Instruction manual for:

PATLab:

Photoacoustic Tomography Laboratory

Application developed in MATLAB app designer

Parsa Omidi,
Lawrence C.M. Yip,
Elina Rascevska,
Mamadou Diop,
Jeffrey J.L. Carson

Ver 1.0
Last revised: July 6, 2022

Table of Contents

1. Overview

This software provides 2D and 3D simulation and image reconstruction utilities for photoacoustic tomography (PAT). It consists of four main panels (Figure *1-1*):

1- *Loading/Generating Raw Data*,
2- *Pre-Processing*,
3- *Image Reconstruction*, and
4- *Plotting & Saving.*

This instruction manual walks the user through installing and using the software.

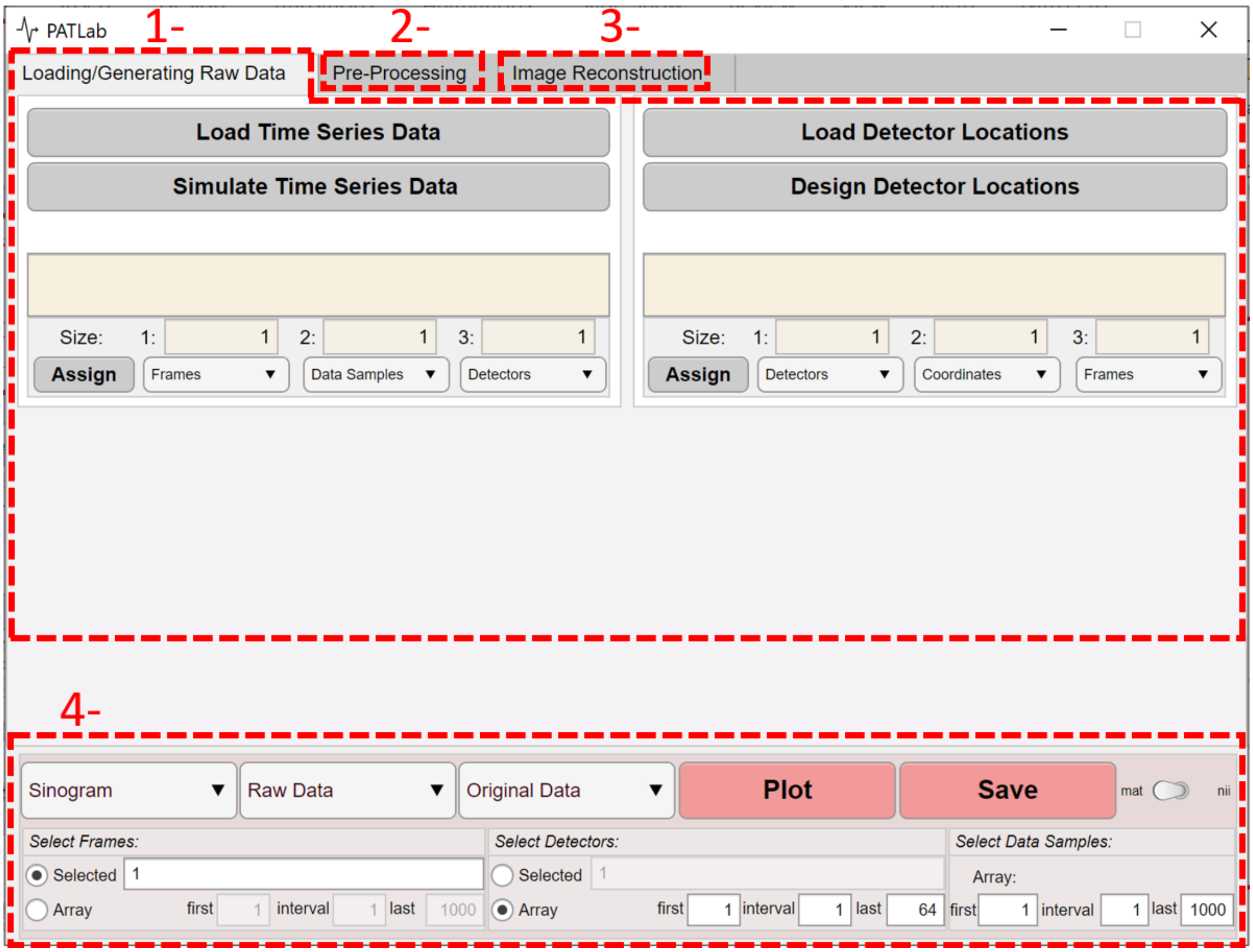


Figure 1-1: PATLab overview.

2. Running the software

2.1. Prerequisites

- MATLAB 2021a or newer

This software is written in MATLAB 2021a. Application functions may perform incorrectly if used with earlier versions of MATLAB.

2.2. Installation and running the software

There are three ways to run the software:
Option 1: Add the software as a MATLAB application.

1- Double click on *PatLab_2.mlappinstall* file in the software package,
2- The app will be installed as a MATLAB app and can be found under the tab APPS. Click on it to run the app.

⚠ After installation, the app will remain in “MATLAB apps” unless uninstalled (right click -> Uninstall).

Option 2: Install the software as a standalone desktop app on your operating system.

⚠ To run the application, users must have MATLAB Runtime installed on their systems. For more information, see https://www.mathworks.com/products/compiler/matlab-runtime.html.

Option 3: Run the software in *AppDesigner.*

1- Open the *AppDesigner* by typing *appdesigner* in MATLAB workspace,
2- Open the app *PAT_Reconstruction_07.mlapp* in the Designer tab.

✓ This method permits modifying the code.

3. Step-by-step instructions

3.1 Loading/Generating Raw Data

This part of the software includes functions concerning:

1- Detector location(s) (indicated with green dashed box in Figure *3-1*):
   - ***Load Detector Locations*** (data that previously generated or experimentally acquired), or
   - ***Design Detector Locations*** (using an additional tool to generate an array of detectors).

2- PA time series data (indicated with blue dashed box in Figure *3-1*):
   - ***Load Time Series Data*** (data that previously generated or experimentally acquired), or
   - ***Simulate Time Series Data*** (using the k-Wave toolbox).

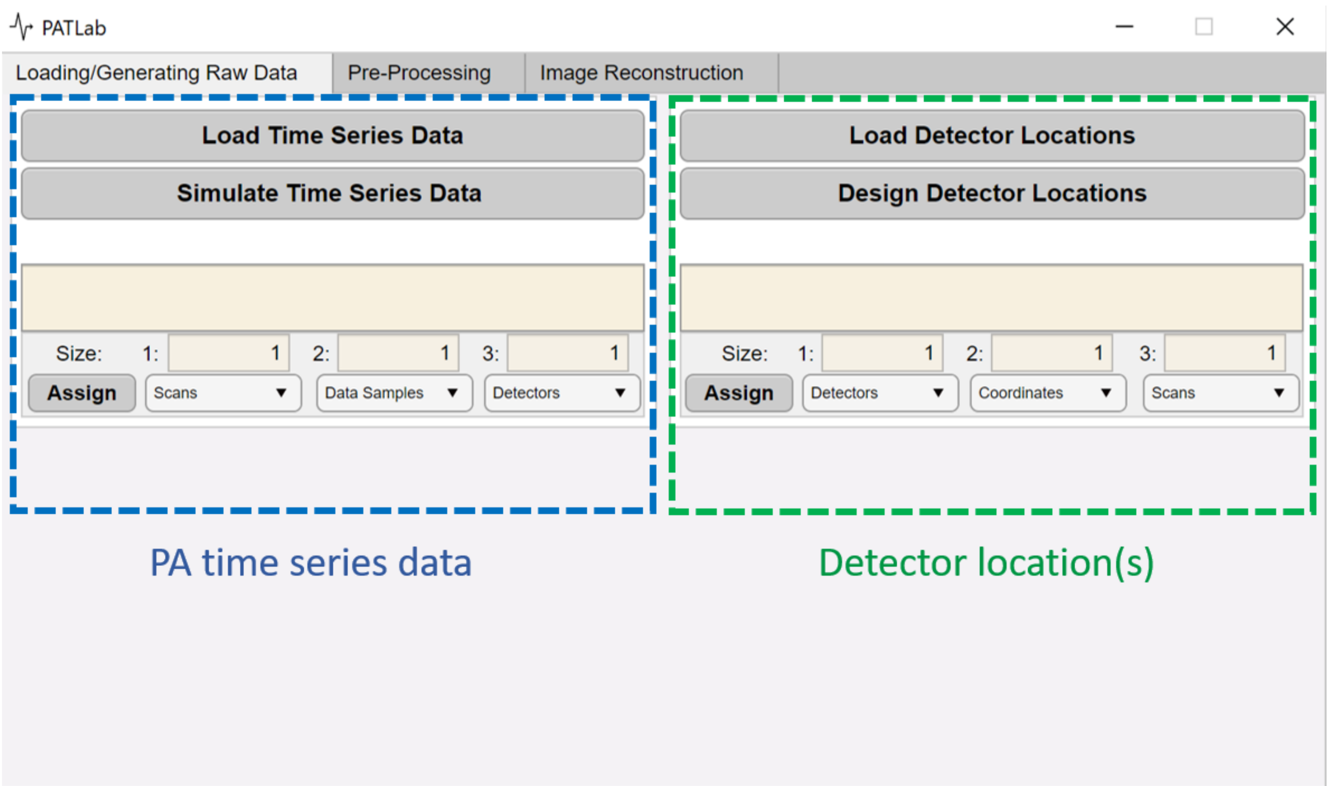


Figure 3-1: Loading/Generating Raw Data panel.

3.1.1 Load Detector Locations

The button ***Load Detector Locations*** (Figure *3-1*) is for loading previously defined detector positions. The accepted data format is ***.mat***. The detector locations can be formatted as 2D or 3D matrices (one dimension for the detectors, one for the number of dimension coordinates (i.e., two for 2D and three for 3D), and the third for the number of frames). The matrix can be 2D when there is only one frame or one detector element available. The order of the dimensions is not important, PATLab allows assigning each matrix dimension to a role. To do so, three ***Size*** boxes are designed for each dimension allowing the user to select their corresponding roles by utilizing the dropdown menus provided. ***Assign*** must be pressed afterwards to confirm the selection. It will turn green if successful.

⚠ Do not forget to press the ***Assign*** button; otherwise, the software will not pass the data to the next step.

3.1.2 Design Detector Locations

The ***Design Detector Locations*** button (Figure *3-1*) is useful when a file with detector element positions is not available, or to simulate a new detector array (configuration). The button opens a new interface offering common 2D and 3D configurations (Figure *3-2*). Furthermore, it is possible to design a combination of different configurations.

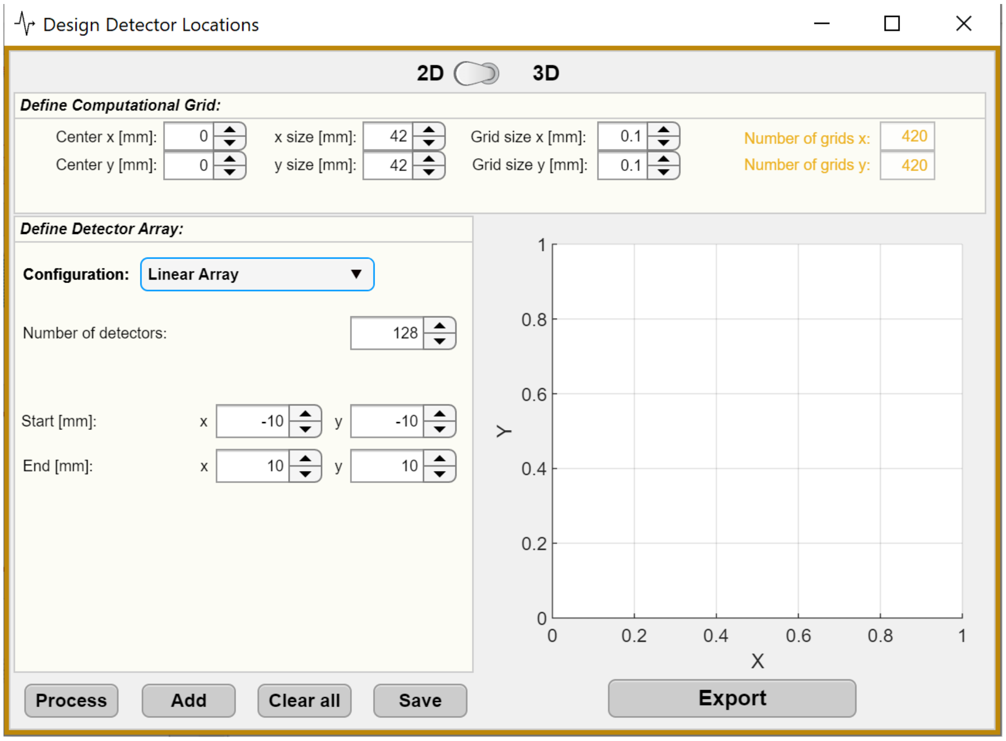
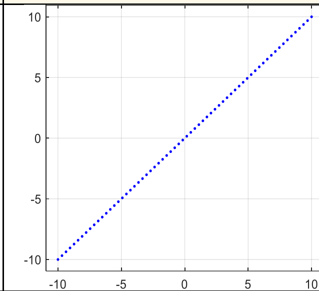
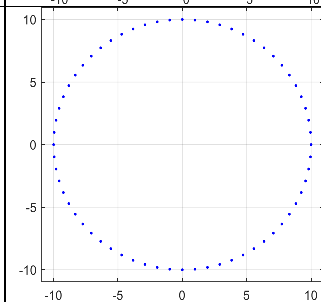
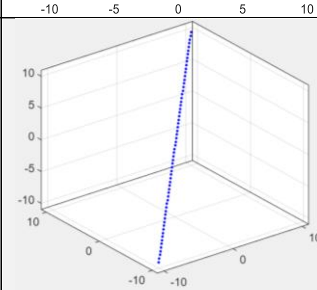
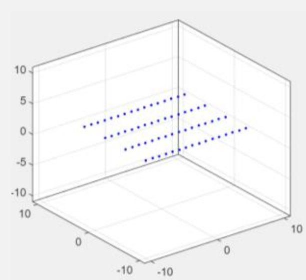


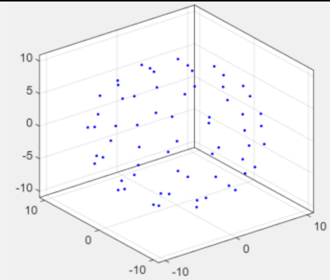
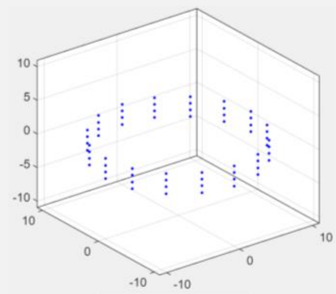
Figure 3-2: Design Detector Locations interface.

To create a set of detector locations:

1) Select the dimensionality of the system (2D or 3D) by using the switch button located at the top center of the interface.
2) Define the computational grid (the space that covers all the detector elements and the field of view (FoV)). In this panel, users may define the area with the following variables:
   - ***Center x*** & ***y [mm]*** (and ***Center z [mm]*** for 3D): indicate the coordinates for the center of the computational area.
   - ***x*** & ***y size [mm]*** (and ***z size [mm]*** for 3D cases): indicate each dimension's size.
   - ***Grid size x*** & ***y [mm]*** (and ***Grid size z [mm]***): indicate the pixel/voxel size in each dimension.
3) Define the detector array details by selecting a ***Configuration*** (a detector arrangement in the defined computational area), number of detectors, and multiple other features for detector positioning based on the selected configuration (Table *3-1*).

Table 3-1: Available detector configurations.

