.

***ATTENTION**: when the AFC goes wrong, the gradient strength can be too high. **Solution**: Reduce the gradient strength for this correction measurement.*

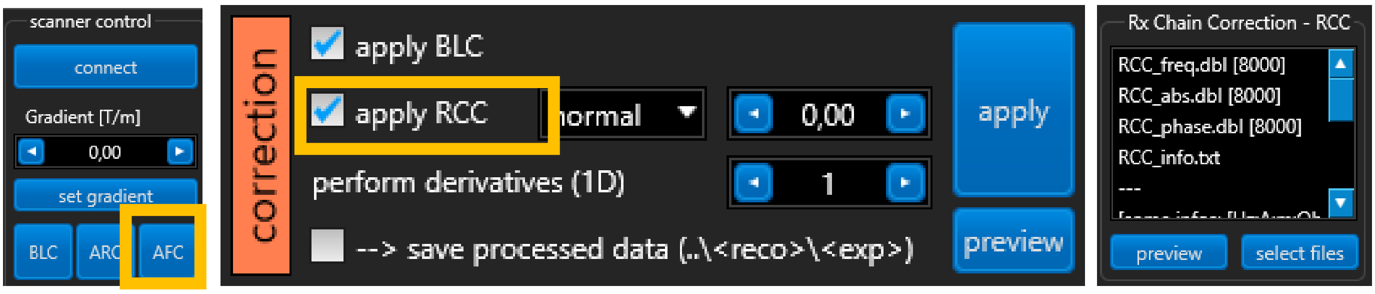


*Figure 5-2: **Left**: The AFC button can be found in the scanner control area. **Middle**: By applying RCC, the selected RCC data are used for calculating the transfer function of the RX chain. **Right**:*

Receive Chain Correction (RCC)

Devices: TWMPI, COMPASS, MPS

Special requirements: 1×Pick-up loop with Y-cable
1×XLR to SMB cable
1×XLR to 2xSMB Y-cable

The receive chain of a scanner consisting of several parts can cause frequency-dependent distortions in amplitude and phase. For a high-quality reconstruction result, this information must be measured in a prior calibration phase.

The use of RCC data sets can be applied in the correction area on the reconstruction tab (see Figure 5-2 middle). These data sets can be selected in the RCC area on the scanner/project tab (see Figure 5-2 right).

Hint 5-3: The RCC does not usually have to be made. Only in special cases such as changing of the cabling or device specific hardware components it is necessary.

The default parameters for this method are given to:

Amplitude	A_{RCC}=TX-value	parameter:	`rcc_Up`
Low frequency	f_{low}=50 Hz	parameter:	`rcc_flow`
High frequency	f_{high}=400,000 Hz	parameter:	`rcc_fhigh`
Frequency step	df=50 Hz	parameter:	`rcc_df`
Pickup radius	radius=0.0019	parameter:	`rcc_pickup_radius`
Pickup resistance	resistance=47 Ohm	parameter:	`rcc_pickup_resistance`
Pickup windings	N=10	parameter:	`rcc_pickup_N`
Pickup current	I_p=0.075 A	parameter:	`rcc_pickup_Ip`
Pickup radius	radius=0.0019 m	parameter:	`rcc_pickup_radius`
Pickup RX_Uin_max	rx_Uin_max=0.2V	parameter:	`rcc_pickup_RX_Uin_max`
Pickup outdir	dir=<temp>	parameter:	`rcc_output_dir`

The parameters represent the pickup coil parameters, which are required for calibrating the magnetic moment of the scanner.

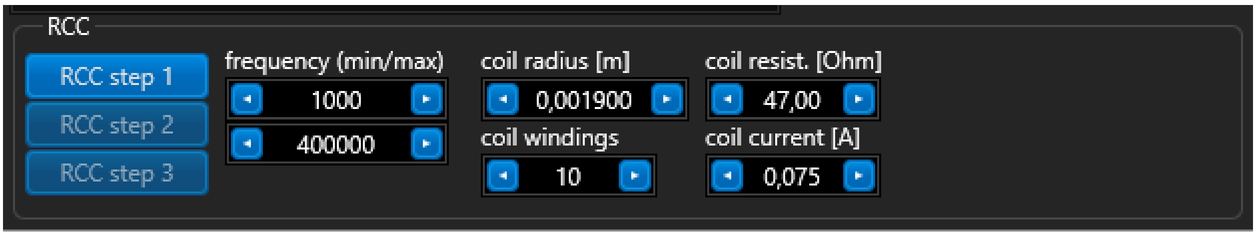


Figure 5-3: Screenshot of the RCC calibration area with buttons to perform step-wise the three steps and adjust the parameters.

