# BRAIN 1.4

# Software Manual

## Installation

### Matlab version

Unzip the files to a suitable folder, say Brain\_1\_4\_matlab (the name of this folder does not matter). Then from within Matlab either:

1. Navigate to this folder and type brain<enter> to start the programme, or
2. Add this folder to the Matlab path and then type brain<enter> to start the program.

Whenever BRAIN is run, a folder, Brain (local data), will be created in the folder from which BRAIN was run if that folder does not already exist. If method 1 above is used, then Brain (local data) will just be a subfolder within Brain\_1\_4\_matlab. However, if method 2 is used the Brain (local data) can be anywhere (and there can be multiple copies). Both Brain\_1\_4\_matlab and Brain (local data) contain subfolders Arrays, Materials, Imaging, Analysis and Instruments. When BRAIN is run, the data and functions in Brain\_1\_4\_matlab and Brain (local data) are both available to the user. The reason for this structure is to allow multiple users to use the same master copy of BRAIN but keep their own data and code separately.

### Standalone version

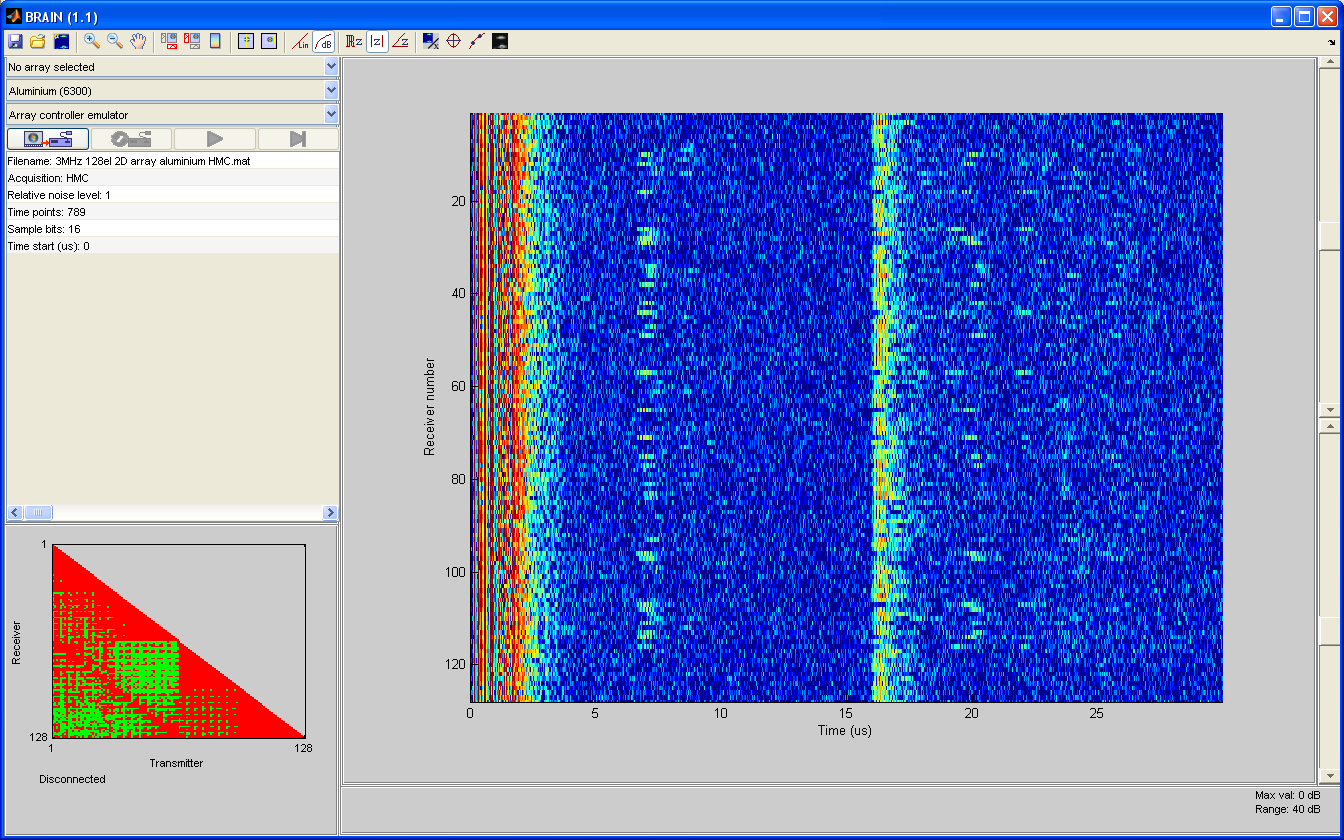
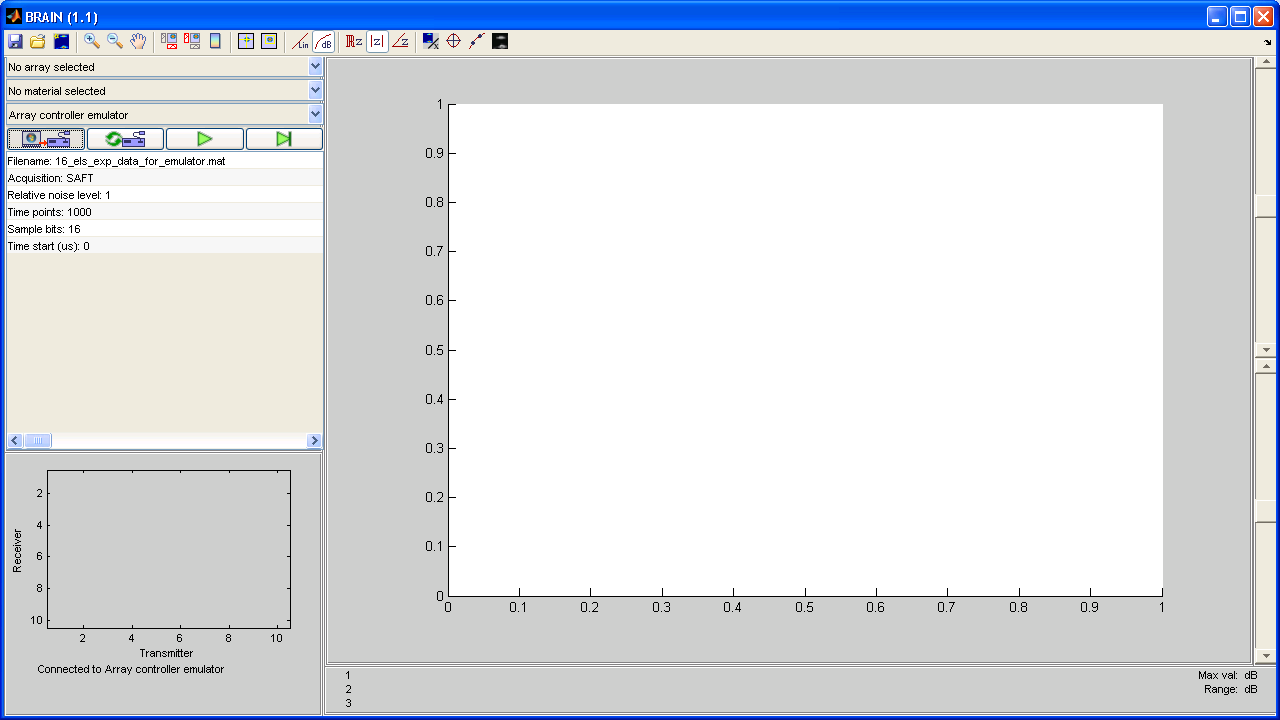
Unzip the files into a suitable folder, say Brain\_1\_4\_standalone (the name of this folder does not matter). If not already installed, download and install (administrator privileges required) the Matlab runtime library from the Mathworks website:

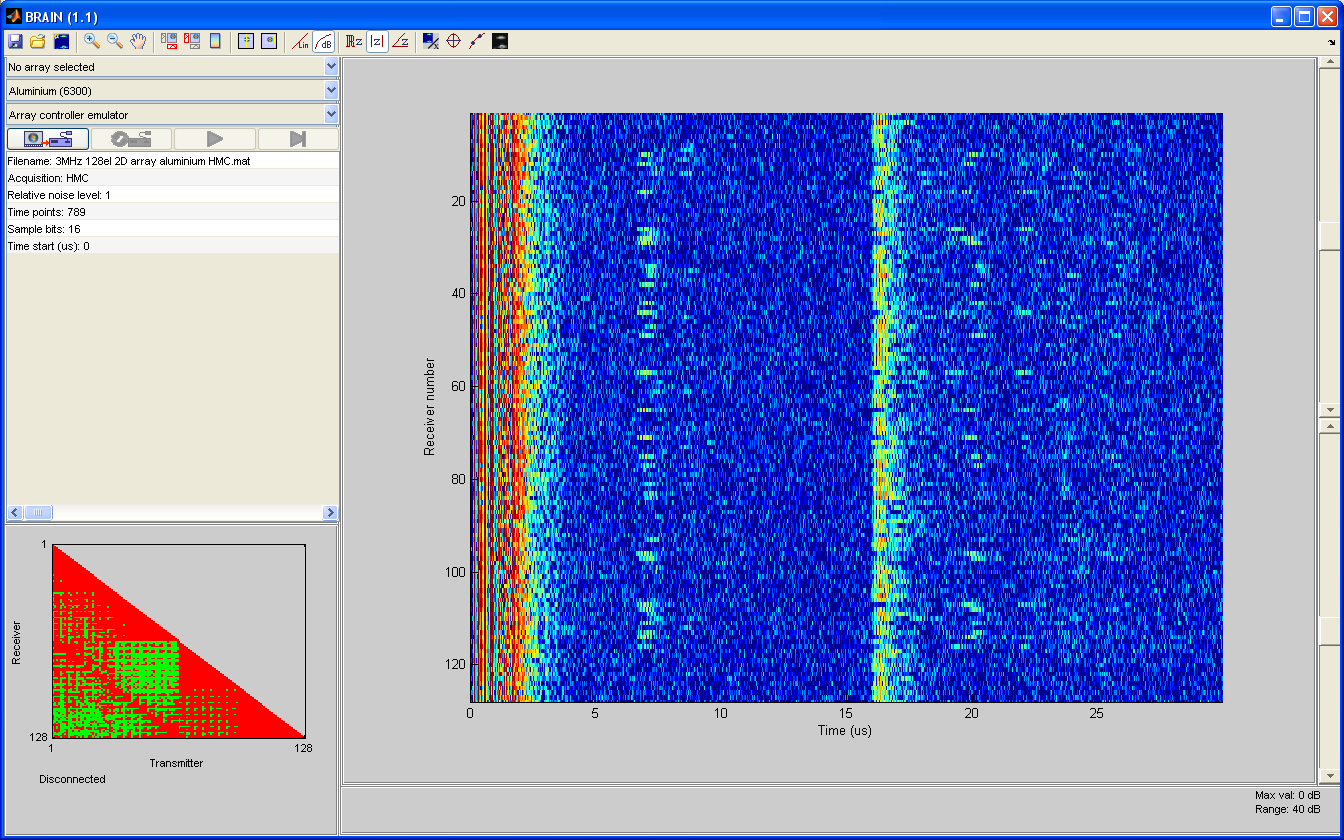
<[www.mathworks.co.uk/products/compiler/mcr/index.html](https://www.bris.ac.uk/mecheng/research/solids/ndt/secure/%22http:/www.mathworks.co.uk/products/compiler/mcr/index.html%22)>