Dimensionality:	Configuration:	Features to define:	Example image of the configuration: 64 detector elements
2D	***Linear Array***	***Number of detectors***: the number of transducers used in the array ***Start [mm] xy:*** the xy coordinates of the start point of the array ***End [mm] xy:*** the xy coordinates of the end point of the array	
	Circular	***Number of detectors***: the number of transducers used in the array ***Radius [mm]:*** the radius of the circle ***Arc angle [deg]:*** the size of the arc (can be a value between 0 and 360) ***Center [mm] xy:*** the xy coordinate of the circle center	
3D	***Linear Array***	***Number of detectors***: the number of transducers used in the array ***Start [mm] xyz:*** the xyz coordinates of the start point of the array ***End [mm] xyz:*** the xyz coordinates of the end point of the array	
	Planar	***Number of detectors in a single linear array:*** the number of transducers used in an array (the plane is assumed to be composed of multiple horizontal linear arrays) ***Space between detectors in the linear array [mm]:*** the detectors pitch in the linear array ***Direction of the array:*** the direction of the linear array which can be along x or y ***Number of repetitions in other direction (x or y):*** the number of horizontal linear arrays ***Spaces between the arrays [mm]:*** pitch between the linear arrays ***First element of the first array [mm] xyz:*** the xyz coordinates of the first point (which has the lowest value for x and y) ***Total number of detectors:*** This field is not editable and only displays the number of transducers used by the plane.	

	Spherical	***Number of detectors***: the number of transducers used in the array ***Radius [mm]:*** the radius of the circle ***Center [mm] xyz:*** the xyz coordinate of the circle center	
	Cylindrical	***Number of detectors in a single ring (xy):*** the number of detectors used in a ring (the cylinder is assumed to be composed of multiple rings on top of each other) ***Number of rings:*** the number of rings that are stacked on top of each other ***Space between the rings in z [mm]:*** the pitch between the rings ***Radius [mm]:*** radius of the rings ***Center of the bottom ring [mm] xyz:*** the xyz coordinates for the center of the bottom ring in the stack ***Total number of detectors:*** This field is not editable and only displays the number of detectors used by the plane.	

There are five buttons at the bottom of the app:

- ***Process***: to create the designed array,
- ***Add***: to add the newly processed array to to previously designed array(s),
- ***Clear all***: to clear all the detectors added so far,
- ***Save***: to save all the detectors added so far as .mat file in the current MATLAB directory, and
- ***Export***: to send the designed detector array into the main interface.

⚠ Do not forget to click ***Add*** after designing a configuration; otherwise, the new configuration will overwrite the previous one.

⚠ By pressing the ***Add*** button, the color of the button will turn to green as an indicator to know that the software is ready for the next configuration.

⚠ Pressing the ***Export*** button will close the current window.

This interface allows designing a combination of detector element arrays, such as the example indicated in the Figure *3-3*.

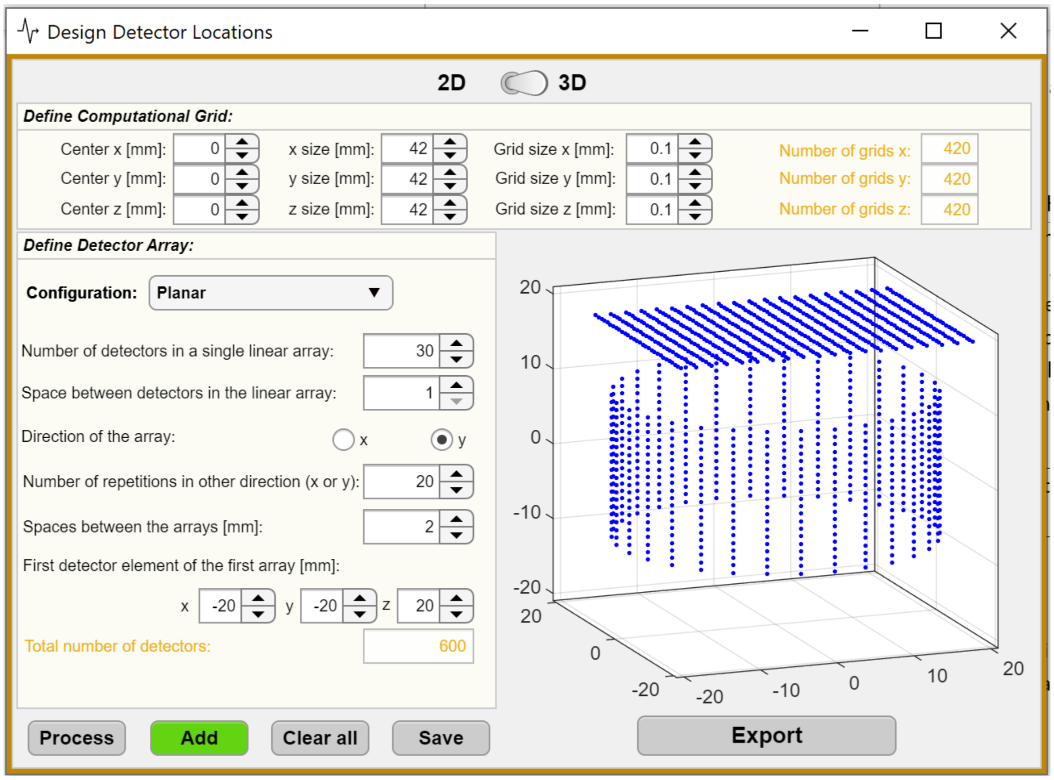


Figure 3-3: A detector array contains 1200 elements as a result of combining a planar configuration and a cylindrical configuration, each contains 600 elements.

3.1.3 Load Time Series Data

Photoacoustic time series data can be directly loaded by clicking the ***Load Time Series Data*** button (Figure *3-1*). In response, a file selection dialog box will open, which allows the user to select data formatted as a **mat** file. The **mat** file can be formatted as a 3D matrix, where the dimensions are as follows: (i) number of detectors, (ii) number of data points per time series, and (iii) number of frames. The matrix can be 2D when only one frame or one detector is available. The order of the dimensions is not important since PATLab allows assignment of each matrix dimension to a role. Three ***Size*** boxes display the dimensions and dropdown menus allow the user to select the role of each dimension. The ***Assign*** button must be pressed afterwards to confirm the selection. It will turn green if successful.

⚠ Do not forget to press the ***Assign*** button; otherwise, the software can not pass the data to the next step.

3.1.4 Simulate Time Series Data

Simulate Time Series Data provides a user-friendly interface (Figure *3-4*) that allows simulation of PA time series data for a previously configured detector configuration and absorber. This interface utilizes k-Wave's computational functions, which are described in greater detail in the k-Wave manual (http://www.k-wave.org/manual/k-wave_user_manual_1.1.pdf).

⚠ Prior to simulating time series information, detector locations must be loaded or generated. Refer to sections 3.1.1 or 3.1.2.

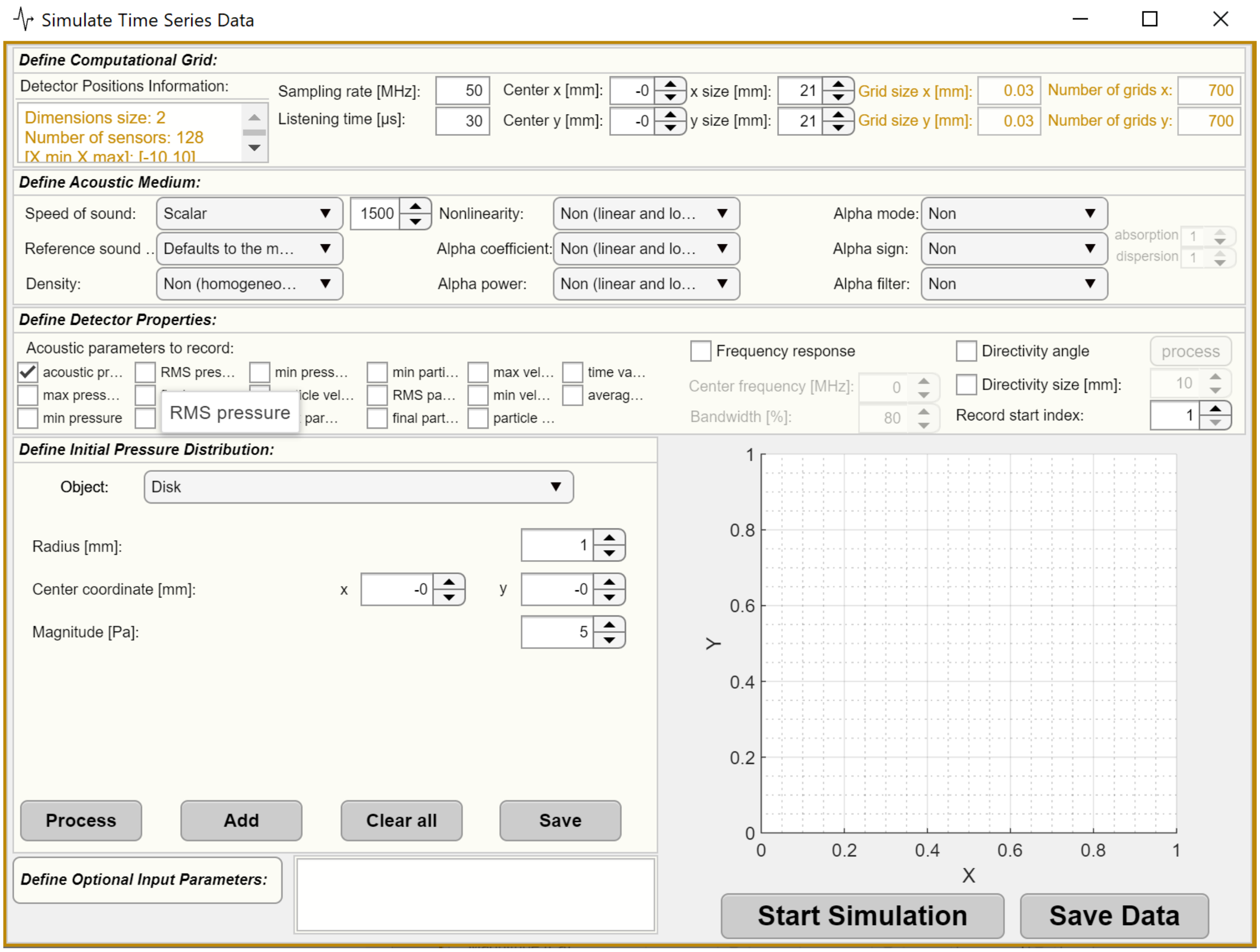


Figure 3-4: Graphical interface for PA *time series* simulation.

The PA time series simulation interface composed of five separated panels:

1- ***Define Computational Grid***: to indicate an area that should contain all loaded detectors and any potential absorbers.

- ***Sampling rate [MHz]:*** sampling frequency that satisfies the Nyquist rate of two grid points per wavelength.
- ***Listening time [µs]:*** length of time series.
- ***Center xyz [mm]:*** xyz coordinates for the center of the computational grid.
- ***xyz size [mm]***: xyz size of the computational grid.

✓ Values for ***Grid size x, y,*** & ***z [mm]*** are automatically filled/calculated based on the ***Sampling rate [MHz]*** and ***Speed of sound [m/s]***. ***Speed of sound [m/s]*** is the scalar value specified in ***Define Acoustic Medium***.
Grid size = Speed of sound / Sampling rate.

⚠ The variables in this panel will be automatically initiated based on the loaded detector configuration; however, they must be adjusted to include the absorber(s) that will be added in the ***Define Initial Pressure Distribution*** panel.

2- ***Define Acoustic Medium***: The medium can be homogeneous, with the speed of sound the same for the entire grid, or heterogeneous, in which the user can specify a speed of sound map and density distribution within the grid.

- ***Speed of sound [m/s]:*** offers a dropdown menu that allows user to select:
    - ***Scalar value –*** selection of this option creates a field where user can enter the desired speed of sound.
    - ***Load a matrix –*** opens a directory to select desired data file with speed of sound data. Dimensions of this file must match the size of the defined computational grid.
    - ***Design a matrix–*** opens a graphical interface for creating the speed of sound maps (Figure *3-5*). The dimension of the matrix will match the dimensionality of the computational grid (i.e. 2D vs 3D). The interface allows the user to divide the entire computational grid (defined previously) into several regions. The data from each region will appear in a table that allows change the speed of sound value manually. This interface consists of three steps:
        - ***Step 1*** is for specifying the number of regions in all dimensions,
        - ***Step 2*** displays the matrix in a form of a table and allows manually to alter the speed of sound values for every region.
          In 3D scenarios, the table must be assigned to different z slices by using the ***Assign*** button.
        - ***Step 3*** is containing 5 funtions: ***Process*** for interpolating the regions to match the size of the computational grid, ***Method of Interpolation*** for choosing which algorithm is used for this process, ***Plot*** for visualizing the generated matrix, ***Save*** for saving the generated matrix, and ***Export*** for transferring the result to the time series simulation interface.

(a) (b)

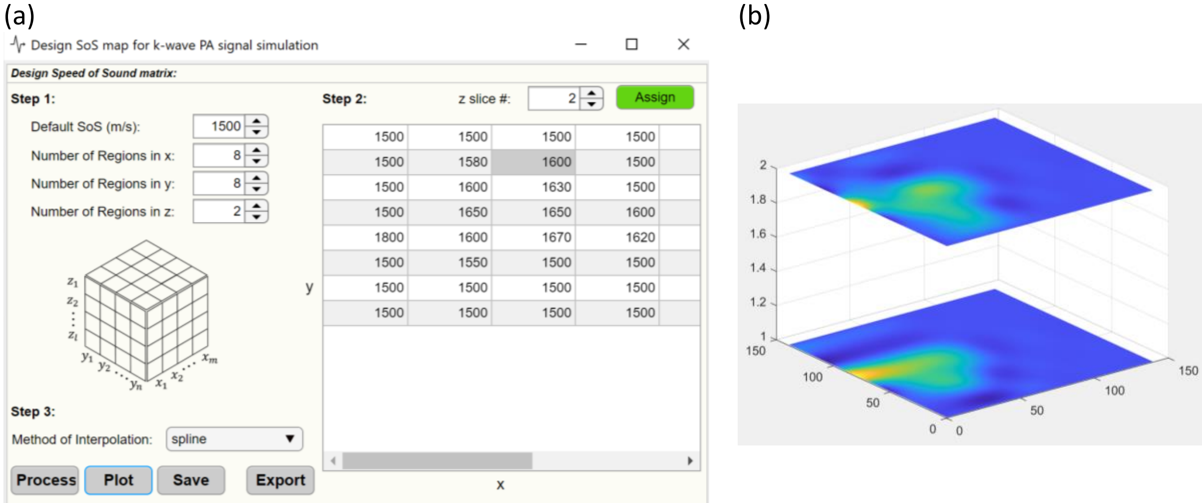


Figure 3-5: a) Design a speed of sound map for a 3D scenario. **b)** Plotting the processed result.

- ***Reference speed of sound used in the k-space operator [m/s]:*** contains dropdown menu that allows to (i) enter a ***scalar value*** or set it as ***maximum speed of sound from the previously defined medium***.
- ***Density [kg/m3]:*** ambient density distribution within the medium. Can be selected to be (i) ***homogeneous and lossless medium*** throughout the grid (no density set to the medium), (ii) ***scalar value*** specified by the user or (iii) ***load a matrix*** containing density distribution data (dimension must match the computational grid size).
- ***Nonlinearity:*** nonlinearity parameter. Can be set as (i) ***linear and lossless*** (no parameter applied to the medium) or (ii) a ***scalar value*** specified by the user.