The correction is separated into **three** steps, which can be performed by pressing the RCC buttons in the settings tab (see Figure 5-3).
However, everything is automated and the user only have to follow the instructions of wiring and re-wiring the components during the calibration process.

ATTENTION: Please follow the instructions carefully since the device can be damaged.

In the following, the three steps are presented in more details:

Step 1: Console calibration: measures the amplitude and phase between the transmit (TX1) and receive (RX) channel of the drive console.

-Disconnect Tx1, Tx2 and TRx cables from drive, TWMPI scanner (MPI unit) and Splitter-Box.

-Connect the TX1 and RX connector directly with the SMB cable from the pick-up loop and place them in the TWMPI scanner using the 5 mm center-holder (Figure 5-4).

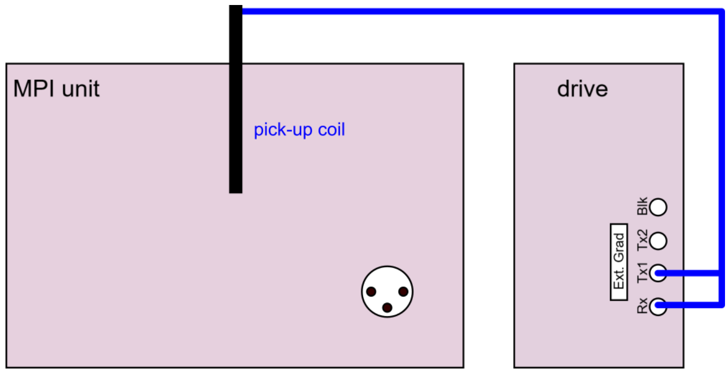


Figure 5-4: Configuration for step 1: disconnect everything from the drive, the MPI unit and the SMB-to-XLR cable to the splitter box. Put in the pick-up coil and connect the Y-SMB-cable to Tx1 and Rx of the drive unit.

-Run following commands:

```
Set_params rcc_fLow 50
Set_params rcc_fHigh 400000
Set_params rcc_Up 0.1
Set_params rcc_df 50
Set_params method rcc_cal1
Start_measurement
```

Hint 5-4: The measurement is performed two times. Wait until Matlab finished the calculation (this can take a while depending on the range of frequencies)! Then close the window and reset the command line. Move on to step 2.

Step 2: Transfer function measurement: measures the amplitude and phase between the transmit (TX1) and receive (RX) over the scanners receive chain.

-Disconnect the TRx connector from the pickup loop cable and connect directly the TRx connector of the drive with the scanners' RX output (Figure 5-5).

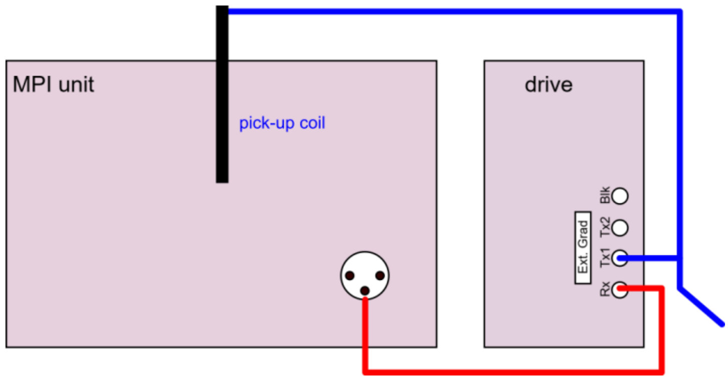


Figure 5-5: Configuration for step 2: disconnect the Rx cable to the pick-up coil and connect the SMB-to-XLR cable between MPI-unit and the Rx of the drive unit.

