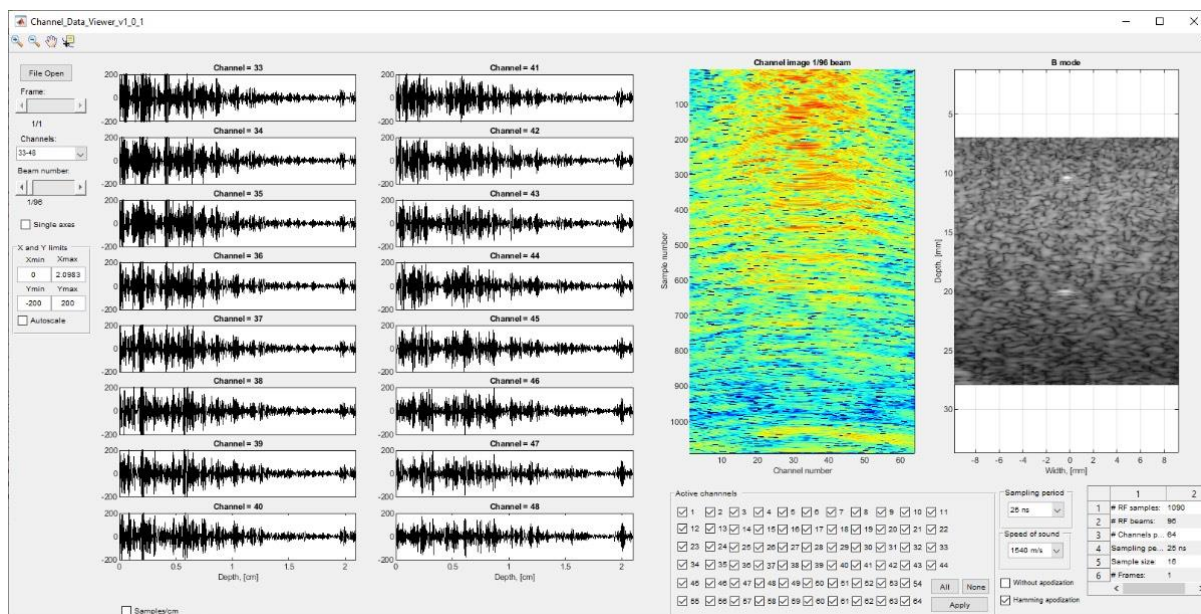


TELEMED Channel Data Viewer

User Manual



TELEMED

Ultrasound Medical Systems

Internet page:

Information E-mail:

Support E-mail:

www.pcultrasound.com

info@pcultrasound.com

support@pcultrasound.com

Contents

Introduction	3
1. ArtUs RF Data Control II (2.0.0 and later)	4
1.1 ArtUs RF Data Control II interface options for Channel Data recording	4
1.2 Structure of the channel data file	5
2. MATLAB tools installation notes	8
3. Description and structure of Channel Data Viewer graphical user interface	9
3.1 Main window and controls	9
4. GUI functions and options	11
4.1 MAIN GUI function (Channel_Data_Viewer.m)	11
4.2 Import channel data to MATLAB (importCH_RFData2MATLAB.m)	11
4.3 DAS Beamforming function (DAS_beamforming.m)	12
4.4 Channel data plotting function (show_plots_of_channel_data.m)	13
4.5 Imaging of B mode and single channel image (visualization.m)	13
4.6 Band-pass filters (band_pass_filter.m)	13
4.7 Logarithmic compression (envelope_2_image.m)	14
4.8 Rotation matrix (rotation_matrix.m)	14
5. Revision history	16
6. References	17

Introduction

TELEMED released ArtUs scanner with an extended functionality to receive raw channel data in real-time and to program excitation pulses and delays of them for the transmitting apertures. Starting from version 2.0.0 of ArtUs RF Data Control program you can receive channel data stream and record the data into bin files for analysis and development. The Channel Data Viewer is a tool for the revision of recorded channel data offline written in MATLAB programming language. The goal of the program is to provide a possibility to review the data, and to get knowledge on the basics of simple B mode image beamforming (delays calculation, apodization etc.).