- ***Alpha coefficient [dB/(MHzcm)]:*** power law absorption prefactor. Can be set to (i) ***linear and lossless*** (no coefficient will be applied to the medium), (ii) a ***scalar value*** specified by the user or (iii) ***load a matrix*** containing alpha coefficient values (dimension must match the size of the selected computational grid.
- ***Alpha power:*** power law absorption exponent. Can be set to (i) ***linear and lossless*** (no coefficient will be applied to the medium) or (ii) a ***scalar value*** specified by the user.
- ***Alpha mode:*** optional input to force either the absorption or dispersion terms in the equation of state to be excluded; valid inputs are "no absorption" or "no dispersion".
- ***Alpha sign:*** 2-element array used to control the sign of absorption and dispersion terms in the pressure-density relation.
- ***Alpha filter:*** frequency domain filter applied to the absorption and dispersion terms in the pressure-density relation.

3- ***Define Detector Properties***: Contains information regarding the acoustic parameters to be recorded, the frequency response and directionality of the detector(s).

- ***Acoustic parameters to record:*** list of parameters that can be recorded; however***,*** only ***acoustic pressure*** is mandatory and will be transferred back to the main interface. The rest of them, if selected, can be accessed by saving the generated data using the button at the bottom right of the interface (***Save Data***).
- ***Frequency response:*** define the ***center frequency [MHz]*** and percentage ***bandwidth [%***] in the frequency domain.
- ***Directivity angle:*** matrix of directivity angles (direction of maximum response) for each detector element. This feature is only supported for 2D simulations.
  When the directivity angle checkbox is selected, a process button appears next to it. This button opens an interface where you can specify the directionality in five different ways (Figure 3-6(a)):
  - ***Global focal point***: consider a point as the global focal point for all detectors.
  - ***Global unit vector***: consider that all detectors have the same directivity. The directivity can be represented as a global unit vector. This method is recommended for linear and planar configurations where all detectors face in the same direction.
  - ***Load focal point***: a 3D matrix can be loaded containing the focal point information about each detector according to any frame position. This matrix must contain the same number of detectors, frames, and coordinates (each dimension represents one of them) as the used detector locations in section 3.1.1 or section 3.1.2. A pop-up interface is then provided for assigning each dimension to the respective roles by selecting corresponding options from the dropdown menus (Figure 3-6(b)).
  - ***Load unit vector***: a 3D matrix can be loaded containing the unit vector information about each detector according to any frame position. This matrix must contain the same number of detectors, frames, and coordinates (each dimension represents one of them) as the used detector locations in section 3.1.1 or section 3.1.2. A pop-up interface is then provided for assigning each dimension to the respective roles by selecting corresponding options from the dropdown menus (Figure 3-6(b)).
  - ***Load direction angles***: in this option, the angles must specified in radians where 0 corresponds to maximum sensitivity in the y direction and pi/2 or -pi/2 to maximum sensitivity in the x direction (in K-wave and PATLab, x and y are reversed).
- ***Directivity size [mm]:*** equivalent detection element size (the larger the element size, the more directional the response).
  Record start index: time point at which the detector should start recording (default = 1).

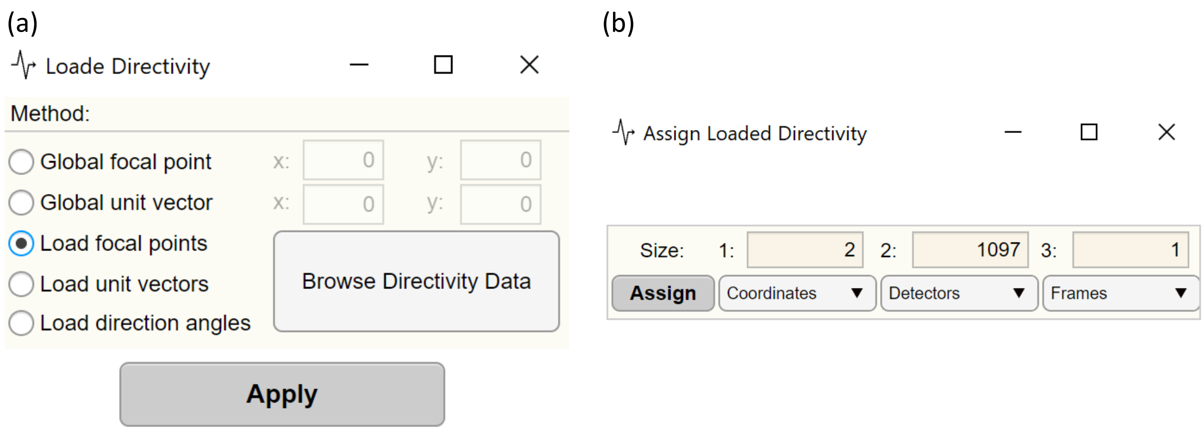
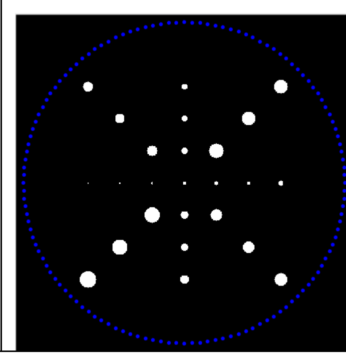


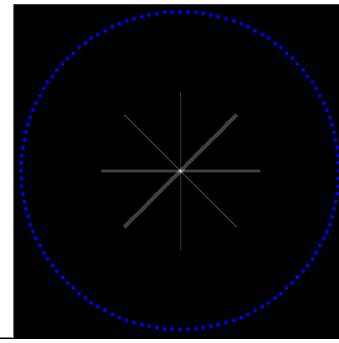
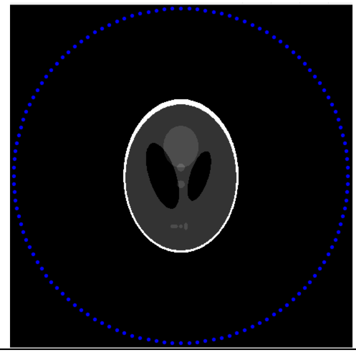
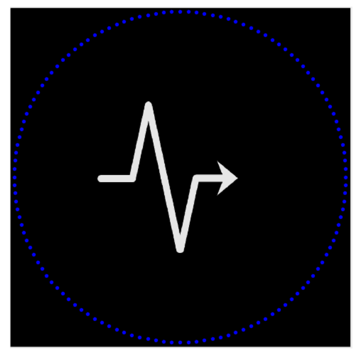
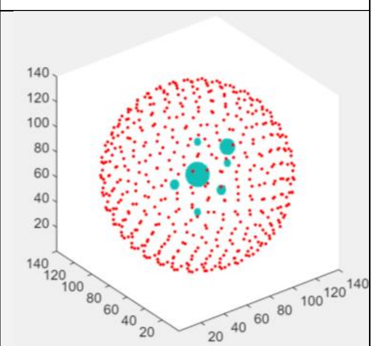
Figure 3-6: **a)** An interface with five different options to specify the directivity angles. **b)** An interface that appears when the ***Browse Directivity Data*** button in (a) is pressed. This interface is to assign each dimension of the loaded directivity unit vectors or focal points to the respective.

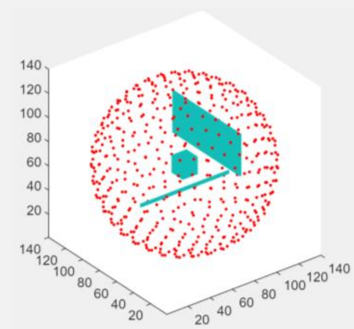
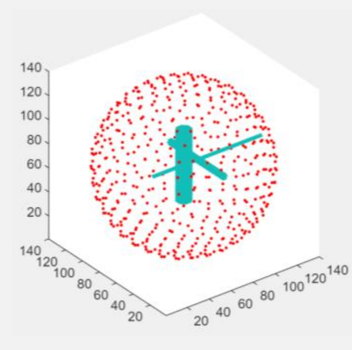
4- ***Define Initial Pressure Distribution***: this panel provides tools for designing absorbers in 2D or 3D scenarios. Multiple shapes are provided in the ***Object*** dropdown. Every shape has several variable to regulate. These variables along with examples are listed in the.

⚠ The ***Object*** list changes based on the dimensionality (2D or 3D).

Table 3-2: Provided options to design absorbers in 2D or 3D scenarios.

Dimensionality:	**Object name:**	**Features to define:**	**Example image for:** **2D: circular detector configuration (in blue)** **3D: spherical detector configuration (in red)**
	Disc	***Radius [mm]***: radius of the disc ***Center coordinate x & y [mm]***: the center of the disc, ***Magnitude [Pa]***: the initial pressure magnitude allocated to the created disc.	

2D	***Rectangle***	***Length [mm]***: the length of the rectangle, ***Width [mm]***: the width of the rectangle, ***Center coordinate x & y [mm]***: the center of the disc ***Angle [deg]***: the angle of the rectangle towards the Y axes (vertical), ***Magnitude [Pa]***: the initial pressure magnitude allocated to the created disc	
	Modified Shepp-Logan phantom	Use a 2D head phantom available in MATLAB. ***Center coordinate x, y & z [mm]***: the center of the image, ***Width [mm]***: the width of the phantom, ***Magnitude [Pa]***: the initial pressure magnitude allocated to the image.	
	Load Image	Allows user to select an image by opening a selection dialog, ***Load Image***: a button to browse and select a binary image (black and white) or a .mat file. ***Center coordinate x, y & z [mm]***: the center of the image, ***Width [mm]***: the width of the image (with proportional changes in length), ***Magnitude [Pa]***: the initial pressure magnitude allocated to the image.	
	Ball	***Radius [mm]***: radius of the ball, ***Center coordinate x, y & z [mm]***: the center of the ball, ***Magnitude [Pa]***: the initial pressure magnitude allocated to the created ball.	

3D	***Cube***	***Dimensions x, y & z [mm]:*** the size of each dimension, ***Center x, y & z [mm]:*** coordinates for the center position, ***Magnitude [Pa]:*** the initial pressure magnitude allocated to the created cube.	
	Bar	***Length [mm]:*** the length of the bar, ***Direction:*** the direction of the linear array which can be along x, y, or z, ***Radius [mm]:*** radius of the bar, ***Center x, y & z [mm]:*** coordinates for the center position, ***Magnitude [Pa]:*** the initial pressure magnitude allocated to the created bar	

⚠ The examples associated with each object in the above table contains multiple of the same object with different defined features, except ***Load Image*** and ***Modified Shepp-Logan phantom*** which their examples contain only one object.

✓ It possible to have a combination of different kind of objects in the designed field of view.

At the bottom of the panel, four keys are located as:

- ***Process***: to start processing the selected object with corresponding variables,
- ***Add***: to add a processed object to the current design,
- ***Clear all***: to clear all the objects added so far,
- ***Save***: to save all the objects added so far as a .mat file in the MATLAB directory. The saved file can be used as your gold standard image to evaluate your reconstruction algorithm in the main interface.

⚠ Do not forget to click ***Add*** as you make changes to the design; otherwise, the new processed object will overwrite the object that you have not added.

✓ By pressing the ***Add*** button, the colour of the button will turn green as an indicator that the software is ready for the next object.

5- ***Define Optional Input Parameters***: a number of other parameters that can control the default behaviour of k-Wave are listed in a pop-up window (Figure *3-7*).

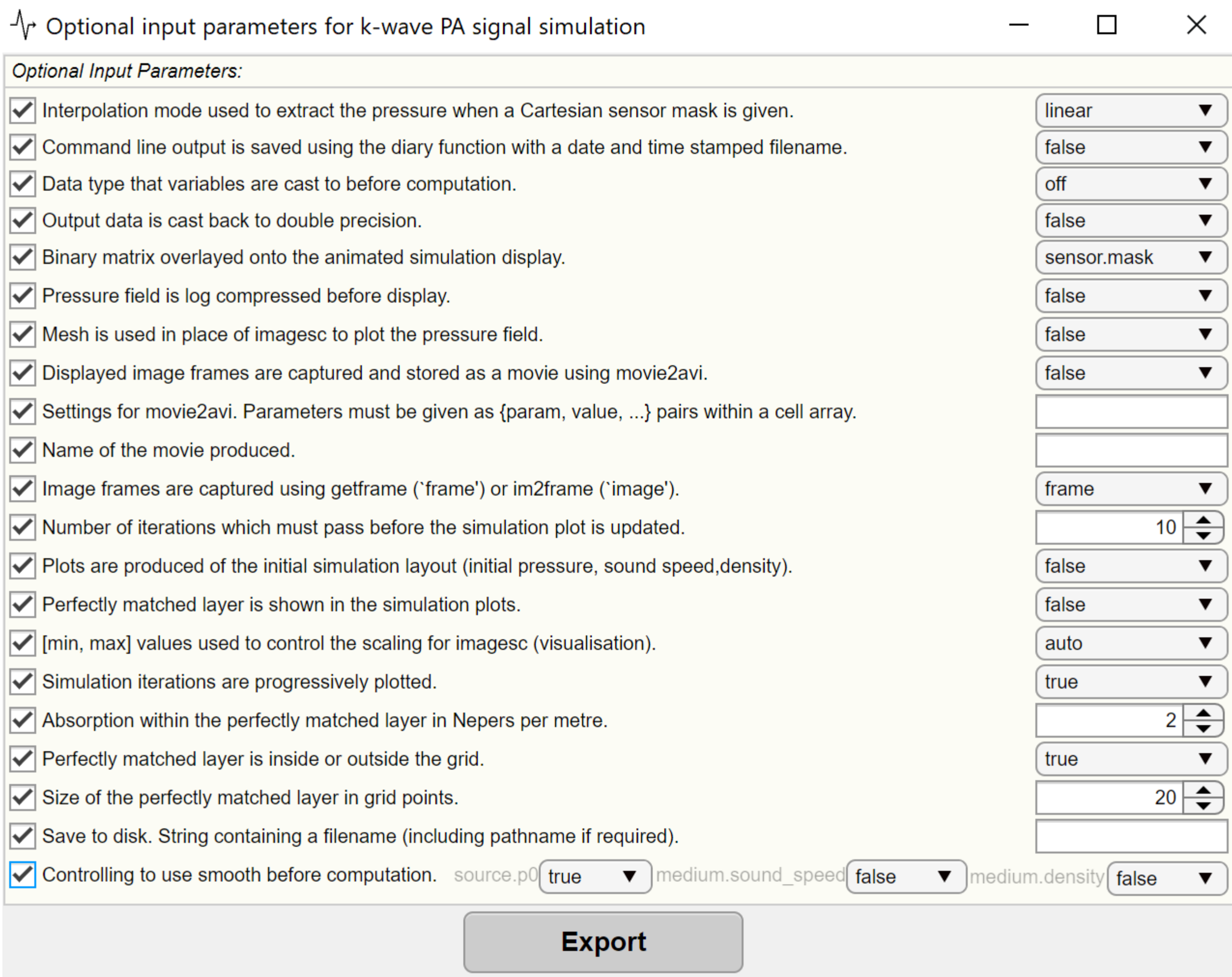


Figure 3-7: Optional input parameters for k-Wave simulation.

When the design is complete, the user must press ***Start Simulation*** to initiate generation of PA signals. After the simulation, the ***acoustic pressure*** (a check box option in the ***Defined Detector Properties*** panel) will be transferred automatically to the main app and the rest of the recorded parameters (all those that have been selected in the ***Defined Detector Properties*** panel) can be saved by ***Save Data*** button.

3.2 Pre-Processing

The ***Pre-Processing*** (Figure *3-8*) panel allows processing the raw time series data prior to image reconstruction. It allows (i) removing undesired frames or excluding data from particular detector(s), (ii) normalizing data using several methods, (iii) applying filters to the data, (iv) down sampling data using multiple methods, and (v) applying a matched-filter to data . With each pre-processing step, a new dataset is created that can be later used for further pre-processing or for the reconstruction.