-Run following commands:

```
Set_params rcc_Up 3.9
Set_params method rcc_cal2
Start_measurement
```

Hint 5-5: Wait until Matlab finished the calculation (this can take a while depending on the range of frequencies)! Then close the window and reset the command line. Move on to step 3.

Step 3: After finishing run, go back to regular measurement mode

-Run following commands:

```
Set_params method TWMPI
```

-Set all the parameters of the pick-up coil and also the measured current (e.g.)

```
Set_params rcc_pickup_radius 0.0019
Set_params rcc_pickup_resistance 47
Set_params rcc_pickup_N 10
Set_params rcc_pickup_Ip 0.075
Set_params rcc_pickup_rx_Uin_max 0.2
```

-Force a data drop out

```
write_rcc
```

-The results of the calibration can be found in the temporary folder, which is

```
get_params rcc_output_dir
```

-Disconnect and remove the pickup loop. Re-connect the XLR-to-SMB cable to the drive TX2 output and the Splitter-Box. Connect the RX output of the TWMPI scanner with the TX1 and TRx connectors of the drive unit (Figure 5-6).

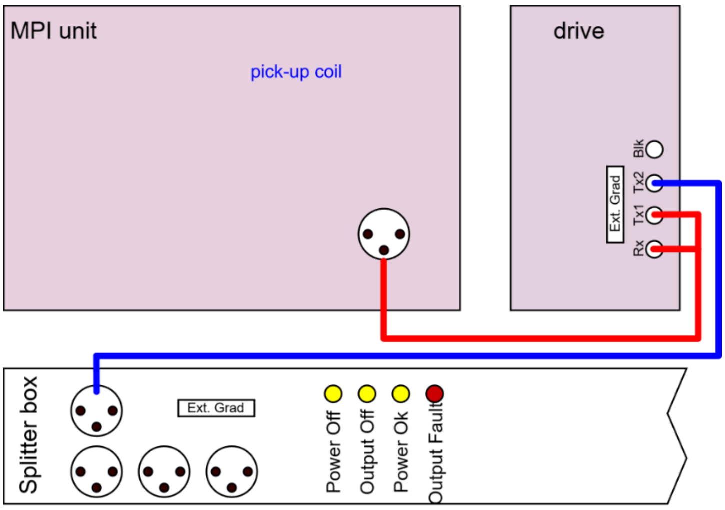


Figure 5-6: Configuration for step 3: remove the pick-up coil and re-connect the drive and the MPI unit.

*Hint 5-6: This feature is available via the reconstruction tab in the signal input area. Activate the "**use RCC**" checkbox to apply automatically the loaded RCC data (see scanner/project tab).*

Hint 5-7: The unit of the abs values of the RCC file is percent/100 of the set RX_Uin_max of the pickup parameter.

ATTENTION: Since the common RCC measurement also corrects the signal induction, which means the derivation of the data, normally one derivation has to be added.

Dysprosium(III)-oxide Phase Correction (DPC)

Devices: TWMPI, COMPASS, MPS

Special requirements: Dysprosium sample (5 mm NMR glass tube)

This kind of correction is used to calibrate the phase between the transmit and receive channel (see GPC).

It is also used as calibration signal for Hysteresis visualization of spectroscopic experiments, e.g., MPS.

Hint 5-8: This feature is available via the MPS reconstruction mode. Use the Dysprosium(III)-oxide sample to adjust the phase for correct hysteresis visualization.

Global Phase Calibration (GPC)

Devices: TWMPI

Special requirements: Point sample of a given particle system

Since the above mentioned phase calibration step using Dysprosium(III)-oxide is mainly used for the phase calibration of MPS experiments, the GPC calibration adjusts the correct global phase for imaging purposes.

Each MNP system shows a specific phase shift, which depends on the amplitude, frequency and temperature. To use the default model-based system matrix reconstruction approach (the default system matrix is calculated with no phase shift), the global phase has to be adjusted on the used MNP phase shift.

Step1: Put in a point-like sample of the desired MNP system and perform a measurement.