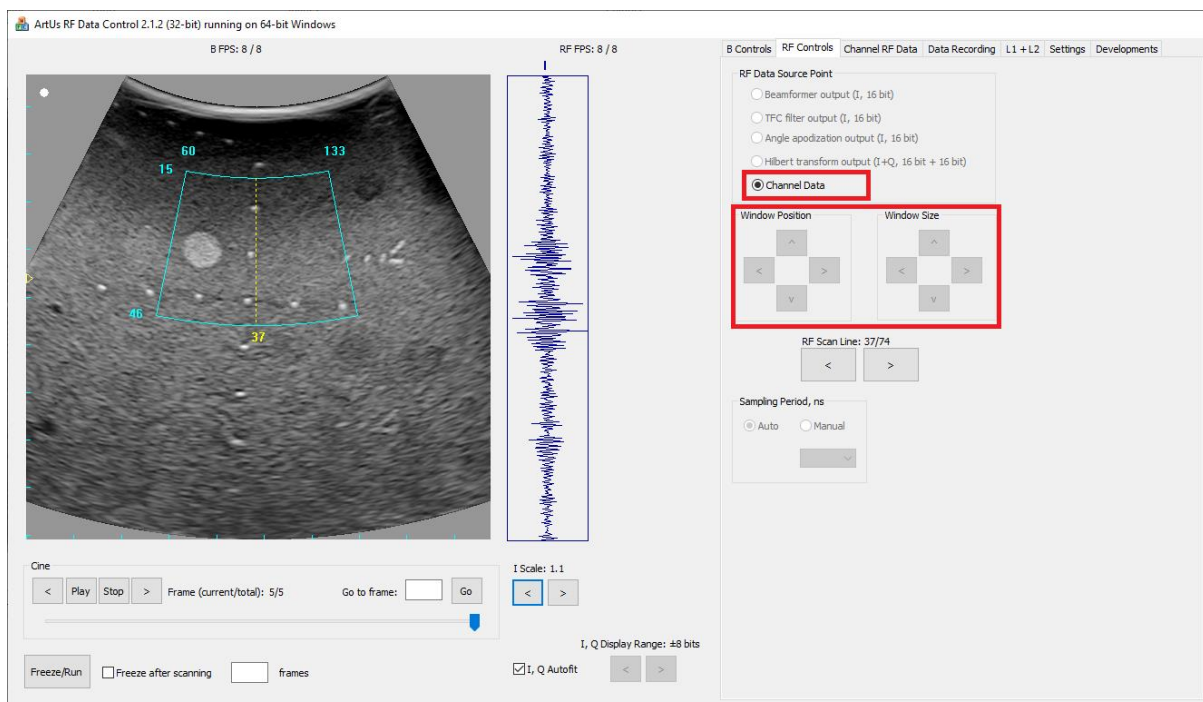
The channel data has the potential to be used for different applications, such as the development of beamforming algorithms, custom apodization methods, custom transmit-receive focusing, and so on.

1. ArtUs RF Data Control II (2.0.0 and later)

The native C++ SDK sample **ArtUs RF Data Control II** should be used for the channel data capture (beamformed RF is also possible), and recording into the data files. The sample allows users to control all the parameters of ultrasound scanner *ArtUs* and to define the dimensions of the channel data stream.

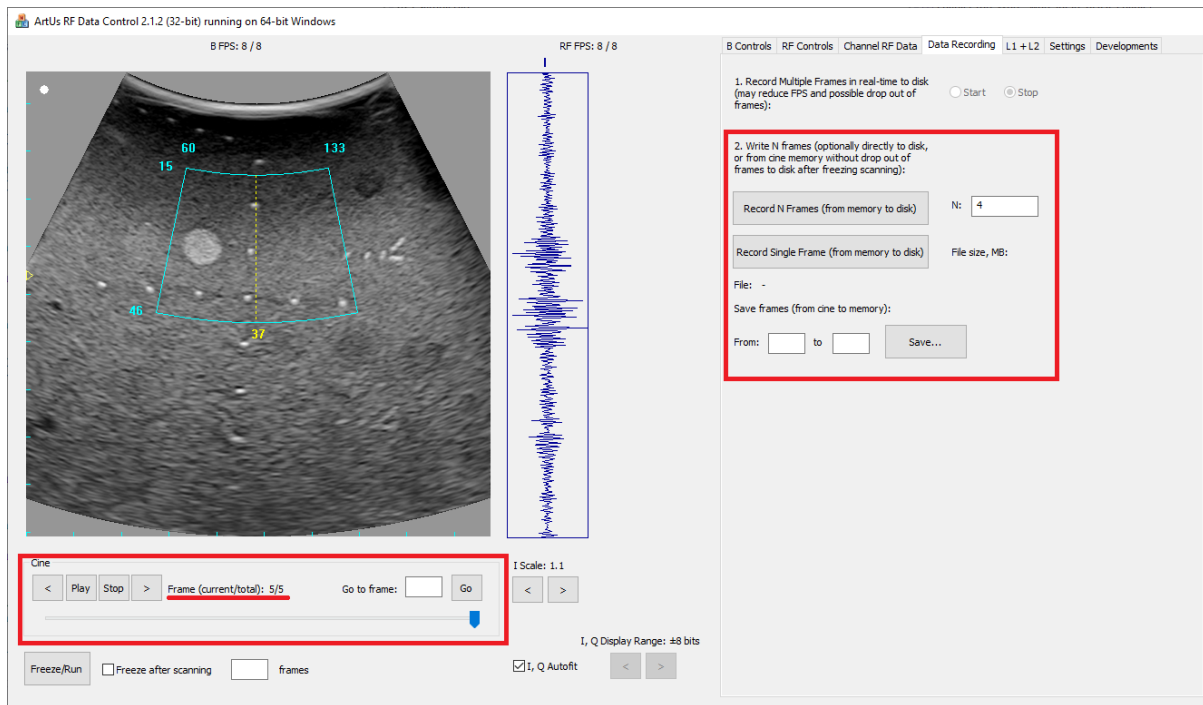
1.1 ArtUs RF Data Control II interface options for Channel Data recording