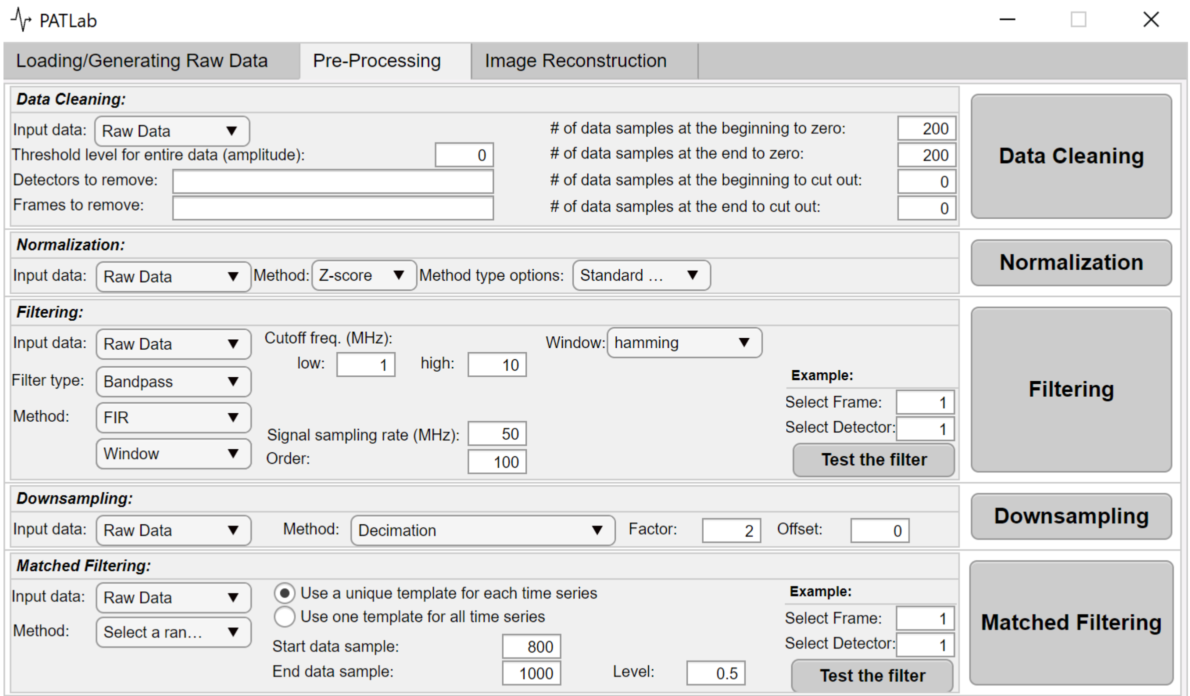


Figure 3-8: Pre-processing panel composed of 5 different task-sections: Data Cleaning, Normalization, Filtering, Downsampling, and Matched Filtering.

3.2.1 Data Cleaning:

Allows removing or thresholding a portion of the selected data. The data to be processed can be selected through the ***Input data*** dropdown menu. This allows processing previously processed data, or the raw data. The options for cleaning data are as follows, with examples in Figure *3-9*.

- ***Threshold level for entire data (amplitude):*** threshold level to apply to the whole data. Zero means no thresholding. Any value above zero will exclude data below the indicated value. Note that thresholds are applied to the absolute values of the data.
- ***Detector to remove:*** indicate the detector element number(s) (comma or space delimited) to remove. The detector element numbering starts at 1 and ends at the number of total detector elements in the data set. Leaving this empty means no detector elements will be removed.
- ***Frames to remove:*** indicate the frame number (comma or space delimited) to remove. The frame numbers start at 1 and end at the number of total frames. Leaving this empty means no frames will be removed.
- ***# of data samples at the beginning to zero:*** the number of data samples (time points) that we want to zero from the beginning of the signal (typically, early time points contain no information and zeroing them may decrease the artifacts in the reconstructed image).

✓ A proper value for this field is related to the distance between the detector elements and the potential absorbers.

- ***# of data samples at the end to zero:*** specifies the number of data samples to zero at the end of the signal for all detector elements.
- ***# of data samples at the beginning to cut out:*** this removes the stated data samples and shifts the entire time series to the left.
- ***# of data samples at the end to cut out:*** this removes the stated data samples from the end of the time series.

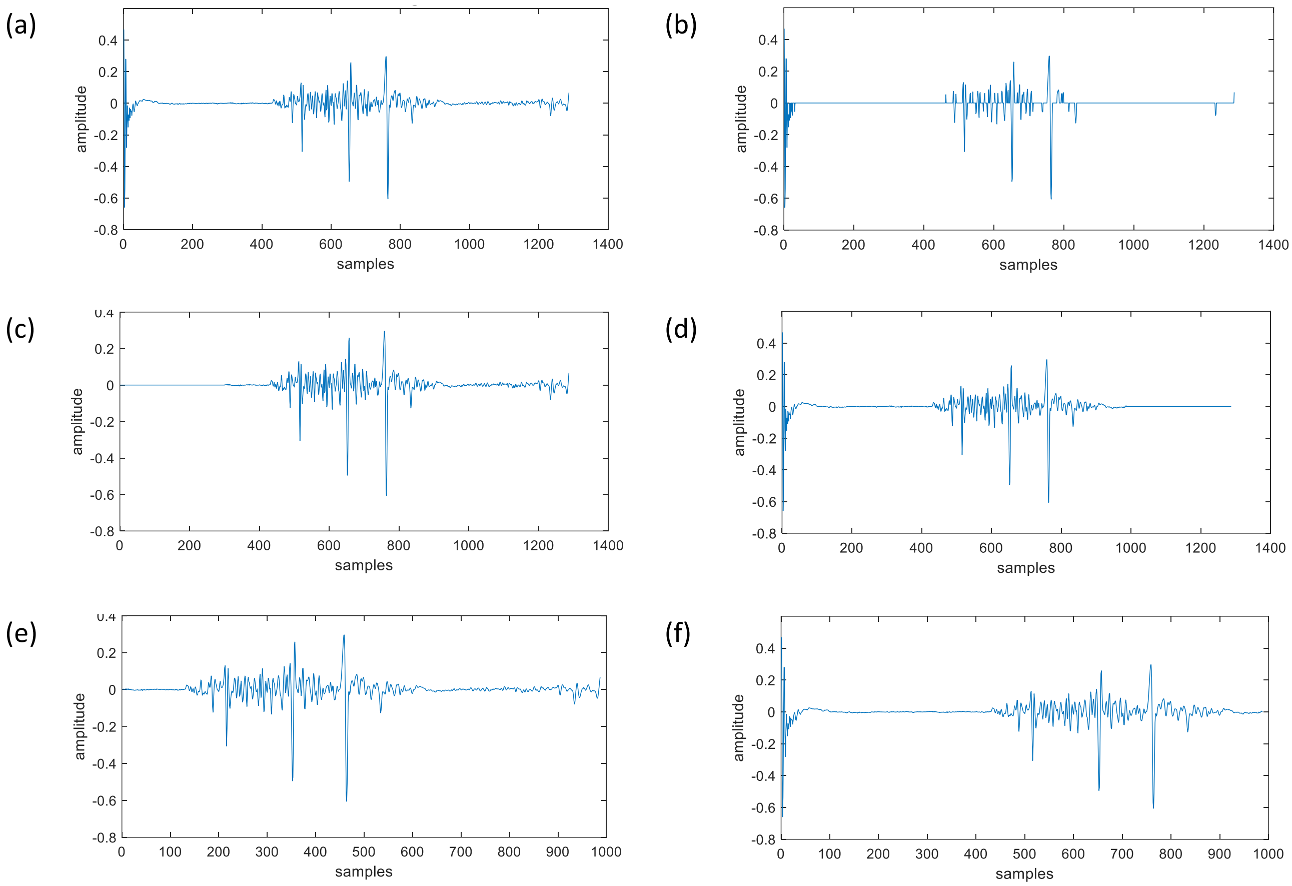


Figure 3-9: Data cleaning example. **a)** Raw data, **b)** Threshold level for entire data (amplitude): 0.05, **c)** # of data samples at the beginning to zero: 300, **d)** # of data samples at the end to zero: 300, **e)** # of data samples at the beginning to cut out: 300, and **f)** # of data samples at the end to cut out: 300.

3.2.2 Normalization:

The ***Normalization*** section allows applying different normalization methods (listed in the dropdown menu under ***Method*** and ***Method type options)*** to the data. It is based on the MATLAB normalize function (more information is available at:
https://www.mathworks.com/help/matlab/ref/double.normalize.html#d123e907220).
In PATLab, the following options are available:

1. ***Z-score***
   - ***Standard deviation***: Center and scale to have mean 0 and standard deviation 1.
   - ***Robust***: Center and scale to have median 0 and median absolute deviation 1.
2. ***Norm***
   - ***Positive numeric scalar***: *p*-norm. The value of p can be set with using ***norm*** numerical field.
   - ***Infinity norm***: Infinity norm.
3. ***Scale***
   - ***Standard deviation***: Scale by standard deviation.
   - ***Median absolute deviation***: Scale by median absolute deviation.
   - ***First element of data***: Scale by first element of data.
   - ***Interquartile range***: Scale data by interquartile range.

- ***Numeric values***: Scale data by numeric values. The array must have a compatible size with the input, which PATLab allows to select one of the time series signal in the input data by specifying the detector number and frame number.

4. ***Range***: Rescale range of data to an interval as the two ends can be defined by user as ***a*** and ***b***.
5. ***Center***
    - ***Mean***: Center to have mean 0.
    - ***Median***: Center to have median 0.
    - ***Numeric values***: Shift center by numeric values. The array must have a compatible size with input, which PATLab allows to select one of the time series signal in the input data by specifying the detector number and frame number.

3.2.3 Filtering:

Users can design finite impulse response (FIR) or infinite impulse response (IIR) filters to reduce noise in the data. To apply a filter user should follow these steps:

1. Select the ***Input data*** from the dropdown menu to apply the filter to.
2. Select ***Filter type***. Different types of filters are available such as low-pass, high-pass, band-pass, and band-stop (Figure *3-10*).

⚠ This step requires DSP system toolbox.

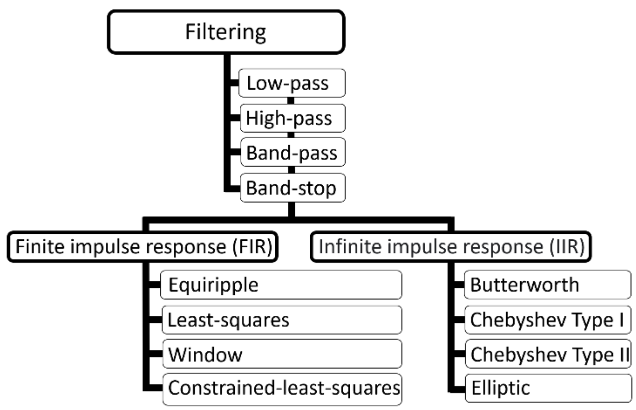


Figure 3-10: Filter options available to PATLab users.

Select ***Method***, finite impulse response (FIR) or infinite impulse response (IIR).

- FIR filter options are:

- Equiripple (Table 3-3 and Figure *3-11*): Parks-McClellan optimal FIR filter design (`firmp` function in MATLAB)
- Least-squares: Least-squares linear-phase FIR filter design (`firls` function in MATLAB). Uses the same parameters as the FIR-equiripple filter.
- Window (Table 3-4 and Figure 3-12): Window-based FIR filter design (`fir1` function in MATLAB).
- Constrained least-squares (Table 3-5 and Figure 3-13): Constrained-least-squares FIR multiband filter design (`fircls` function in MATLAB).

- IIR filter options are:

- Butterworth: (`fdesign` function in MATLAB). This filter uses the same parameters as the FIR-Window filter.
- Chebyshev Type I (Table 3-6 and Figure 3-14): (`fdesign` function in MATLAB).
- Chebyshev Type II (Table 3-7 and Figure 3-15): (`fdesign` function in MATLAB).

- Elliptic (Table 3-8 and Figure 3-16): (`fdesign` function in MATLAB).

3. Select cut off frequency range, (***Cutoff freq. (MHz)*** and ***low*** frequency and ***high*** frequency).
4. Select ***Sampling rate (MHz).*** Must satisfy the Nyquist rate.
5. Select ***Order.***

Table 3-3: Parameters for FIR-Equirepple filters.

Lowpass (Figure 3-11 a):	**Highpass (Figure 3-11 b):**	**Bandpass (Figure 3-11 c):**	**Bandstop (Figure 3-11 d):**
f_1= cutoff freq. ***high*** f_2= cutoff freq. ***highX*** ***Band1*** = weight value for pass band ***Band2*** = weight value for stop band	f_1= cutoff freq. ***lowX*** f_2= cutoff freq. ***low*** ***Band1*** = weight value for stop band ***Band2*** = weight value for pass band	f_1= cutoff freq. ***lowX*** f_2= cutoff freq. ***low*** f_3= cutoff freq. ***high*** f_4= cutoff freq. ***highX*** ***Band1*** = weight value for stop band ***Band2*** = weight value for pass band ***Band3*** = weight value for stop band	f_1= cutoff freq. ***lowX*** f_2= cutoff freq. ***low*** f_3= cutoff freq. ***high*** f_4= cutoff freq. ***highX*** ***Band1*** = weight value for pass band ***Band2*** = weight value for stop band ***Band3*** = weight value for pass band

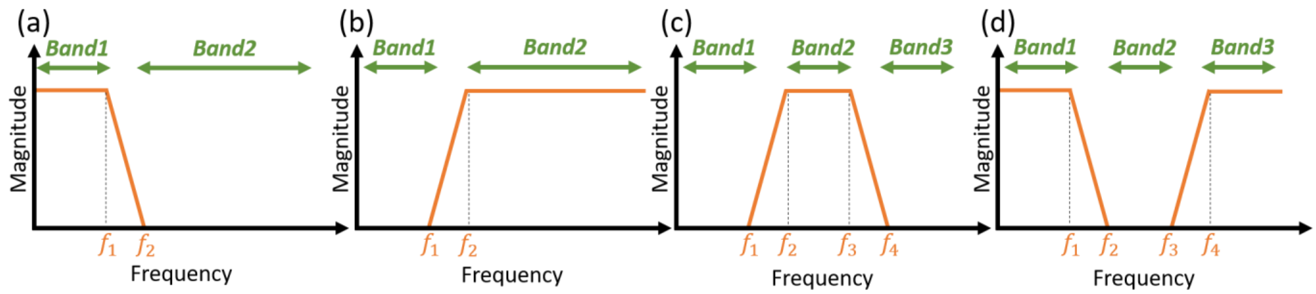


Figure 3-11: FIR-Equiripple and Least-squares. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Table 3-4: Parameters for FIR-Window and IIR-Butterworth filters.

Lowpass (Figure *3-12* a):	**Highpass (Figure *3-12* b):**	**Bandpass (Figure *3-12* c):**	**Bandstop (Figure *3-12* d):**
f_1= cutoff freq. ***high***	f_1= cutoff freq. ***low***	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high***	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high***

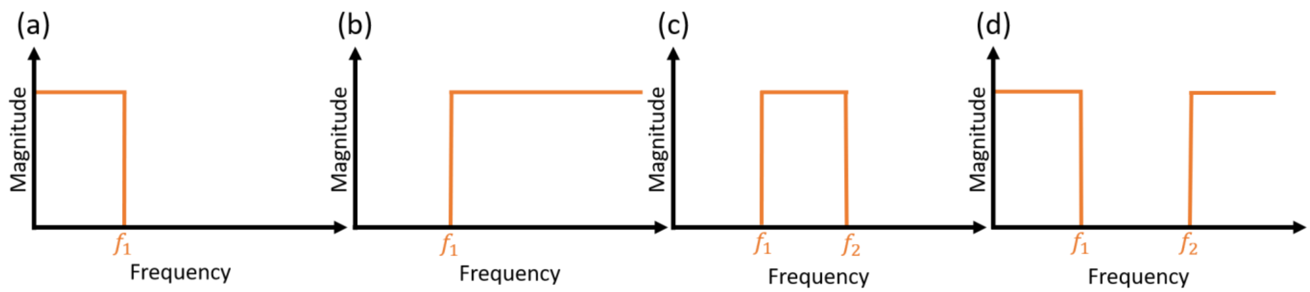


Figure 3-12: FIR-Window and IIR-Butterworth. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Table 3-5: Parameters for FIR- Constrained-least-squares filters.