Double click the brain.exe file in folder Brain\_1\_4\_standalone and BRAIN will start. The first time BRAIN is executed there will be a minute or so delay while other files are uncompressed and some new folders are created within the Brain\_1\_4\_standalone folder.

## Quickstart

Launch BRAIN using the appropriate method. The main data acquisition window will appear. If a suitable GPU and driver software is available then the title of the window will BRAIN 1.4 (GPU) otherwise it will be BRAIN 1.4 (no GPU).

Ensure that Array controller emulator is selected in the pull-down list and then click the left-most connect button () in the row of four large buttons below the list. At the bottom left of the window it will say Connected to Array controller emulator and the other three large buttons will become enabled. Click the third large button from the left, play () and the data acquisition process should start, with frames of raw data being displayed in the main panel of the window. The default emulator data file is from a contact inspection using a 5 MHz, 64 element array on a 50 mm thick aluminium sample with a number of artificial defects at a depth of 20 mm. Other files can be selected and changes made to the emulator operation by clicking the appropriate row in the table below the buttons and entering a new value.

To launch a new imaging window, click the third button on the top tool bar (New process, ) and a list of available imaging algorithms will appear. Select Contact TFM from the list and click OK. A new window will appear showing a TFM image created from the simulated data. Imaging parameters can be changed by clicking on the appropriate line in the table to the left of the image and entering a new value.

## Operation

### Main data acquisition window

#### Overview

The main BRAIN window is concerned with the acquisition and display of raw data from an array and an example is shown in Fig. 1. At the top of the window is a toolbar with buttons that provide access to most of the available functions. Below, the window can be divided into two parts. The left third is for operation of the array controller instrument, while the other two-thirds provide a display of the raw array data itself.