This user manual gives only essential guidelines on how to record the channel data and a description of the structure of the recorded files. For explicit knowledge of the functionality of the ArtUs RF Data Control II program, we invite you to read the ArtUs RF Data Control User Manual [1]. The picture below shows a screenshot of ArtUs RF Data Control II which from version 2.0.0 was reorganized into tabs structure of controls.



Channel data mode could be turned on by choosing the appropriate RF Data Source Point (**radio button Channel Data**) in the **RF Controls tab**. It is possible to adjust the size of the Channel data window by using **Position** and **Size** controls.

It is possible to record bin files of channel data after the scanning is frozen, so the sequence is about such: 1) scan the desired region (B mode images on the left of GUI are used for navigation purposes), 2) press freeze to stop scanning, 3) use recording options and cine loop controls for channel data bin file creation (recording buttons will be disabled during real-time scanning). For the beamformed RF Data, it is possible to make a recording during scanning, meanwhile, for Channel data, you can record retrospectively to keep the scanning speed as high as possible. The picture below shows the **Data Recording tab** of ArtUs RF Data Control II program. For channel data, there are 3 options: 1) to **record N frames** (**Cine controls** show and indicate how many frames are in the memory buffer, and you have to move the **Cine slider** to the first frame you wish to record and N should not exceed the buffer size), 2) option to **record single frame** (current), and 3) option to record frames from cine loop **From-To** by entering exact numbers of the beginning and end frames.



1.2 Structure of the channel data file

2.2. Structure of the channel data file

Channel data BIN file as beamformed RF data file contains header and channel data. The main difference is that header information is written only once before channel data frames and not before each channel data frame in contrary to beamformed RF data files. The header contains information which is mandatory for B mode image beamforming from channel data. Channel data could only be written after freezing scanning – retrospectively from Cine Loop memory to preserve data transfer speed. The file structure for channel data defined as follows:

Contents of the file	Data type	Number of elements (bytes)	Description
<i>RF file type</i>	char	1 (6 bytes)	Channel RF file version (currently CH0001).
<i>flag</i>	int32	1 (4 bytes)	Flag which indicates if the tag of beam vectors position and start depth information exists (if exists flag = 1, otherwise flag = 0). The flag = 1 for TELEMED scanning modes (standard, wide view, and compound), meanwhile for custom transmit delays or beams information on beam position and orientation, and start depth is not provided.
<i>sampling_period</i>	int32	1 (4 bytes)	Sampling period value in nanoseconds.
<i>BeamsPerFrame</i>	int32	1 (4 bytes)	Number of scanned beams per frame.
<i>SamplesPerChannel</i>	int32	1 (4 bytes)	Number of samples for each channel.
<i>ChannelsPerBeam</i>	int32	1 (4 bytes)	Number of active channels for each beam.
<i>SampleBitCount</i>	int32	1 (4 bytes)	Number of BITS of channel RF Data.
<i>FrameSize</i>	int32	1 (4 bytes)	Size in bytes of channel RF Data frame.
<i>SubFramesNumber</i>	int32	1 (4 bytes)	Number of subframes (actual for compound mode)
<i>SubBeamsNumber</i>	int32	1 (4 bytes)	The number of beams that are used to calculate final beam. This structure