Lowpass (Figure 3-13 a):	**Highpass (Figure 3-13 b):**
f_1= cutoff freq. ***high*** up=[1+***band1(upper)*** ***band2(upper)***] lo=[1-***band1(Lower)*** -1****band2(Lower)***]	f_1= cutoff freq. ***low*** up=[***band1(upper)*** 1+***band2(upper)***] lo=[-1****band1(Lower)*** 1-***band2(Lower)***]
Bandpass (Figure 3-13 c):	**Bandstop (Figure 3-13 d):**
f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** up=[***band1(upper)*** 1+***band2(upper)*** ***band3(upper)***] lo=[-1****band1(Lower)*** 1-***band2(Lower)*** -1****band3(Lower)***]	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** up=[1+***band1(upper)*** ***band2(upper)*** 1+***band3(upper)***] lo=[1-***band1(Lower)*** -1****band2(Lower)*** 1-***band3(Lower)***]

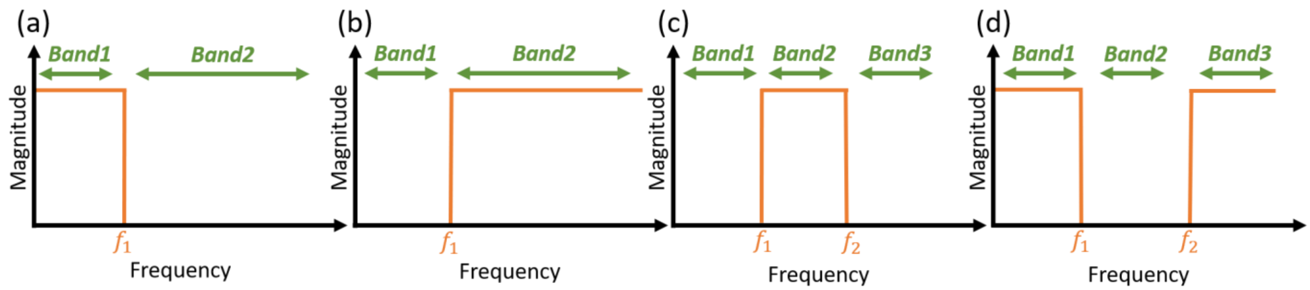


Figure 3-13: FIR-Constrained-least-squares. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Table 3-6: Parameters for IIR Chebyshev Type I filters.

Lowpass (Figure *3-14* a):	**Highpass (Figure *3-14* b):**	**Bandpass (Figure *3-14* c):**	**Bandstop (Figure *3-14* d):**
f_1= cutoff freq. ***high*** ***Apass*** = passband ripple	f_1= cutoff freq. ***low*** ***Apass*** = passband ripple	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Apass*** = passband ripple	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Apass*** = passband ripple

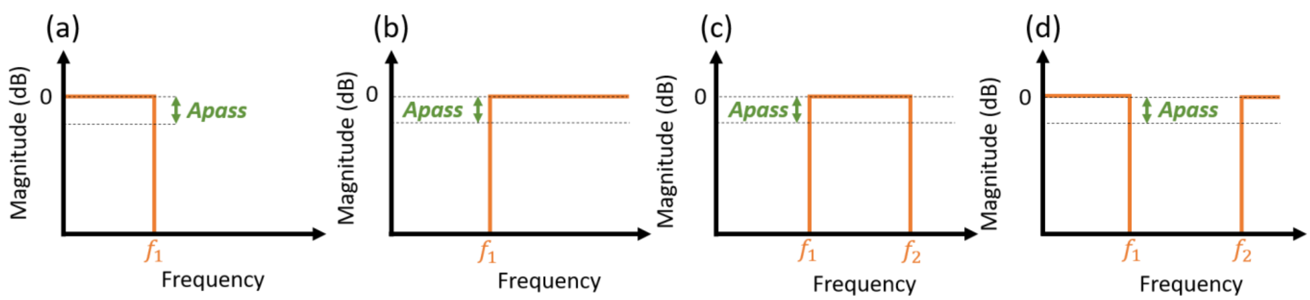


Figure 3-14: IIR Chebyshev Type I. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Table 3-7: Parameters for IIR Chebyshev Type II filters.

Lowpass (Figure *3-15*Figure *3-14* a):	**Highpass (Figure *3-15* b):**	**Bandpass (Figure *3-15* c):**	**Bandstop (Figure *3-15* d):**
f_1= cutoff freq. ***high*** ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Astop*** = stopband attenuation

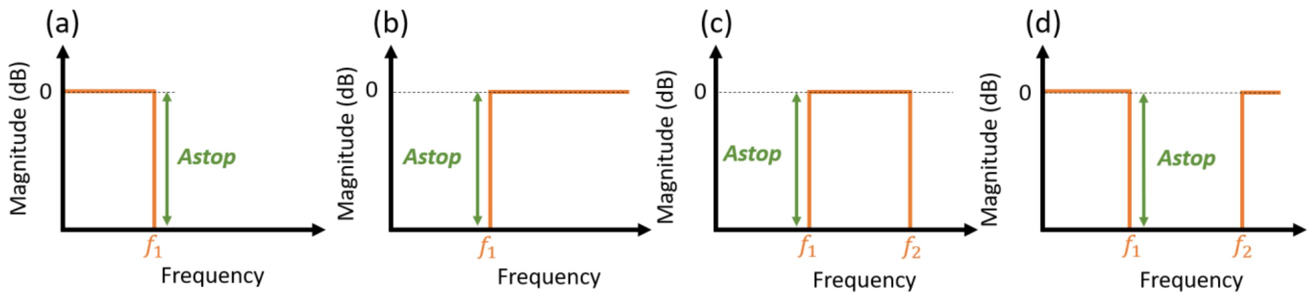


Figure 3-15: IIR Chebyshev Type II. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Table 3-8: Parameters for IIR Elliptic filters.

Lowpass (Figure *3-16*Figure *3-14* a):	**Highpass (Figure *3-16* b):**	**Bandpass (Figure *3-16* c):**	**Bandstop (Figure *3-16* d):**
f_1= cutoff freq. ***high*** ***Apass*** = passband ripple ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** ***Apass*** = passband ripple ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Apass*** = passband ripple ***Astop*** = stopband attenuation	f_1= cutoff freq. ***low*** f_2= cutoff freq. ***high*** ***Apass*** = passband ripple ***Astop*** = stopband attenuation

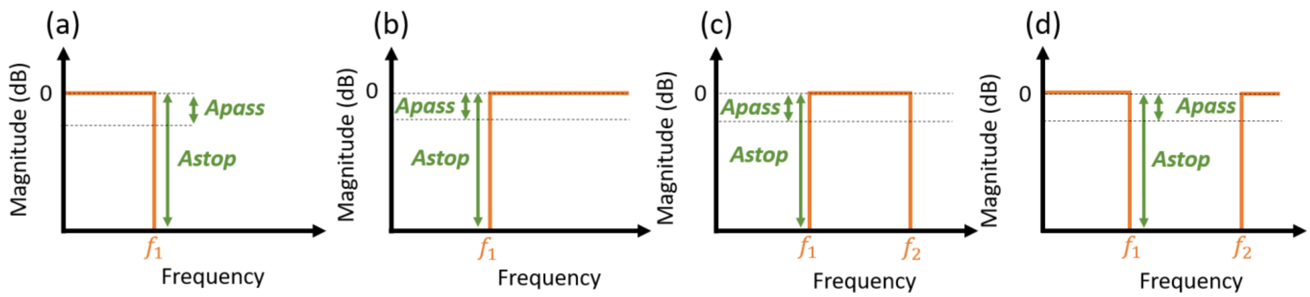


Figure 3-16: IIR Elliptic. **a)** lowpass, **b)** highpass, **c)** bandpass, and **d)** bandstop.

Before applying the chosen filter to the selected input data, it is possible to test it over a selected time series sample. Users can select the time series sample by stating the frame and detector numbers in the field shown in Figure *3-17* and pressing ***Test the filter***.

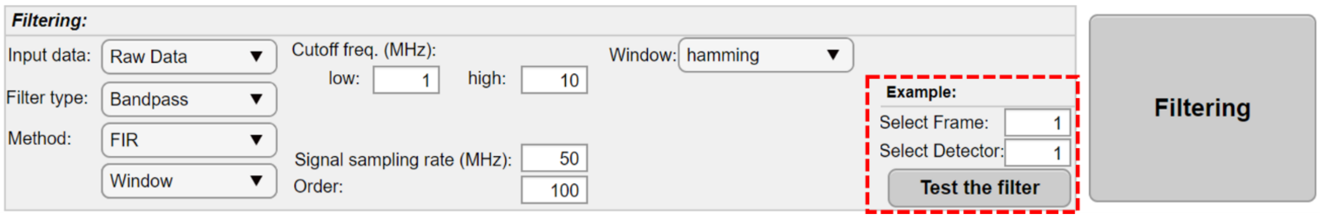


Figure 3-17: Filtering section. The red dashed box shows the field that can select an example of time series data to test the designed filter.

In addition, multiple filters can be applied to the data by filtering the selected data, changing the input data dropdown to ***Filtered Data***, applying a new filter, and so on.

(a) (b)

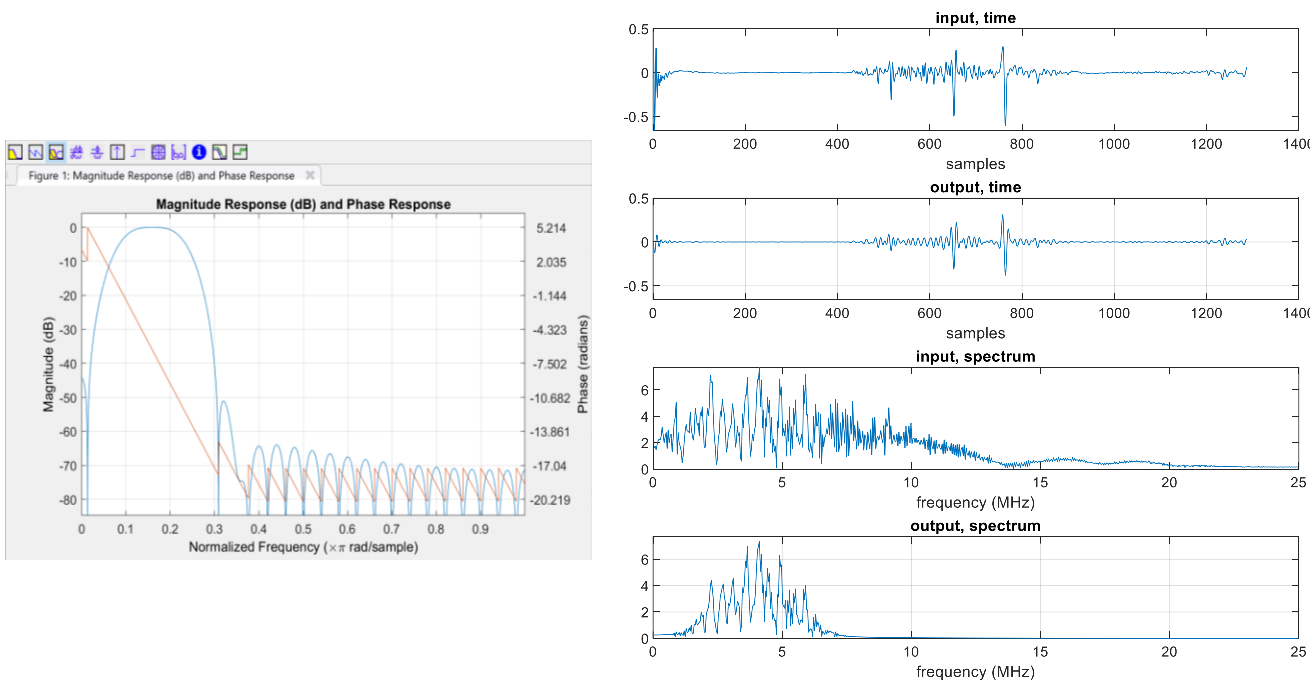


Figure 3-18: Example of designed filter with PATLab (bandpass-FIR-window filter with using hamming as window. Low cut off frequency = 2 MHz, high cut off frequency = 6 MHz, and order of the filter was 50). **a)** Magnitude and phase response of the filter and **b)** the automatic plot displays the original input and output data in the time and frequency domains.

3.2.4 Down-sampling

This panel is designed to decrease the sample rate of the input signal by an integer factor. To doing so, two methods have been implemented:

1. ***Decimation:*** Keep one sample for every k samples of the input time series signal and discard k-1 samples. The ***Factor*** integer field indicates the k and the ***Offset*** numerical field indicates the offset that takes a positive integer from 0 to k.

2. ***Moving average decimation***: Use a moving average over the input time series signal and then apply the decimation method described previously. The moving average window size equals the ***Factor*** numerical field.

3.2.5 Matched Filtering:

In order to improve the signal to noise ratio (SNR) of the time series data, a matched filter may be essential. Matched filters are created by making a correlation between a known template and an unknown signal to detect the template in the signal. There are several ways of determining the template. In a more complex way, the template can be the impulse response for the transducers used to detect the signals. In a simpler way, the template can be a portion of the unknown signal that show a desire noise free signal.
PATLab offers four options to be used as template:

3. ***Select a range of the signal as template***.
    - ***Use a unique template for each time series***: By specifying a range (i.e. ***Start data sample*** and ***End data sample***), each time series uses that range of itself as template. The desire range can be selected by visualizing examples from ***Example***.
    - ***Use one template for all time series***: In this case a time series must be selected (from ***Select a Signal***) and a desire range must be specified (i.e. ***Start data sample*** and ***End data sample***) to find a desire template in that time series. The matched filter must be generated before applying that filter to the entire data.
4. ***Select a part of the signal containing the global maximum as template***.
    The template here is the part of the time series which contain the global maximum. Users can select the ***# of the samples before the max*** and ***# of the samples after the max.*** Same as previous, there is 2 options to use the desire template:
    - ***Use a unique template for each time series***
    - ***Use one template for all time series***

5. ***Load a 1D template (time domain)***.
    This option allows users to browse their own template as a 1D signal to be used for the input time series. The 1D signal must be a time series of data samples saved as .mat file. The number of samples are not restricted.

    ⚠ If all time series data are acquired with one detector, then that detector's pulse response can be used here.
6. ***Load a 2D template (time domain)***.
    This option allows users to browse their own template as a 2D signal. The 2D signal must be a .mat file with one dimension indicating the detectors and another dimension indicating the samples (the order of dimensions is not crucial and can be set using a designed tool). The number of detectors must match the number of detectors assigned in section 3.1.3 or section 3.1.4.

    ⚠ When the time series has been acquired with multiple detectors, it is possible to load a 2D matrix of pulse responses for all the detectors with this option.

It is possible to test the matched filter over a selected time series example before applying it to the selected input data. Users can select the time series example by stating the frame and detector numbers in the field shown in Figure *3-19* and pressing ***Test the filter***. As a result, a figure with 4 panels similar to Figure *3-20* appears to show the matched filtering procedure. The top-left panel shows the input time series signal with 5120 data samples. The top-right panel displays the template containing 10240 samples. In this example, the template is the detector's pulse response that has been loaded using the third method (i.e. Load a 1D template (time domain)). Convolution of the input signal with the template will be calculated next (not shown in the figure). Afterwards, the maximum values of the convolution result are identified using a sensitivity indicator (i.e. a numerical value from 0 to 1, higher value indicate less maximums). Finally, the local maximums and the flipped template must be convolved together to produce the final signal (i.e. matched filtered input signal) as displayed in the bottom right panel.