Step2: press the "GPC" button (Figure 5-7 left) to execute the automatic phase correction.

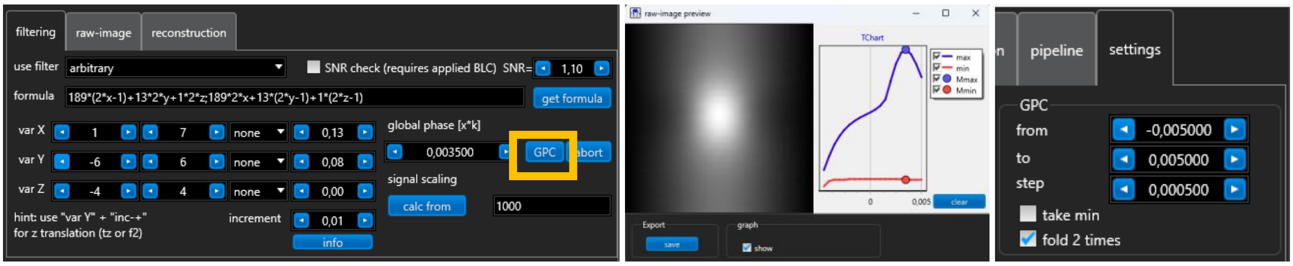


Figure 5-7: Left: In the filtering tab, the automatic GPC feature can be executed. Middle: Result of the GPC calculation: the GPC tries to find the optimal signal within the folded raw-image. Right: Settings for GPC calculation.

The phase calculation is based on overlapping the left&right scan on a raw-image.
There is a pre-defined script available within this software (auto GPC).

Hint 5-9: The GPC feature is directly available via the filtering tab in the reconstruction area. By activating this correction, the systems automatically try to find the optimal value by overlapping (foldX+Y) all four areas of the raw-image.

ATTENTION*: The best results can be achieved using a point-like sample positioned in the center of the scanner and performing a 2D experiment.*

System Matrix Calculation (SMC)

Devices: TWMPI

Special requirements: MFS5 software & Scanner parameters

The system matrix reconstruction, also called 'black-box' approach, is used for calculating the distribution of the MNPs within the scanners' FOV including all MNP and hardware parameters as well as their distortions. As mentioned in section Device principles, the system matrix reconstruction solves the ill-posed problem of determining the inverse of a system matrix. The naming 'black-box' approach comes from the fact that it is possible to measure a system matrix point-by-point using a small point-like sample and a robot. By taking for each possible point in space a data set with the desired MNP system, all parameters, from particle side as well as from scanner side can be determined. Putting all together in a system matrix and calculating the inverse, it represents a direct solution without the requirement of any knowledge of the systems.

However, since it takes a long time, especially for highly-spatial-resolved system matrices, model-based system matrices provide much more flexibility. The SMC is an approach, which uses a virtual copy of the MPI scanner and the given parameters to compute a system matrix for the hardware parameters.
For adjusting the MNP based phase shifts, the GPC approach is used.

Hint 5-10: It has to be mentioned that this approach is just an approximation. For higher quality images, a robot-based measurement of the FOV is mandatory.

For more information about this topic, see additional documentation: **systemmatrices**.

Smart Frequency Selection (SFC)

Devices: TWMPI

Special requirements: none

...in preparation

Dot Product Correction (DPC)

Devices: COMPASS

Special requirements: none

...in preparation

Channel Phase Calibration (CPC)

Devices: TWMPI

Special requirements: none

...in preparation

Auto Rx Calibration (ARC)

Devices: TWMPI, COMPASS, MPS

Special requirements: none

The drive console consists of a receive input with an adjustable pre-amplifier. This is used to amplify the incoming signal to a level where the sensitivity or dynamic range of the analog-digital converter (ADC) is used with the maximum available ENOBs (efficient number of bits) to ensure the best signal-to-noise ratio (SNR).

The automatic receive-chain calibration (ARC) is a feature, which uses a short sequence to check the saturation of the ADC and set up a pre-amplifier level, which guarantees an 80% saturation of the ADC.