			member is used when several beams are captured from the same position with different scanning set. For example - in multibeam scanning mode, when several beams with different focus settings are captured from the same position.
<i>SubFrameIndex</i>	int32	1 (4 bytes)	Index of subframe actual for compound mode (index of the first recorded channel RF data frame). For example, if compound frames number is equal to 5 and subframe index is equal to 3 that means that sequence of recorded subframes starts from the 4 th frame.
<i>number_of_frames_to_record</i>	int32	1 (4 bytes)	Number of channel RF data frames recorded into the file.
<i>aperture_size</i>	int32	1 (4 bytes)	Maximal possible number of active channels (64 for ArtUs system).
<i>mask_of_active_channels</i>	int32	aperture_size × 4 byte (256 bytes for ArtUs)	Mask (array) of zeros and ones to indicate which aperture channel is active.
<i>index_and_position_of_aperture_and_dummy_channels</i>	int32	BeamsPerFrame × 5 × 4 bytes	BeamsPerFrame × 5 array containing information for each beam: • aperture index (0 – BeamsPerFrame-1), • aperture_pos_x (in μm) respective to centre of the probe, • aperture_pos_y (in μm) respective to centre of the probe, • angle_aperture (in radians multiplied by 1000000) respective to centre of the probe, • dummy channels (if the channels are outside the aperture, what is actual for beams closer to probe edge, the number is negative and number of channels are declared, for example if value is equal to -32, that means that for first 32 channels of the beam there is no channel RF Data). The information for each beam are arranged in the following order (aperture index₀, aperture_pos_x₀, aperture_pos_y₀, angle_aperture₀, dummy channels₀, aperture index₁, aperture_pos_x₁, aperture_pos_y₁, angle_aperture₁, dummy channels₁, ...).
<i>position_of_channels</i>	int32	ChannelsPerBeam × 4 × 4 bytes	Each channel index, position and angle information (ChannelsPerBeam × 4 array): • chanel_idx (0 – ChannelsPerBeam – 1), • channel_pos_x (in μm) respective to centre of aperture, • channel_pos_y (in μm) respective to centre of aperture, • angle_elements (in radians multiplied by 1000000) respective to centre of aperture. The information for each channel is arranged in the following order (chanel_idx₀, channel_pos_x₀, channel_pos_y₀, angle_elements₀, chanel_idx₁, channel_pos_x₁, channel_pos_y₁, angle_elements₁, ...).

<i>start_end_indices_channels</i>	int32	ChannelsPerBeam×2× BeamsPerFrame×SubBeamsNumber× SubFramesNumber× 4 bytes	Array contains excitation pulse beginning and end indices for each channel which were active in transmission of ultrasound. Array contains information: - Start indexes, - End indexes.
<i>Start_Depth</i> (value exists only if not custom transmit delays are selected)	int32	1 (4 bytes)	Start depth index in samples for the RF data window.
<i>beam_position_and_orientation</i> (array exists only if not custom transmit delays are selected)	int32	BeamsPerFrame × SubFramesNumber × 3× 4 bytes	Information of position and orientation for each ultrasound beam: - beam_position_x (in μm) respective to centre of probe, - beam_position_y (in μm), respective to centre of probe, - beam_angle (in radians multiplied by 1000000), respective to centre of probe. The information for each channel is arranged in the following order (beam_position_x ₀ , beam_position_y ₀ , beam_angle ₀ , beam_position_x ₁ , beam_position_y ₁ , beam_angle ₁ , ...).
<i>channel_data</i>	int16	number_of_frames_to_record × FrameSize/2 × 2 bytes	Channel RF Data of N recorded frames. The data in the buffer are arranged as follows: <i>SamplesPerChannel</i> × <i>ChannelsPerBeam</i> × <i>BeamsPerFrame</i> × <i>number_of_frames_to_record</i> .