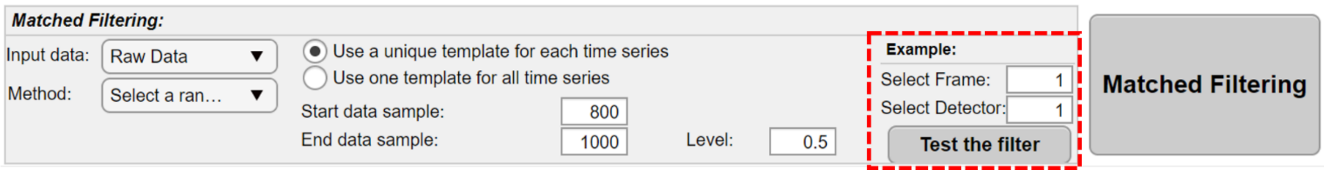


Figure 3-19: Matched filtering section. The red dashed box shows the field that can select an example of time series data to test the matched filter.

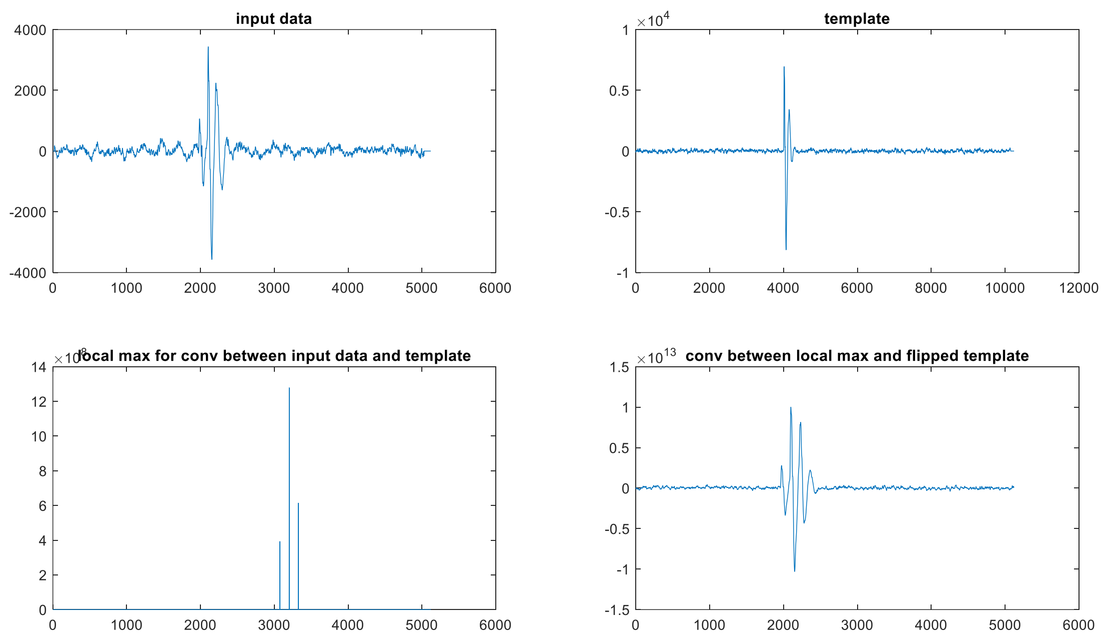


Figure 3-20: An example of matched filtering for a raw input time series signal (top-left) and a template signal (top-right). PATLab generates local maximums for convolution between the input signal and the template (bottom-left). Finally, convolve the flipped template to the local maximums to produce the final signal (bottom-right).

3.3 Image Reconstruction

Image Reconstruction tab encompasses 5 panels to define the image characteristics, the input signal, and the reconstruction algorithm (Figure *3-21*).

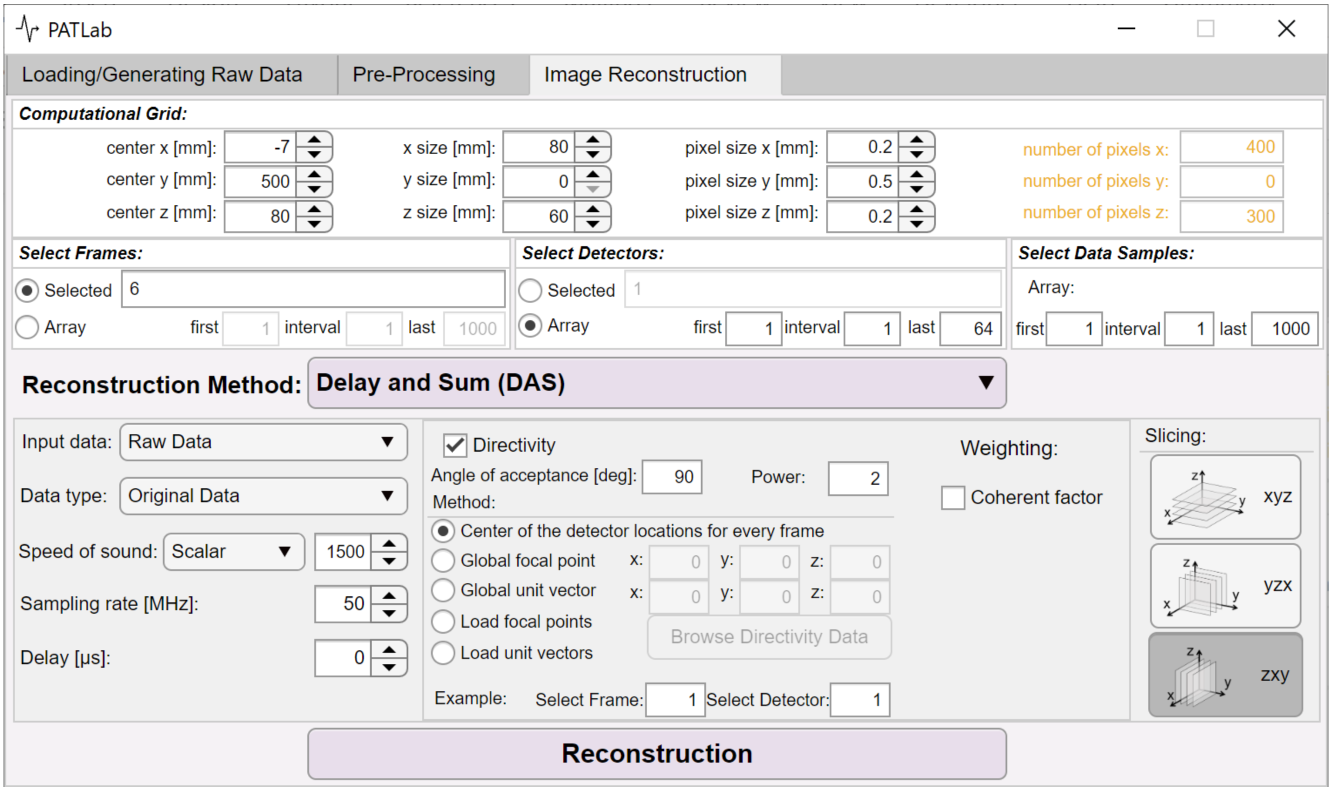


Figure 3-21: Image reconstruction panel.

3.3.1 *Computational Grid*:

This panel takes the information for the imaging area such as:

- ***center xyz [mm]***: the center of the area,
- ***xyz size [mm]***: the xyz dimensions,
- ***pixel size xyz [mm]***: the size of the pixels in xyz coordinates,
- ***number of pixels xyz*** = `round` (***xyz size [mm]***/ ***pixel size xyz [mm]***)

✓	The ***number of pixels xyz*** are “only read” fields to demonstrate the number of pixels for the output reconstructed image.
✓	Having 2 non-zero values for ***pixel size xyz [mm]*** will provide a 2D image.
✓	Having 3 non-zero values for ***pixel size xyz [mm]*** will provide a 3D image (a multi-slice image).

3.3.2 *Select Frames*:

This panel provides two approaches to select a range of frames:

- ***Selected*** button: by entering only the desired frames (separated by comma or space), or
- ***Array*** button: by entering three inputs matching a *for* loop to generate a desired array:

```
selectedArray = inputData (first: interval: last);
```

3.3.3 *Select Detectors*:

This panel provides two approaches to select a range of detectors (refer to the ***Select Frames***).

3.3.4 ***Select Data Samples***:

This panel provides one approach to select a range of detectors (refer to the second approach in ***Select Frames***).

3.3.5 **Reconstruction**:

This panel offers multiple algorithms to reconstruct an image from a set of selected data. Two dropdown options are included in all algorithms, allowing users to pick the data they want.

1- ***Input data:***
- ***Raw Data***
- ***Cleaned Data***
- ***Normalized Data***
- ***Filtered Data***
- ***Downsampled Data***
- ***Matched Filtered Data***

⚠ Before setting input data, make sure it has been prepared in the pre-processing step.

2- ***Data type:***
- ***Original Data:*** no alteration
- ***Inverse:*** Inverse the signal amplitude with reference to the zero value.
- ***Envelope:*** `abs(Hilbert(selectedData))`

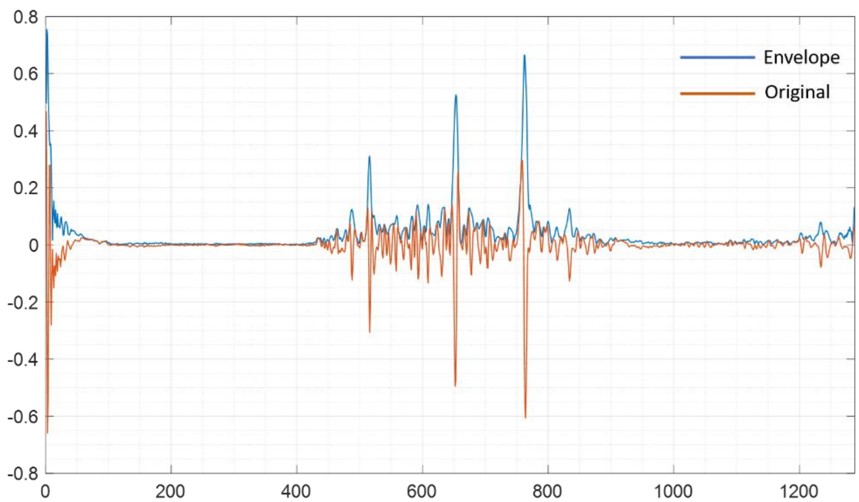


Figure 3-22: Example of the signal Envelope.

- ***Positive:*** Only the positive parts of the time series are considered, and the negative parts are thresholded into zero.
- ***Negative:*** Only the negative parts of the time series are considered, and the positive parts are thresholded into zero.
- ***Absolute Value:*** `abs(selectedData)`
- ***Complex-Hilbert:*** uses Hilbert transform to generates complex time series.

3.3.5.1 Delay and Sum (DAS)

- ***Speed of sound [m/s]:*** scalar value that represents the speed of sound.

- ***Sampling rate [MHz]:*** scalar value that represents the sampling rate of the time series.
- ***Delay [µs]:*** a scalar value that indicates the presence of a delay in the time series.
- ***Directivity:*** a checkbox that enables five ways of applying directivity information to the transducers.
    1. ***Center of the detector locations for every frame:*** consider the center of the array for each frame. This method is recommended for circular and spherical configurations where the detectors all have focal points at the center.
    2. ***Global focal point xyz:*** consider a point as the global focal point for all detectors.
    3. ***Global unit vector xyz:*** consider that all detectors have the same directivity. The directivity can be represented as a global unit vector. This method is recommended for linear and planar configurations where all detectors face in the same direction.
    4. ***Load focal points:*** a 3D matrix can be loaded containing the focal point information about each detector according to any frame position. This matrix must contain the same number of detectors, frames, and coordinates (each dimension represents one of them) as the used detector locations in section 3.1.1 or section 3.1.2. There are two coordinates for 2D coordinate systems and three for 3D coordinate systems. A pop-up interface is then provided for assigning each dimension to the respective roles by selecting corresponding options from the dropdown menus (Figure *3-23*).
    5. ***Load unit vectors:*** a 3D matrix can be loaded containing the unit vector information about each detector according to any frame position. This matrix must contain the same number of detectors, frames, and coordinates (each dimension represents one of them) as the used detector locations in section 3.1.1 or section 3.1.2. There are two coordinates for 2D coordinate systems and three for 3D coordinate systems. A pop-up interface is then provided for assigning each dimension to the respective roles by selecting corresponding options from the dropdown menus (Figure *3-23*).

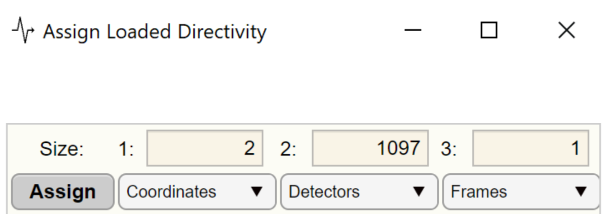


Figure 3-23: Interface to assign each dimension to the corresponding roles when users load focal points or unit vectors as directivity.

- ***Full angle of acceptance (degree):*** indicates the acceptance angle for the transducers (it will be enabled if the ***Directivity*** option has been selected).
- ***Coherent factor:*** a check box that enables to apply coherent factor weighting method.

3.3.5.2 Universal Back Projection (UBP)

- Same with ***delay and sum (DAS)*** method***.***
- The UBP algorithm allocates a gain for the derivation engagement described as an extra numerical field (named ***derivative gain***).

3.3.5.3 Filtered Back Projection (FBP)

- Same with ***Delay and Sum (DAS)*** method***.***

3.3.5.4 Fourier Transform (FT)

This method of reconstruction is adapted from k-Wave toolbox and it is only operational for linear array and planar array configurations for the current version of the PATLab.

✓ In this method of reconstruction, the location of detectors is not needed.

- ***Speed of sound [m/s]:*** scalar value that represents the speed of sound.
- ***Sampling rate [MHz]:*** scalar value that represents the sampling rate of the time series.
- ***Delay [µs]:*** a scalar value that indicates the presence of a delay in the time series.
- ***Configuration***: allows users to select between ***linear array*** and ***planar array*** configurations.
- ***Sampling pitch [mm]:*** a non-editable field that shows the space interval between adjacent samples in the selected signal.

`= Sampling rate * Speed of sound`

- ***Number of available detectors***: by default, it calculates the number of detectors based on the ***Select Frame*** and ***Select Detector,*** and it is also possible to be modified.

1. Linear array configuration

- ***Length of the array [mm]:*** This property must represent the length of the array that has been used for data collection.
- ***Detectors pitch [mm]:*** `= Length of the array/(Number of available detectors - 1).`

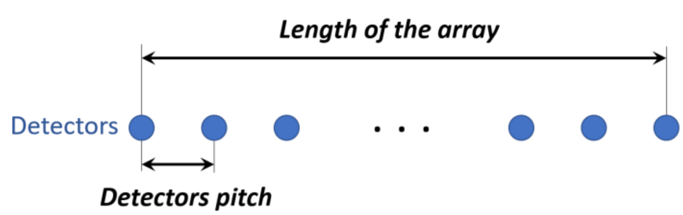


Figure 3-24: Linear array configuration

2. Planar array configuration

- ***Length of the array [mm]:*** This property must represent the length of the array that has been used for data collection.
- ***Number of detectors in length:*** Number of detectors in a single linear array (the detectors with the same color in Figure 3-25).
- ***Pitch in length [mm]:*** `= Length of the array/(Number of detectors in length - 1).`
- ***Width of the plane [mm]:*** The width of the array that has been used for data collection.
- ***Number of detectors in width:*** Number of linear arrays were placed along the width of the plane (e.g., 4 in Figure 3-25).
- ***Pitch in width [mm]:*** `= Width of the plane/(Number of detectors in width - 1).`

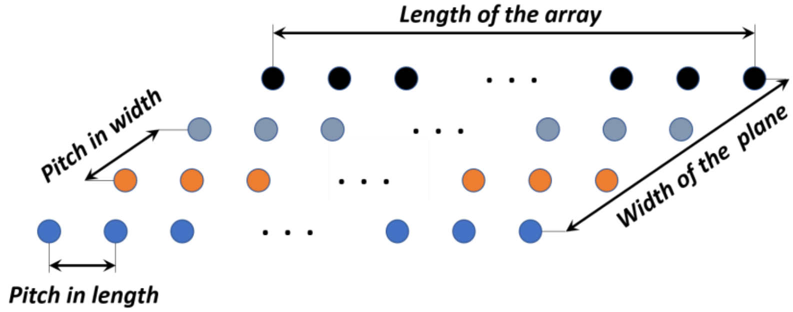


Figure 3-25: Planar array configuration

- ***Interpolation (nearest or linear):*** A recommended option in cases where the detectors pitch (or pitch in length/width for planar configuration) is larger than the sampling pitch.