Depending on the selected device, the software prepares a specific sequence depending on the given parameters:

TWMPI -take the sequence as it is (**ATTENTION**: averaging values are not affected)
-disable gradient fields (A_ch1 = A_ch2 = A_ch3 = A_ch4 = 0)

COMPASS -set AC-steps to 1; DC-steps to 2, frequency-steps to 1 (shortest possible sequence).
-set the maximum frequency (ACFrequencySteps=1)
-set scansym to 0

Hint 5-11: The user should put in the sample with the maximum expected signal before performing the ARC.

Magnetic Field Calibration (MFC)

Devices: TWMPI, COMPASS, MPS

Special requirements: 1×Pick-up loop

Disconnect TX1 and RX cabling from drive console.

Place a small pickup-coil (5 mm NMR glass tube) with the sample holder in the scanner and connect the cable to the RX of the drive console.

Set up the scanner via TCPIP interface with the following commands:

```
Reset_all_params
Set_params_amptx 1
```

The first command reset all parameters to the default settings, among others set the gradient field to zero. The second command sets the main coil (drive field) to a given value representing an optimal value for 1 T/m (default sequence).
With the given parameters of the pick-up coil, the magnetic field B_{max} can be calculated:

$$B_{max} = \frac{U_{max}}{\pi \cdot N \cdot r^2 \cdot f}.$$

E.g., with the parameters N=10; r=1.9 mm; f=9.450 Hz; U_{max}=57 mV), the maximum magnetic field can be calculated to B_{max}=8.6 mT.

Based on the calculated magnetic field, the scanners' parameters for gradient strength can be adjusted and validated.

Hint 5-12: The U_{max} value is directly available in the uncorrected time signal from the calibrated drive system.

6. Data I/O interfaces

The Octoview software supports two different interfaces:

1. **MMF (memory mapped file)**
   The MMF interface is used to share data sets between different software packages on the same OS, e.g., rapid transfer of generated data sets in MFS5 software to OctoView.
   The MMF protocol requires an MMF name (case sensitive!) and a defined data structure on both sides (source and destination). For data transfer, the structure is defined as followed:

Table 6-1: Memory mapped file (MMF) structure.

struct	type	
tag1	int (4 Bytes)	indicates new data: • last digit toggles between 1 and 0: 0=last data 1=new data • second last digit indicates the data-type: 0=TWMPI data 1=COMPASS data
tag2	int (4 Bytes)	indicates data status
SR	int (4 Bytes)	samplingrate of the data set
data length	int (4 Bytes)	length of the data set
data array	double (*N*×8 Bytes [Int])	data set values *

* The data array is static and pre-defined with a maximum length of 2,000,000 data points.

Hint 6-1: the system checks the length of the incoming data and compares it with the setup data length in the sequences tab.

2. **TCP/IP (transmission control protocol/internet protocol)**
   The TCP/IP protocol is used for operating the scanner. For that, the OctoView software acts as client and connects to a server module (MPI_Server.exe), which directly interacts with the drive console and the scanners hardware.
   The data I/O pipeline uses several commands:

Table 6-2: Command list for server-client communications.

command	**description**
`initialize`	Initialize the MPI scanner: connects to console and forces the starting procedure, pre-testing phase.
`is_initialized`	Check for correct initialization parameters: 0 or 1
`close`	Close the session, switch off the MPI hardware and stops the server.
`query_lasterror`	Check for error on the server's side. As result an error number is provided. parameters: -2 → timeout -1 →no connection to server 0 → no error 1 → generic error 2 → error initializing device

	3 → unknown command 4 → unknown argument 5 → error starting measurement 6 → error transmitting data 7 → no data available 8 → no valid MMF 9 → unknown method
`query_lasterror_str`	Check for error in plain text (see query_lasterror)
`reset_lasterror`	Reset the error memory on servers side.