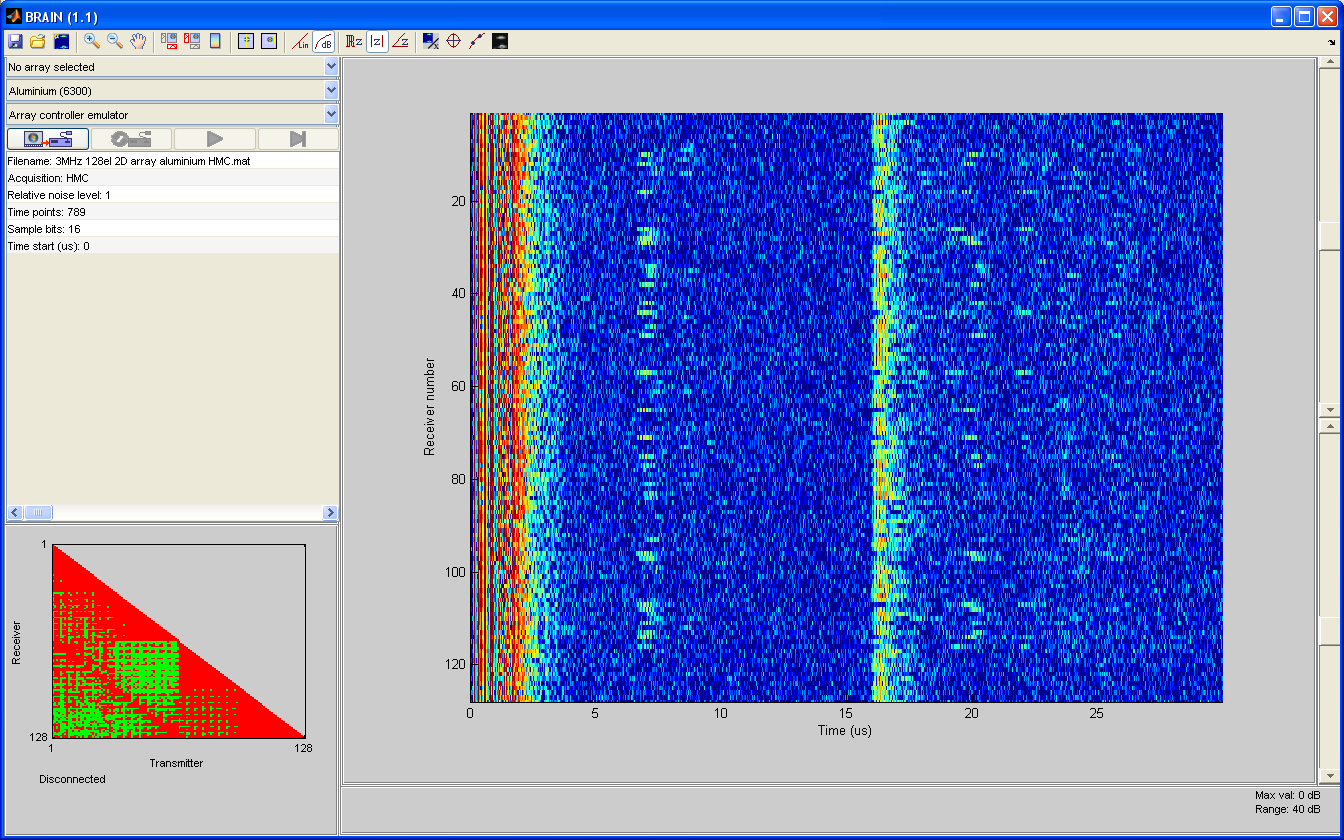