3.3.5.5 Time Reversal (TR)

This method of reconstruction is adapted from k-Wave toolbox.

3.4 Plot & Save:

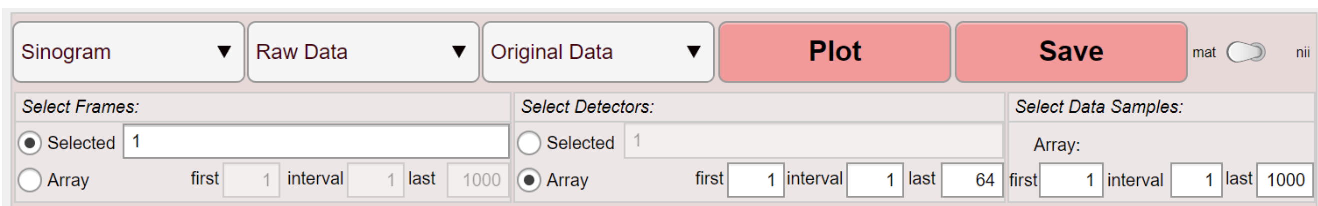


Figure 3-26: Plot & Save panel

This panel is fixed and located at the bottom of the software and allows the user to visualize and save several available options.
The first dropdown from the left includes all four graphical representations of data that PATLab provides.

3.4.1 Sinogram:

- ➢ The ***Sinogram*** plot represents multiple selected time series with respect to the detectors (channels) in a 2D format.
- ➢ It is possible to select among time series by using tools such as:

1- ***Select Frames***: This panel provides two approaches to select a range of frames:

- ***Selected*** button: by entering only the desired frames (separated by comma or space), or
- ***Array*** button: by entering three inputs matching a *for* loop to generate a desired array:

```
selectedArray = inputData (first: interval: last);
```

2- ***Select Detectors***: This panel provides two approaches to select a range of detectors (refer to the ***Select Frames***).

3- ***Select Samples***: This panel provides one approach to select a range of detectors (refer to the second approach in ***Select Frames***).

- Choosing sinogram enables a dropdown menu that allows selecting input data (***Raw Data***, ***Cleaned Data***, ***Normalized Data***, ***Filtered Data***, ***Downsampled Data***, or ***Matched Filtered Data***) and another dropdown menu that allows selecting the shape of the selected input data (***Original***, ***Inverse***, ***Envelope***, ***Positive***, ***Negative***, ***Absolute Value***, or ***Complex***).

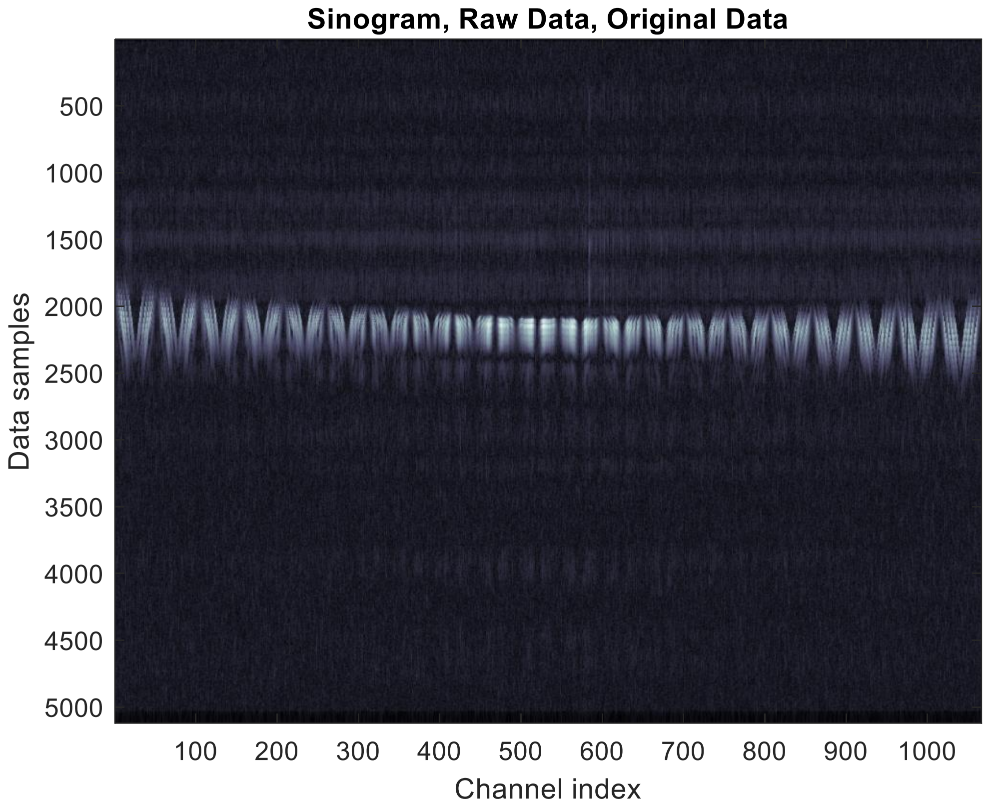


Figure 3-27: An example of Sinogram display consists of 1066 detectors (channels) each with 5120 data samples. The Raw Data with its Original shape has been selected.

3.4.2 *Time series:*

- The ***Time Series*** plot displays the amplitude against time (samples) of selected time series in a 1D format.
- With the designed tools (refer to ***Sinogram***), it is possible to choose between time series.
- Just like ***Sinogram***, this plot enables input data to be selected via a dropdown and the data shape to be selected via another dropdown.

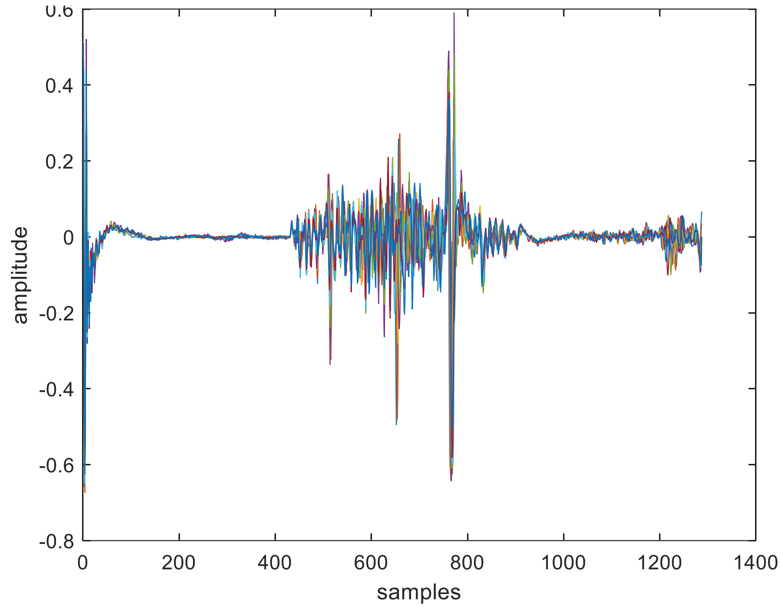


Figure 3-28: An example of Time Series display consists of 8 detectors (channels) each with 1287 data samples. The Raw Data with its Original shape has been selected.

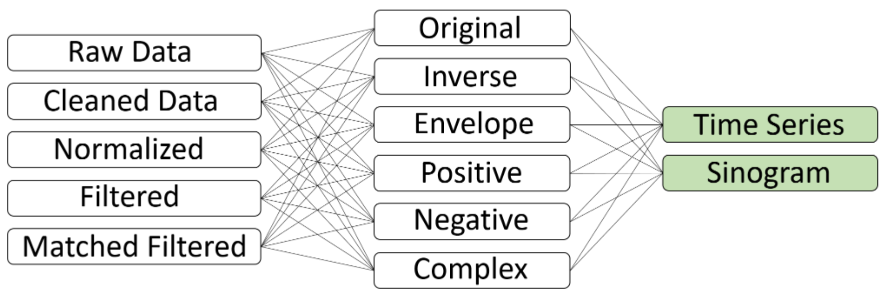


Figure 3-29: All options for choosing the input data and the data shape for Sinogram and Time Series displays.

3.4.3 Detector Locations:

- This plot represents the location of the detectors in a 2D or 3D coordinates format.
- It is possible to select among frames and detectors:
  1. ***Select Frames***: This panel provides two approaches to select a range of frames:
     - ***Selected*** button: by entering only the desired frames (separated by comma or space), or
     - ***Array*** button: by entering three inputs matching a *for* loop to generate a desired array:

```
selectedArray = inputData (first: interval: last);
```

  2. ***Select Detectors***: This panel provides two approaches to select a range of detectors (refer to the ***Select Frames***).

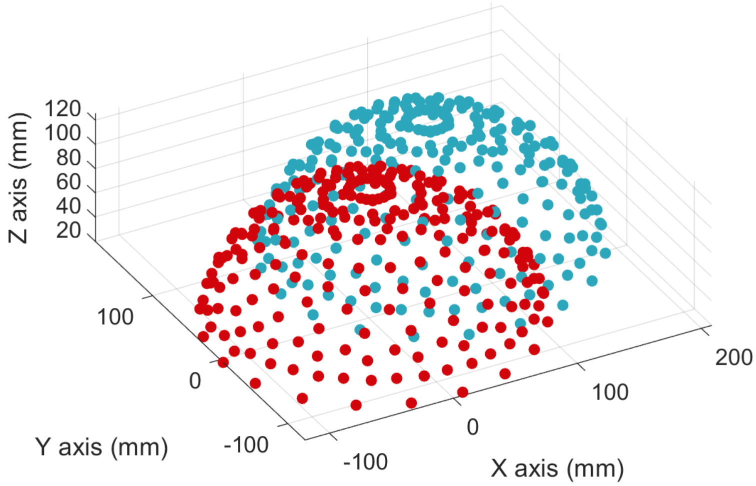


Figure 3-30: An example 3D plot of Detector Locations contains 2 frames, each has 240 detectors arranged in a hemispherical array. The same color will be used to illustrate all detectors in a particular frame.

3.4.4 Reconstructed Image:

Using this option will enable you to visualize the reconstructed image in a new GUI with several tools (Figure *3-32*).

In 2D scenarios, the GUI contains 1 figure to display the reconstructed image. A color bar is presented in the figure to show the image's intensities.

In 3D scenarios, the GUI contains 3 figures initialized with the maximum intensity projection (MIP) of the reconstructed image. Figure *3-31* depicts the xy, zy, and xz coordinates of the image. The xy figure is accompanied by a color bar which indicates the intensity of the xy corresponding image.

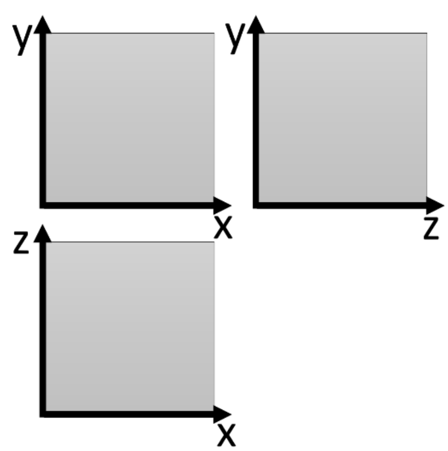


Figure 3-31: Arrangement of 3 figures for 3D scenarios.

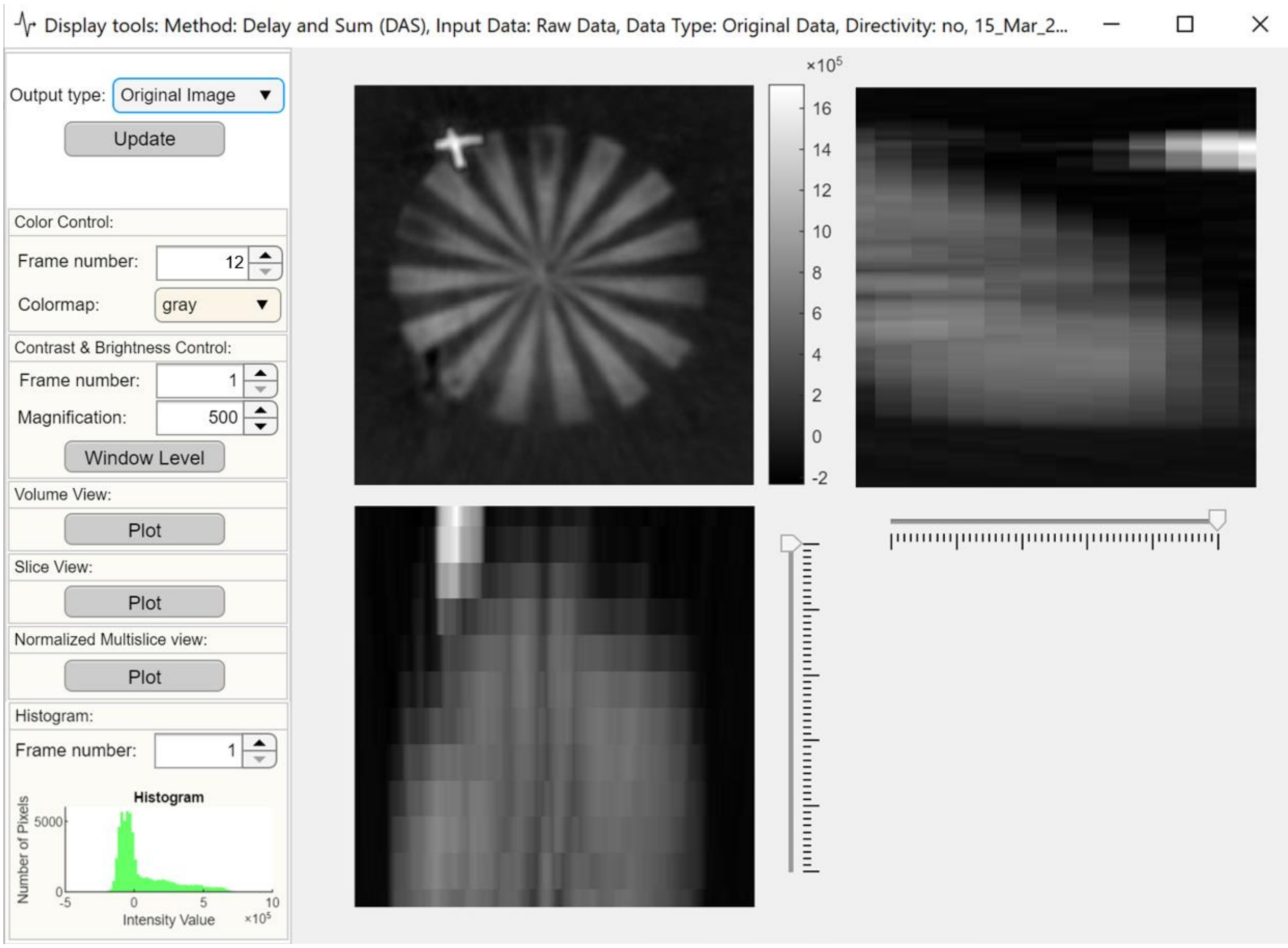


Figure 3-32: An example of displaying a 240*240*12 reconstructed image.