`set_params`
 	
(*) these parameters can be set by "default" phrase to device specific values, e.g., Set_params <parameter> default
	set a specific parameter:
	`set_params <parameter> <value>`
	special:
	`method {TWMPI, TWMPI_heat, COMPASS, COMPASS_heat, COMPASS_V2, COMPASS_V2_heat, gradio_cal, rcc_cal1, rcc_cal2}`
	→ Example: `set_params method TWMPI`
	
	parameters:
	`channelTX (1/2) (d: 1)`
	desc.: major transmit channel for ch5
	`swiperadius <double> [m]`
	desc.: maximum displacement of the FFL from the center by the swipe frequency
	`AmpTX <int> [mT]`
	desc.: amplitude of RF field (ch5)
	`AmpSwipe <double> [T/m]`
	desc.: deflection of FFL in swipe direction
	`AmpGrad1 <double> [T/m]`
	`AmpGrad2 <double> [T/m]`
	`AmpGrad3 <double> [T/m]`
	`AmpGrad4 <double> [T/m]`
	desc.: gradient amplitude for gradient coils (ch1-4)
	`phaseGrad1 <double> [DEG]`
	`phaseGrad2 <double> [DEG]`
	`phaseGrad3 <double> [DEG]`
	`phaseGrad4 <double> [DEG]`
	desc.: phases for gradient coils (f2)
	`phaseGradEnvelope1 <double> [DEG]`
	`phaseGradEnvelope2 <double> [DEG]`
	`phaseGradEnvelope3 <double> [DEG]`
	`phaseGradEnvelope4 <double> [DEG]`
	desc.: phases for gradient coils (f1) - envelope
	`trampupgrad <double> [s]`
	`trampupholdgrad <double> [s]`
	`trampdowngrad <double> [s]`
	`trampdownholdgrad <double> [s]`
	desc.: set ramp-up/down settings for gradients (ch1-4)
	`trampuptx <double> [s]`
	`trampupholdtx <double> [s]`
	`trampdowntx <double> [s]`
	`trampdownholdtx <double> [s]`
	desc.: set ramp-up/down settings for RF (ch5)
	`invertTX (0/1)`
	`invertTX_phase <double> [DEG]`
	`invertTX_factor <double>`
	desc.: settings for active rx-signal supression

	kf1 <int> kf2 <int> kf3 <int> desc.: number of periods during acquisition for rotational frequency (f1), travel frequency (f2) and swipe frequency (f3) sequencetype <int> (d: 0) desc.: TWMPI sequence type phaseCH14 <double> [DEG] desc.: additional phase correction for ch1-4 phaseCH5 <double> [DEG] desc.: additional phase correction for ch5 RX_Uin_max <double> [V] (*) desc.: set ADC sensitivity SR <int> [samples] (d: 2.5e6) desc.: sample frequency avg <int> (d: 1) desc.: set averages tRep <double> [s] (*) desc.: repetition time in case of multiple segments tAQ <double> [s] desc.: acquisition time nPhaseTX <int> (*) desc.: number of measurements per experiment plotdata (0/1) (d: 1) desc.: enables data plot from server plotseq (0/1) (d: 1) desc.: enables initialization before each sequence start data_dobackup <int> (0/1) (d: 1) desc.: activates automatic data backups from server data_MMF_dt <double> [s] (d: 0.5) desc.: set the waiting time between MMF data transfer rcc_Up <double> [V] rcc_fLow <int> [Hz] rcc_fHigh <int> [Hz] rcc_df <int> [Hz] rcc_pickup_radius <double> [m] rcc_pickup_resistance <double> [Ohm] rcc_pickup_n <int> rcc_pickup_Ip <double> [A] rcc_pickup_RXgain <double> [V] (d: 0.2 V) rcc_output_dir <string> desc.: set the pick-up coil parameters temperatureCoilGrad1 <double> [°C] temperatureCoilGrad2 <double> [°C] temperatureCoilGrad3 <double> [°C] temperatureCoilGrad4 <double> [°C] temperatureCoilCH5 <double> [°C] desc.: heating model parameters temperature_set <double> [°C] (d: 30°C) temperature_max <double> [°C] (d: 60°C) temperature_check (0/1) (d: 0) temperature_checkdelay <double> [s] (d: 0) temperature_control (0/1) (d: 0) desc.: temperature control parameters SequenceStartTime (YYYYMMDD-hhmmssttt) desc.: schedules the start time for the next start_measurement