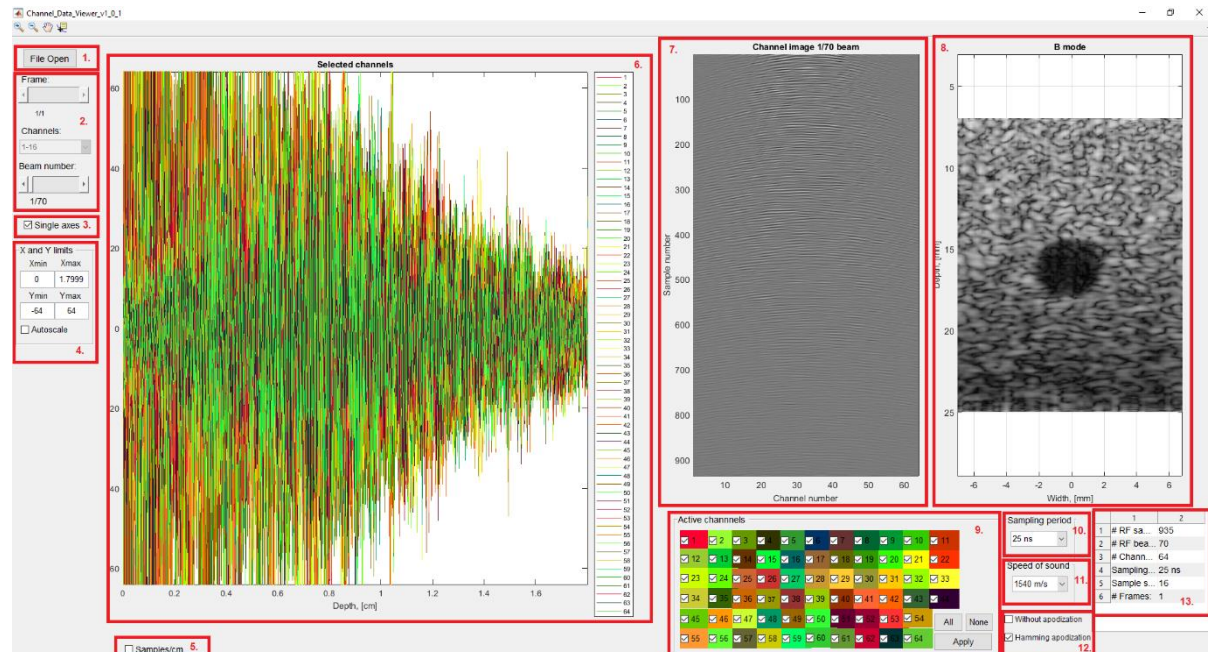
2. MATLAB tools installation notes

The software package contains a standalone compiled version of the Channel Data Viewer tool graphical user interface ([Channel_Data_Viewer_v1_0_1.exe](#)) and a collection of functions (M-files) including the main file of the GUI ([Channel_Data_Viewer_v1_0_1.m](#)). Users can run GUI even without full MATLAB installation. It is enough to set up standalone MATLAB Runtime (mandatory shared libraries that allow execution of compiled programs). MATLAB compiler runtime could be downloaded from the website and installed. Please install version for 64-bit Windows: <https://se.mathworks.com/products/compiler/matlab-runtime.html>. The GUIs were compiled by using MATLAB R2020a and version 9.8 of Runtime is necessary. **Please note:** the runtime allows to use of GUI only, but aiming to use the scripts (`...lscrip`ts), modify and build your applications full version of MATLAB should be installed.

3. Description and structure of Channel Data Viewer graphical user interface

This section presents the MATLAB Channel Data Viewer tool graphical user interface main window and the available controls. GUI allows to import of data files recorded by sample *ArtUs RF Data Control II (from the 2.0.0 version)*.

3.1 Main window and controls



Num	Component	Description
1	File Open button	The button calls the <code>importCH_RFData2MATLAB.m</code> function and the result returns recorded Channel Data and header which contains mandatory information for B image reconstruction (such as the mask of active channels, channels position, and apertures position, etc.)
2	Channel data selection controls	There are three controls to navigate through the recorded Channel Data: Frames slider to navigate through frames, popup-menu Channels is dedicated to select which sixteen channels will be plotted (1 – 16, 17 – 32, 33 – 48, 49 – 64), the menu is enabled only if 3 . The Single axes checkbox is unchecked, the Beam number slider allows to selection of the scanned beam, and plots of channel data are updated accordingly.
3	Channel data plotting option	Checkbox Single axes allow to select Channel Data plotting mode: if checked all the channels of a single beam are displayed in single axes (as shown in GUI screenshot), if unchecked plots the selected 16 channels (popup-menu Channels allows to select which 16).
4	Scaling options	X and Y limits will help you to adjust the scales of Channel Data plots that are displayed (just enter the desired values into 4 Edit Boxes). Also, there is an option to select Autoscale mode by checking the checkbox .
5	Axes type option (Samples/cm)	Channel data X axis could be represented in samples or in metric units, the checkbox if checked turns into samples, otherwise in cm.