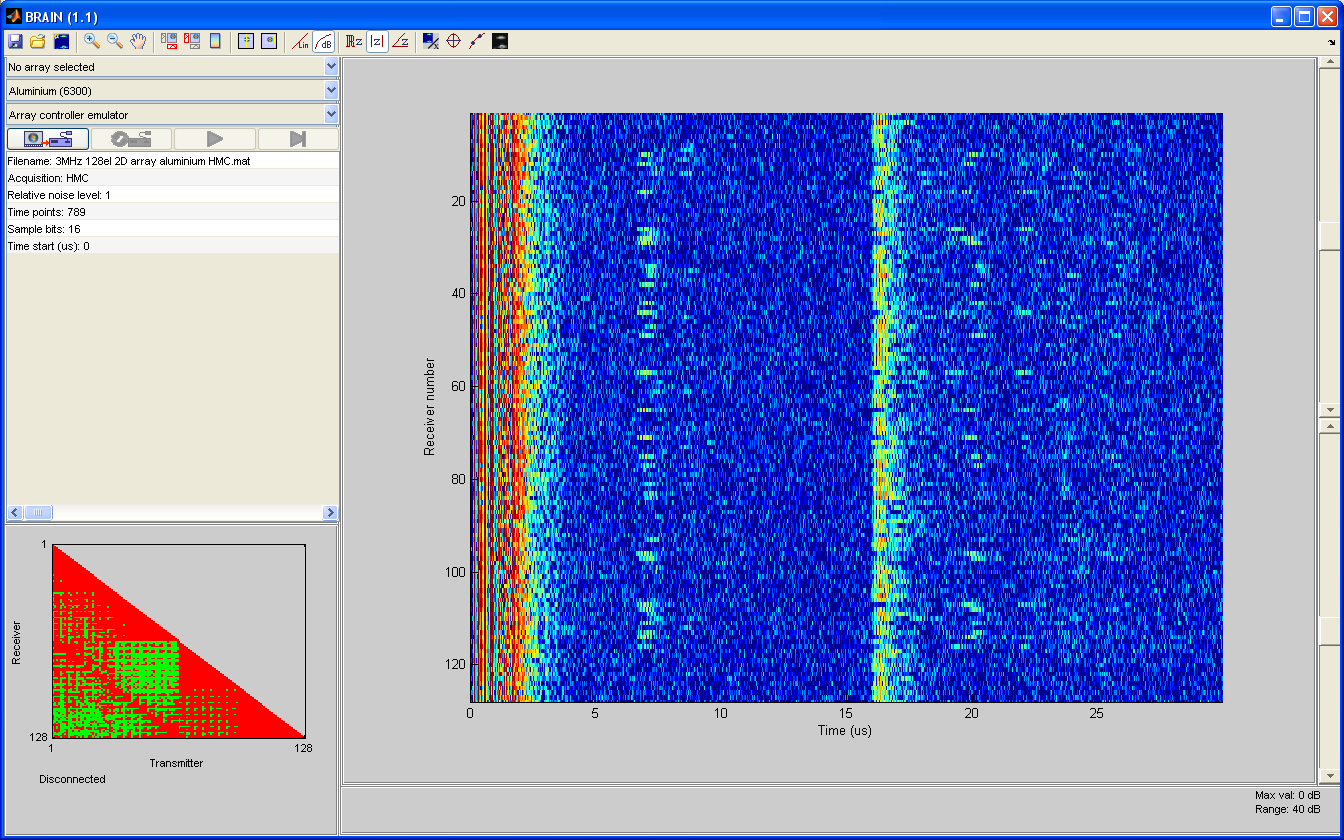
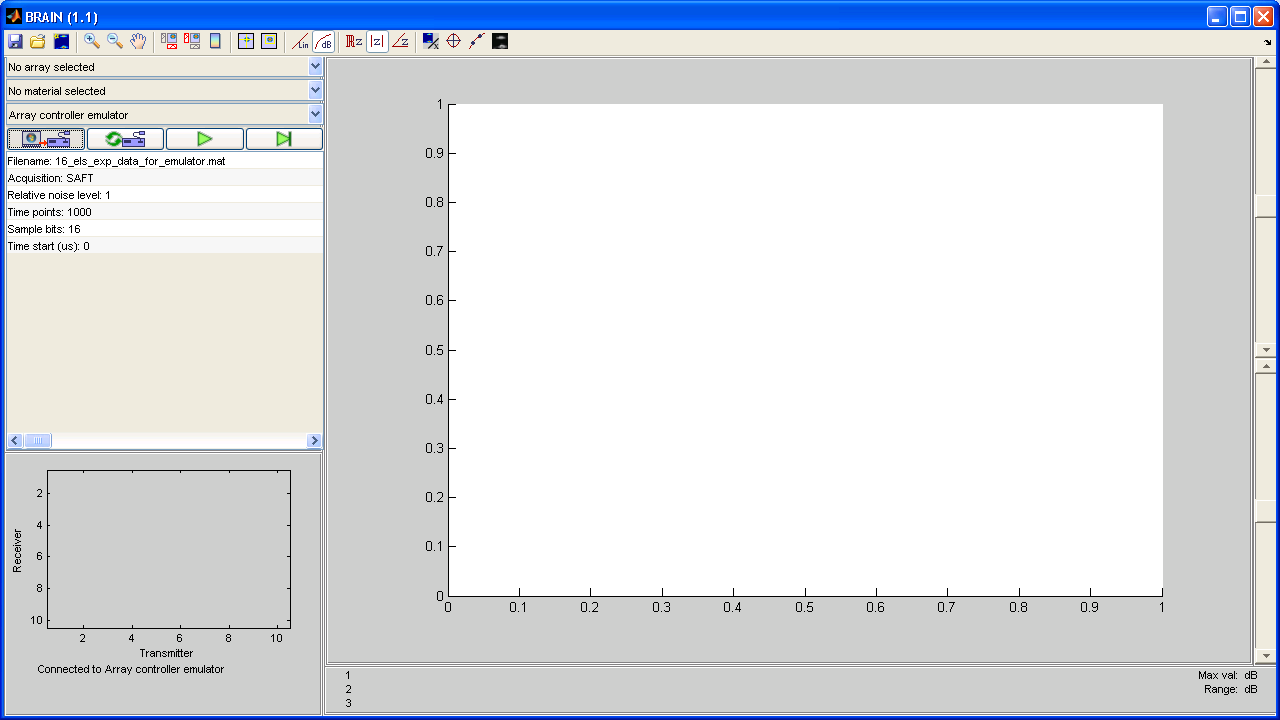