	`TriggerUse (0)`
	desc.: activates the digital outputs
	
	COMPASS related parameters:
	
	`dutycycle (0..1) <double> (d: 0.25)`
	desc.: duty cycle of COMPASS sequences
	`ACfrequencySteps <int> (d: 1)`
	desc.: number of frequencies per experiment
	`ACfrequencySpacing <int> (d: 0)`
	desc.: linear (=0) or logarithmic (=1) frequency spacing
	`ACfrequencyMin <int> [Hz] (d: 1000)`
	`ACfrequencyMax <int> [Hz] (d: 10000)`
	desc.: min/max frequency (if ACfrequencySteps=1 then ACfrequency=ACfrequencyMax)
	`AQnPeriodsOffset <int> (d: 5)`
	desc.: number of periods before acquisition
	`AQnPeriods <int> (d: 5)`
	desc.: number of periods during acquisition (averaging)
	`AQnPeriodsPostset <int> (d: 1)`
	desc.: number of periods after acquisition
	`Scansym <int> (0=no-sym; 1=DC; 2=AC; 3=DC+AC)`
	desc.: data acquisition symmetry setting
	`nStepsAC <int> (d: 5)`
	`nStepsDC <int> (d: 5)`
	desc.: number of AC/DC steps
	`DCmin <double> [T] (d: 0.000)`
	`DCmax <double> [T] (d: 0.005)`
	desc.: min & max DC value
	`ACmin <double> [T] (d: 0.000)`
	`ACmax <double> [T] (d: 0.005)`
	desc.: min & max AC value
	`ACsweep <int> (0=mode1; 1=mode2) (d: 0)`
	desc.: set sequence mode
	`previewCOMPASSdata <int> (d: 0)`
	desc.: enables COMPASS data preview from server
	`previewHarmonics <int> (d: 6)`
	desc.: number of harmonics for preview from server
	`previewInterpolation <int> (d: 2)`
	desc.: set the interpolation factor for preview from server
	`saveCOMPASSdata <int> (d: 1)`
	desc.: save COMPASS data from server
	`saveInterpolation <int> (d: 0)`
	desc.: interpolation factor for saved COMPASS data
	`saveHarmonics <int> (d: 11)`
	desc.: number of harmonics for saved COMPASS data
	`OversamplingFactor <double> (d: 4)`
	desc.: oversampling factor of highest harmonic to be saved
	`nSampleMax <int> (d: 4096)`
	desc.: maximum number of samples per AQ window
	`ACamplitudeSpacing <int> (d: 0)`
	desc.: set spacing for AC amplitudes (0=linspace; 1=logspace)
	`DCamplitudeSpacing <int> (d: 0)`
	desc.: set spacing for DC amplitudes (0=linspace; 1=logspace)

	Sample-Heater specific commands SH_set_target_temperature desc.: SH_get_target_temperature desc.: SH_check_temperature desc.: SH_is_airflow_on desc.: SH_get_air_pressure_value desc.: SH_get_temperature_port1 desc.: SH_get_temperature_port2 desc.: **special parameters** inductance <double> [H] (COMPASS calibration) desc.: calibration parameter (temporary) rx_norm_max <double> [%] (READONLY) desc.: normalized rx-signal for set ADC range
get_params	Get the current value of the parameter (s.a.) get_params <parameter> →server reponse !get_params <parameter> <value> parameters: see set_params
get_temperature	Gets the current temperature value of sensor 1
get_temperature2	Gets the last temperature value from internal memory
get_mpi_sn	Gets the serial number of the MPI device
get_console_sn	Gets the serisl number of the drive console
reset_all_params	Reset all parameters of the scanner to the default settings.
start_measurement	Start a new measurement (defined by set_params method) The scanner is busy during measuring ➔ cmd is_measuring provides the answer 1
prepare_pulse_program	Pre-calculates the new sequence.
update_pulse_program	Renews the pulse program to calculate the frequencies without starting a measurement
get_ac_frequencies_1d	After changing sequence specific parameter, the frequency table has to be renewed. These parameters are available after start_measurement or update_pulse_program. <#f> {f_n} (READONLY)
get_ac_amplitudes_1d	After changing sequence specific parameter, the AC amplitude table has to be renewed. These parameters are available after start_measurement or update_pulse_program. <#AC> {AC_n} (READONLY)
get_dc_amplitudes_1d	After changing sequence specific parameter, the DC amplitude table has to be renewed. These parameters are available after start_measurement or update_pulse_program. <#DC> {DC_n} (READONLY)
get_harmonics_1d	Get the number of generated higher harmonics for each frequency taking the Nyquist theorem into account. <#f> {#h_n}