6	Channel data plotting window	Chanel data are plotted in the window, there are two plotting options (single axes or 16 channels separately)
7	Channel image of single beam window	The imaging window shows a Channel data image of a single selected beam.
8	Beamformed B mode imaging window	The imaging window represents an example of a B mode image obtained by applying Delay-and-sum (DAS) beamforming. Note the image quality is lower in comparison to that obtained by TELEMED scanners in clinical mode due to the lack of post-processing procedures and beamforming peculiarities. Note! If channel RF data were recorded by using custom delays or arbitrary beams the beamforming will not be performed axes will be empty.
9	Active channels selection controls	64 checkboxes allow to selection of channels that will be used in the B mode image beamforming procedure. Note if some channels were unchecked during data recording using ArtUs RF Data Control interface, the corresponding checkboxes will be disabled. The optional channel selection allows us to evaluate how the focusing quality is affected by different combinations of channels. Buttons All and None allow the user to (Un)check all checkboxes. After the desired channels are selected button Apply must be pressed.
10	Sampling period selection (decimation)	Popup-menu Sampling period in principle allows for the decimation of the data from maximal sampling 25 ns period (x2 (50 ns), x4 (100 ns), x8 (200 ns) times) and to evaluate how the reconstructed image quality suffers then.
11	Speed of sound selection	Popup-menu Speed of sound (SoS) allows us to evaluate how the beamformed image looks by using different SoS constants (options: 1440 m/s, 1540 m/s, 1640 m/s).
12	Optional apodization	Apodization Checkboxes allow the user to evaluate how the reconstructed B mode image looks then Hamming window-based apodization is applied, and without applying weighting coefficients to channel data.
13	Acquired Channel Data parameters	The table lists the main acquired Channel Data parameters: • RF samples number – number of samples for each channel • RF beams number – number of scanned beams • Channels number – number of active channels • Sampling period (in ns) • Sample size (data bits) • Frames – number of recorded channel data frames

4. GUI functions and options

4.1 MAIN GUI function (**Channel_Data_Viewer.m**)

The main GUI function calls 7 external functions. The functions allow performing basic B mode imaging-related signal processing and data visualization tasks and could be used in the development of new projects. The list of the functions is provided below and the functions (inputs and outputs) are described explicitly in the following subsections:

No.	Function	Description
1	<code>importCH_RFData2MATLAB.m</code>	The function allows to import of the recorded Channel data and main acquisition parameters needed for imaging into MATLAB.
2	<code>DAS_beamforming.m</code>	Function to calculate the delays for delay-and-sum beamforming and to make a summation of delayed signals. The function calculates the spatial position for each beamformed RF data point.
3	<code>show_plots_of_channel_data.m</code>	The function is used for plotting channel data in separate or single axes (GUI window 6).
4	<code>visualization.m</code>	This function is used for 2D imaging of single beam channel data or reconstructed B mode images in axes.
5	<code>band_pass_filter.m</code>	The function could be used for zero-order digital filtering of received RF signals before envelope detection and scan conversion.
6	<code>envelope_2_image.m</code>	Function allows to convert envelope data of RF signal to log-compressed B mode image.
7	<code>rotation_matrix.m</code>	Function rotates scan line coordinates vector around the origin of the coordinate system (0,0) in a counter-clockwise direction by a predefined angle – θ .

4.2 Import channel data to MATLAB (**importCH_RFData2MATLAB.m**)

The function allows importing recorded Channel data and main acquisition parameters needed for image reconstruction and imaging into MATLAB.

Inputs of the function:

FILENAME – *.bin binary file recorded using ArtUs device and Artus RF Data Control II software,
DIR – directory of the file.

Outputs of the function:

HEADER – header containing information needed for B mode image reconstruction from recorded channel data and some data acquisition parameters,
HEADER.flag – flag which is equal to 1 if tag scanned beams position and angle information exists (exists only not for custom delays),
HEADER.sampling_period – channel data sampling period in [ns],
HEADER.BeamsPerFrame – number of scanned beams per frame,
HEADER.SamplesPerChannel – number of samples for each channel,
HEADER.ChannelsPerBeam – number of active channels per beam,
HEADER.SampleBitCount – data bits,
HEADER.FrameSize – size of single channel data frame (in bytes),
HEADER.SubFramesNumber – number of subframes (not 0 for B Compound mode),

HEADER.SubBeamsNumber – the number of beams that are used to calculate final beam. This structure member is used when several beams are captured from the same position with different scanning set. For example - in multibeam scanning mode, when several beams with different focus settings are captured from the same position,

HEADER.SubFrameIndex – subframe index of first recorded subframe (For example, if compound frames number is equal to 5 and subframe index is equal to 3 that means that sequence of recorded subframes starts from the 4th frame),

HEADER.number_of_frames_to_record – number of channel data frames in a recorded file,