A number of tools are available on the left sidebar of the GUI.

1. ***Color Control:*** Defines the colormap for figure/s in the main panel.

➢ ***In*** the figure depicted by the xy axis, when the ***Frame number*** is changed, the 3D image will display the xy slices that correspond to that value.

➢ ***Frame number*** is only is enabled for 3D scenarios.

➢ The ***Frame number*** value by default indicates the maximum number of slices in z.

2. ***Contrast & Brightness Control:*** Provides a control over the contrast and brightness for a single slice with a function called `WindowLevel` written by H.J. Wisselink.
   This function allows you to adjust the window and level (contrast and brightness) of images by dragging.

⚠ The colorbar on the WindowLevel reflect the resized values of the image, not the original values.

➢ ***Frame number*** is only is enabled for 3D scenarios.

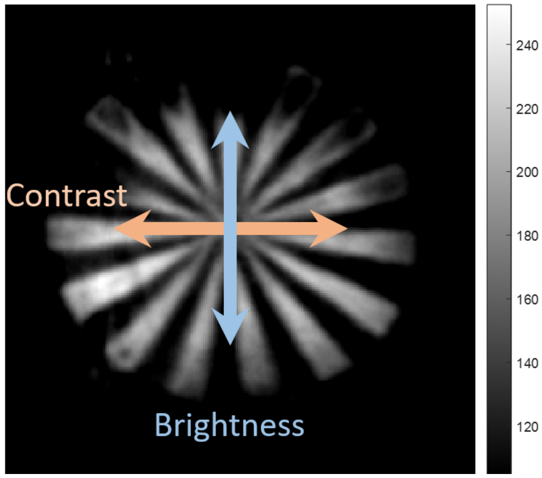


Figure 3-33: An example of the WindowLevel control; drag left-right to alter the contrast and drag up-down to alter the brightness (`WindowLevel` function written by H.J. Wisselink).

3. ***Volume View:*** Opens the reconstructed image in the MATLAB `volumeViewer` app.

⚠ Only functional for 3D scenarios.

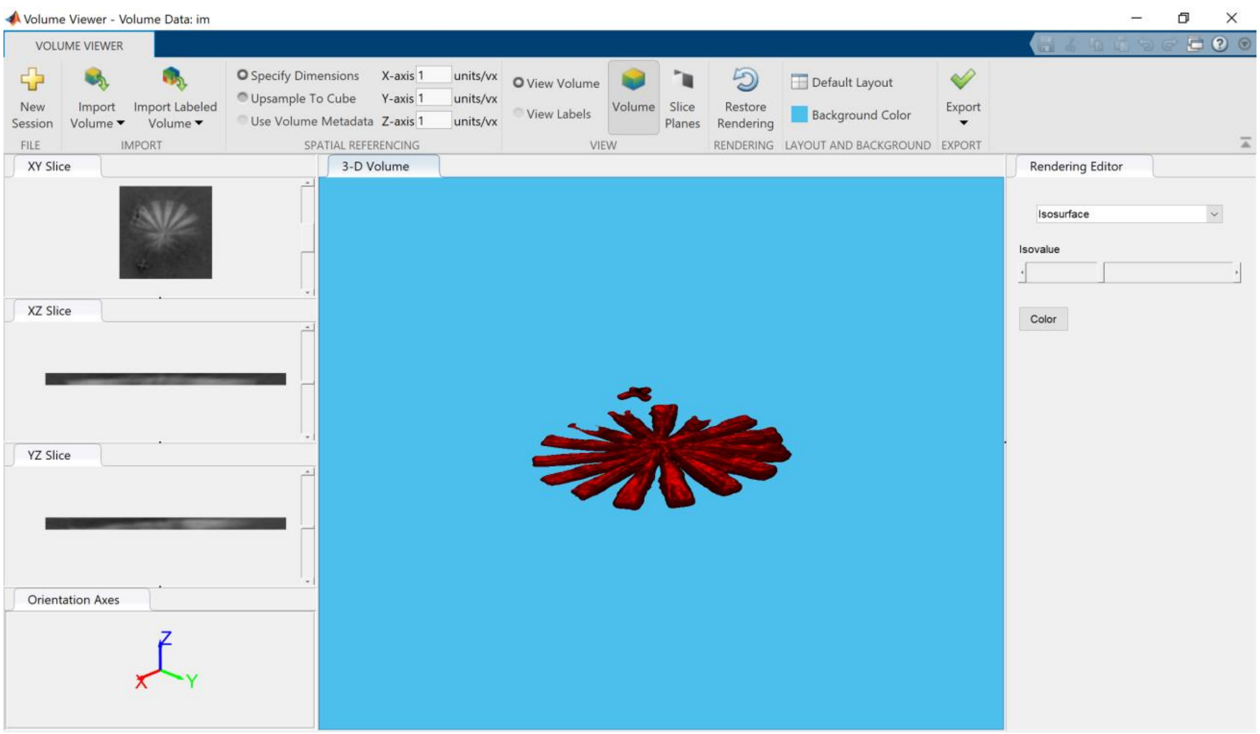


Figure 3-34: An example of Volume View for a 3D image (MATLAB `volumeViewer` app).

4. ***Slice View:*** Opens the reconstructed image in the MATLAB `sliceViewer` app.

⚠ Only functional for 3D scenarios.

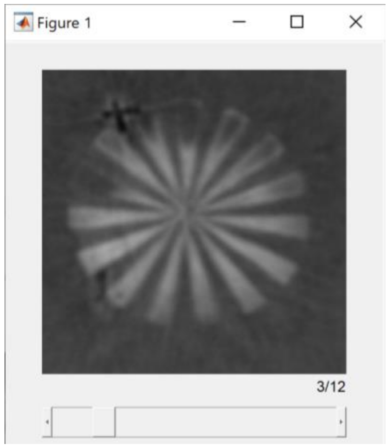


Figure 3-35: An example of Slice View for a 3D image (MATLAB `sliceViewer` function).

5. ***Normalized Multi-Slice View:*** Displays all slices side-by-side in a universal normalized scale.

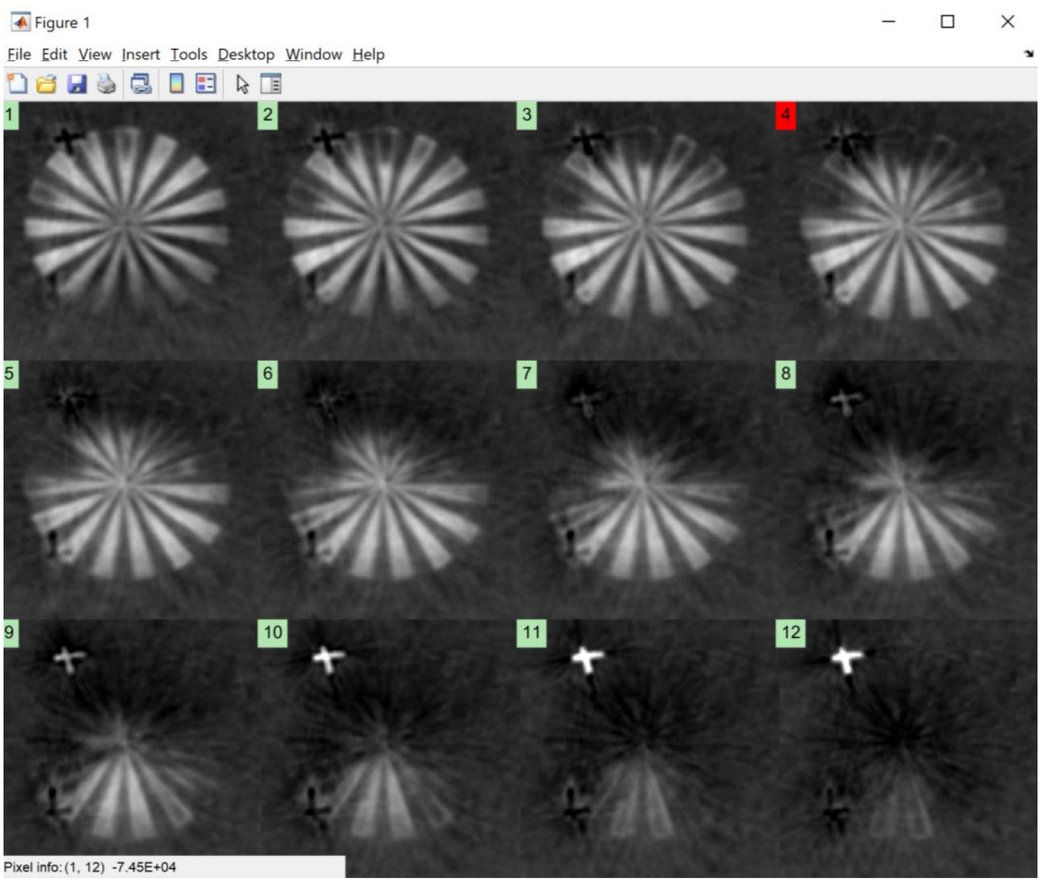


Figure 3-36: An example of Normalized Multi-Slice View for a 3D image.

6. ***Histogram:*** Displays a histogram for the selected slice (***Frame number***).

4. Examples

5. Appendix

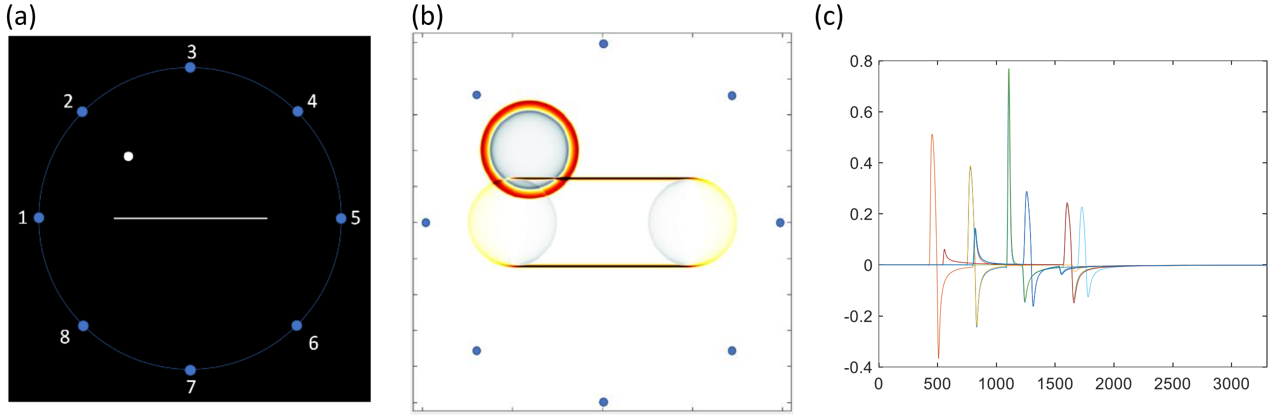


Figure 5-1

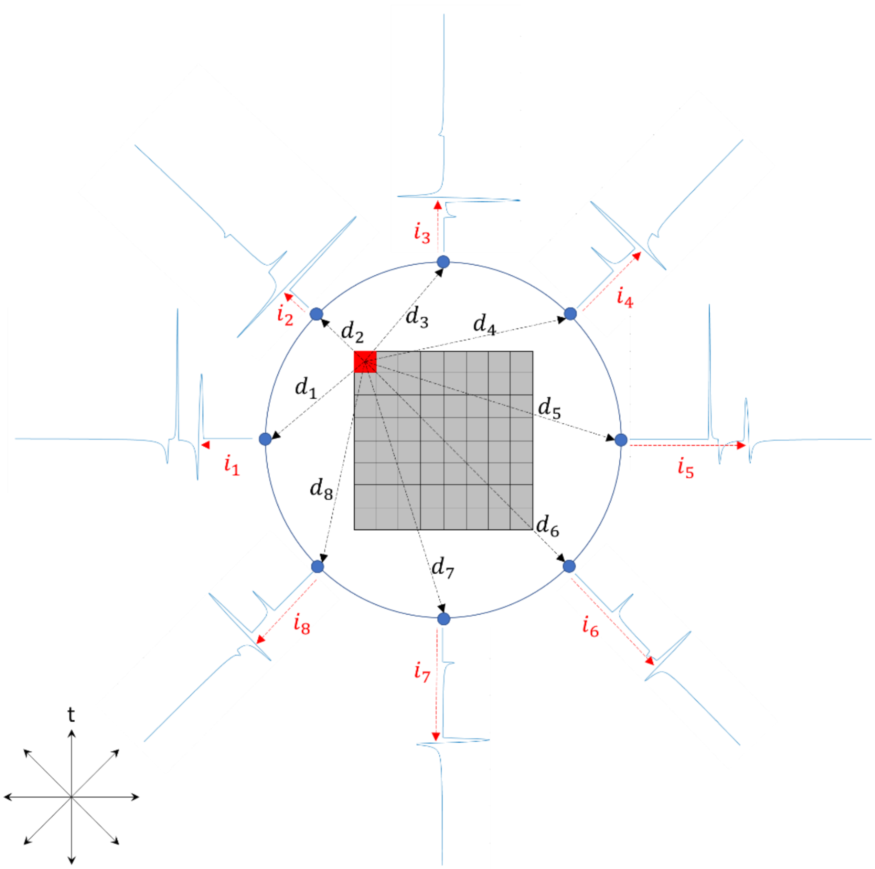


Figure 5-2: Principle of DAS, FBP, and FBP algorithms.

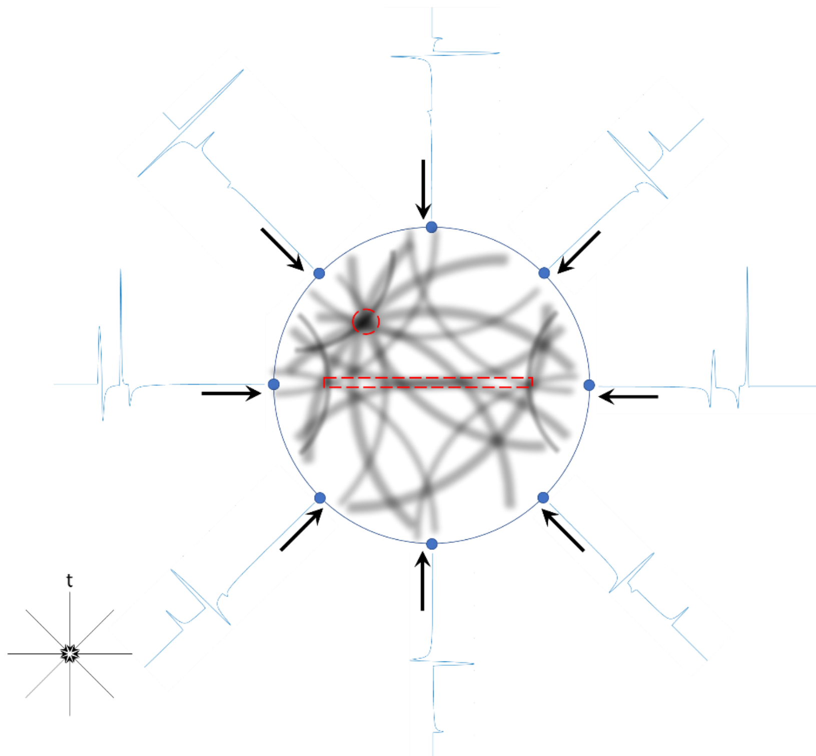


Figure 5-3: Principle of TR algorithm.