`abort_measurement`	Abort a running measurement as fast as possible.
`get_measurement_duration`	Get the length of the sequence. Requires `prepare_pulse_program` `<time>` (READONLY)
`is_measuring`	Check for measurement is running.
`create_mmf`	Creates the memory for MMF data transfer
`write_mmf`	Writes a new data set to the MMF interface (server → client)
`write_rcc`	Writes the RCC files after RCC measurement (`set_params method rcc_cal1/2`)

The sending of a command **<cmd>** is confirmed by the server with **<!cmd>**.
An unknown command is confirmed by the server with **<?cmd>**.
Depending on the command, additional information is provided separated by a **blank**.

7. Q&A

Software-related issues

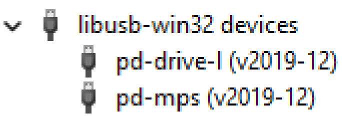
1. **Software does not start!**
   Solution: The executable (Octoview.exe) does not lie in its intended location. Make sure to keep the structure of the provided zip-archive (`\..\Octoview\Win64\Release\Octoview.exe`).

2. **Issues in 3D visualization.**
   Solution: Make sure you have installed the latest drivers for the PCs' graphic card.

3. **TCPIP-communication issues**
   The client does not work anymore. No response on given commands to the server.
   All commands sent to the server should get an answer with a preceding exclamation mark, e.g.,
   Client sent: `start_measurement`
   Server sent: `!start_measurement`
   Solution: Use the 'reset' button on the 'pipeline' tab in the 'cmdlist overview' area to renew the cmdlist. In addition, increase the polling interval of the timer.

Hardware-related issues

1. **The scanner does not start correctly: the initializing status did not vanish.**
   Possible issue: the server cannot connect to the scanner because of erroneous drivers.
   Solution: Please check the driver status: go to the device manager (Windows OS) and check if two USB devices are visible:
   - pd-drive-l shows the connection to the drive console:
   - pd-mps shows the connection to the scanner

   Please check the log-files of the server for more information (can be found in the installation path):
   `\..\Pure Devices\PD OctoView\Win64\Release\MPI_server.log`

   Possible issue: the drive-console cannot be started.
   The log-file shows the error:
   Warning: Error using winqueryreg
   Specified key is invalid.
   Please check the registry entries:
   Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Pure Devices\Settings
   Solution: please reinstall the framework to repair the registry.

2. **There is no signal on the receive chain although I put a strong sample into the scanner.**
   Possible issue: receive chain is not connected.
   Solution: Please control the cabling of the scanner.

   Possible issue: receive coil is broken.

Check: switch off all electronic parts. Disconnect all cables from the MPI unit. Use an Amp-meter to measure the resistance of the receive coil directly in the XLR connector (pins 1→2).
If no resistance can be measured, the receive coil is broken. Please contact the support.

Possible issue: CH5 does not provide sufficient high magnetic fields.
Solution 1: control the cabling of the amplifiers. Make sure all electronic components are switched on.
Solution 2: control the strength of the transmitting signals to the amplifiers.
Solution 3: remove the cables and check for interruptions.

3. **The measurement process does not start.**
   Possible issue: amplifiers are switched off.
   Solution: switch on amplifiers.

   Possible issue: control cable between drive, splitter-box and amplifiers is interrupted.
   Solution: check the connections. Remove the cable and check against interruptions.

   Possible issue: broken amplifier.
   Solution: Check the status LEDs during experiment. Call the support.

ToDo

-add description for „eventbutton“

-update images for GUI: new settings-tab