HEADER.aperture_size – maximal possible channel number (64 for ArtUs system),

HEADER.mask_of_active_channels – mask of zeros and ones indicating which channels are active,

HEADER.index_of_aperture – aperture index (0 – *BeamsPerFrame*-1),

HEADER.aperture_pos_x – x coordinate of each aperture position relative to probes centre (in μm),

HEADER.aperture_pos_y – y coordinate of each aperture position relative to probes centre (in μm),

HEADER.angle_aperture – the angle of aperture in radians

HEADER.dummy_ch – if the channels are outside the aperture, what is actual for beams closer to probe edge, the number is negative and the number of channels is declared, for example, if value is equal to -32, that means that for the first 32 channels of the beam, there is no channel RF Data,

HEADER.chanel_idx – number of each channel in aperture *chanel_idx* (0 – *ChannelsPerBeam* - 1),

HEADER.channel_pos_x – x coordinate of each channel position relative to probes centre (in μm),

HEADER.channel_pos_y – y coordinate of each channel position relative to probes centre (in μm),

HEADER.angle_elements – angle of each channel in radians,

HEADER.start_indices – array contains excitation pulse beginning indices for each channel which were active in transmission of ultrasound,

HEADER.end_indices – array contains excitation pulse end indices for each channel which were active in transmission of ultrasound,

HEADER.Start_Depth – Start depth index in samples for the RF data window (flag = 1),

HEADER.beam_position_x – x coordinate of each beam position relative to probes centre (in μm), array exists only if not custom transmit delays are selected (flag = 1),

HEADER.beam_position_y – y coordinate of each beam position relative to probes centre (in μm), array exists only if not custom transmit delays are selected (flag = 1),

HEADER.beam_angle – angle of each beam relative to probes centre (in radians), array exists only if not custom transmit delays are selected (flag = 1),

CH_DATA – Channel RF Data of N recorded frames. The data in the buffer are arranged as follows: *SamplesPerChannel* × *ChannelsPerBeam* × *BeamsPerFrame* × *number_of_frames_to_record*.

4.3 DAS Beamforming function (**DAS_beamforming.m**)

The function is used for classical delay-and-sum beamforming. In the first stage, the delays are calculated, afterward the channel signals are delayed and summed to obtain beamformed RF data, and finally, the spatial positions of the RF data points are calculated for imaging.

Inputs of the function:

HEADER – Channel Data header containing information mandatory for image reconstruction,

channel_RF – Channel Data Matrix,

speed_of_sound – ultrasound velocity used for the beamforming (delays calculation, in m/s),

apod – mask of apodization coefficients (could be all ones for the case without applying weights), a vector of 1 X *K* (where *K* – number of channels),

vecor_of_selected_channels – mask of 0 and 1 indicating which channels are active (1) and participate in the beamforming and which are not (0),

discretization_period – sampling period in ns used for delays calculation.

Outputs of the function:

beamformed_RF – beamformed RF data (*N* X *M*, where *N* – number of scanned beams, and *M* – number of samples per beam),

x_axis, *y_axis* – spatial coordinates for beamformed RF Data matrix for imaging.

4.4 Channel data plotting function ([show_plots_of_channel_data.m](#))

The function displays channel RF data optionally in separate axes or in overlaid in single axes.

Inputs of the function:

handles – all the objects and data stored in GUI handles.

Outputs of the function:

h - handle to a graph legend.

4.5 Imaging of B mode and single channel image ([visualization.m](#))

Function displays channel image or beamformed B mode image.

Inputs of the function:

AX – axis handle in which displays pcolor plot,

x_axis – X coordinate matrix of the image to display,

y_axis – Y coordinate matrix of the image to display,

beamformed_RF – beamformed RF data $N \times M$ (or channel RF data of single beam for channel image)

angle – if angle parameter is equal to zero it is assumed that the probe was linear, if not equal to zero and greater than convex, if it is negative than the channel image of a single beam is displayed,

beam_number – the number of beam to display for channel image,

total_number_of_beams – number of beams in an image,

fontsize_proportion – coefficient resizes the font of labels proportionally when resizing the GUI window,

discretization_period – actual sampling period in ns.

Outputs of the function:

displayed_pcolor_plot (single beam channel or beamformed B mode image)

4.6 Band-pass filters ([band_pass_filter.m](#))

The function could be used for zero-order digital filtering (forward and reverse directions) of beamformed RF signals before envelope detection and scan conversion. The cut-off frequencies (upper *fH* and lower *fL* frequency of pass-band) are user-predefined. There are 3 filtering options: (a) using infinite impulse response Butterworth type filter, (b) finite impulse response filter design using window method, and 3rd (c) “None” turns off the filter. By default, taps of the filter (*n*) are set to a) *n*=9, b) *n*=200 (reduced to 100 if number of samples is lower than 3*n*).

Inputs of the function:

in – received single RF frame ($N \times M$), where *N* – number of samples, *M* – number of scanning lines (rows),

low_freq_cutoff – lower cut-off frequency of the pass-band,

up_freq_cutoff – upper cut-off frequency of the pass-band,

Sampling_frequency – sampling frequency,

flag – sets the filtering method 0 for infinite impulse response (Butterworth type), 1 - for finite impulse response (FIR filter designed using the window method), 2 - turns of the filters.

Output of the function:

out – filtered signals of a single RF frame.

Figure below shows examples of frequency responses of IIR and FIR filters with $f_L=0.5$ MHz and $f_H=3$ MHz.

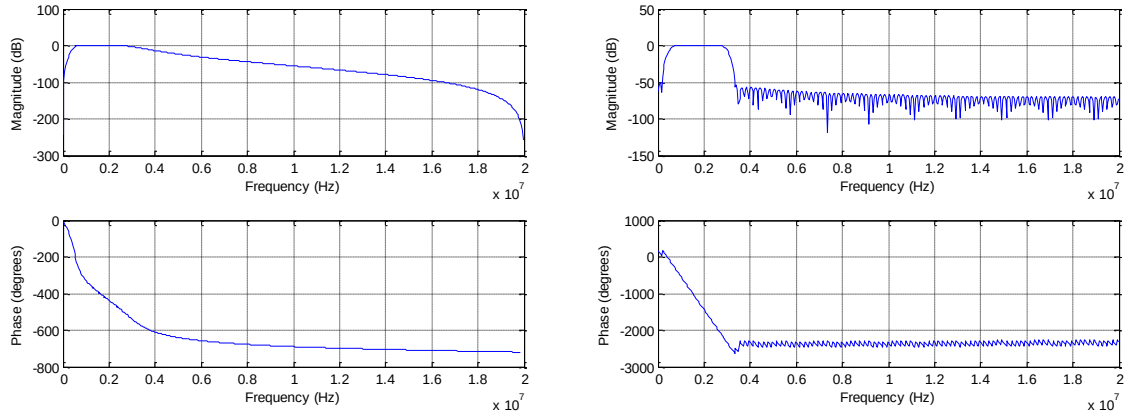


Fig. Frequency responses of the infinite impulse response filter (left) and finite impulse response filter (right). Cut-off frequencies for both: lower cut-off frequency $f_L = 0.5$ MHz, upper cut-off frequency $f_H=3$ MHz.

4.7 Logarithmic compression (**envelope_2_image.m**)

Function allows to adjust dynamic range of displayed image through log-compression of detected envelope of RF signal. The displayed image could be expressed as follows:

$$I = e^{\alpha \cdot \ln(\text{env}(t))} / \beta, \quad (4)$$

where α – coefficient of non-linearity (set to $\alpha=0.47$), env – detected envelope of signal, $\beta = \alpha \cdot \ln(2^{BIT} - 1) / k$, k – number of gray levels of output image, $\ln$ – natural logarithm.

Input of the function:

alpha – coefficient of non-linearity,
num_gray_levels – number of gray levels of output image (i.e. 256 for 8-bits),
BIT – sample size of RF data in bits,
envelope – detected envelope of RF signal.

Output of the function:

image_8_BIT – logarithmically compressed image.

4.8 Rotation matrix (**rotation_matrix.m**)

Function rotates scan line coordinates vector around origin of coordinate system (0,0) in counter-clockwise direction by a predefined angle – θ . Rotation matrix is defined:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (6)$$

where θ – rotation angle.

Inputs of the function:

vector_in – input vector (x, y) coordinates,
theta – rotation angle.

Outputs of the function:

out – rotated vector (x, y) coordinates.

5. Revision history

Version	Date	Description of Revision	Revision author
1.0.0	11/09/2022	Initial Release	A. Sakalauskas
1.0.1	12/06/2024	Minor grammar editing	A. Sakalauskas

6. References

[1] ArtUs RF Data Control User II Manual.

(c)1992-2024 TELEMED, UAB

Company address: Savanoriu pr. 178A, Vilnius, LT-03154, Lithuania

Internet: www.pcultrasound.com, www.telemed.lt

Information, sales e-mail: info@pcultrasound.com, info@telemed.lt

Technical Support e-mail: support@pcultrasound.com, support@telemed.lt