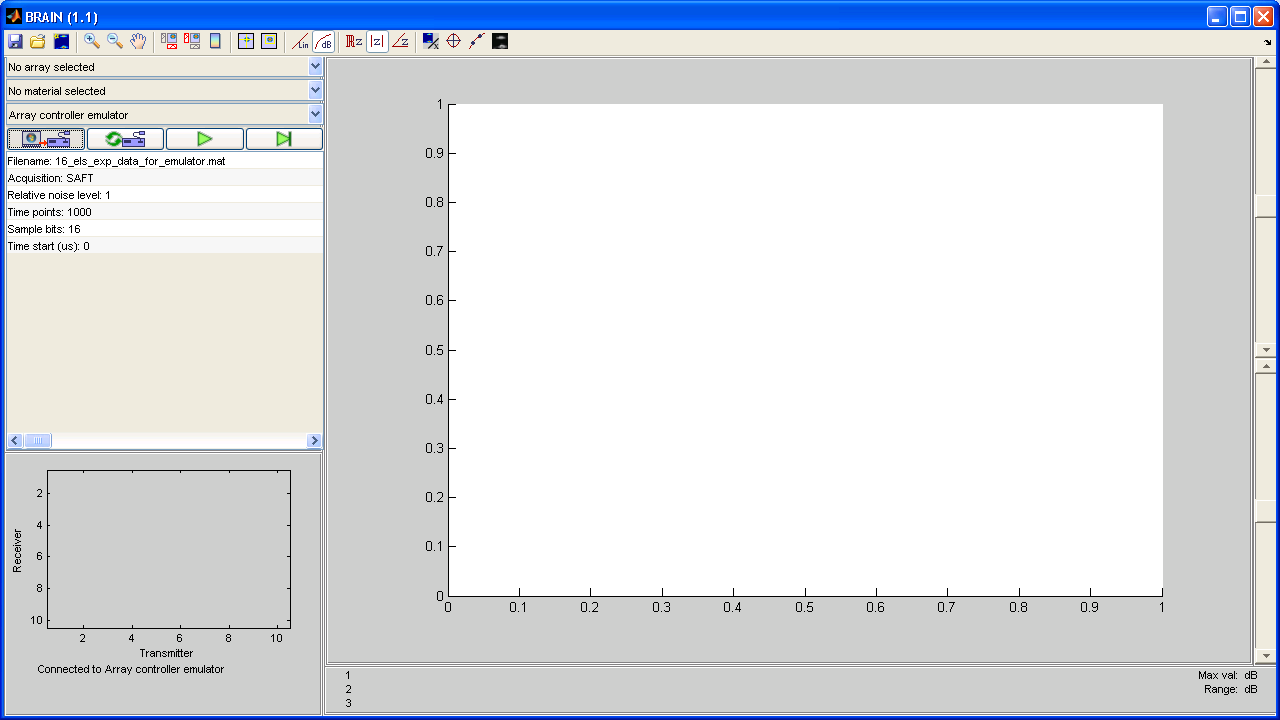
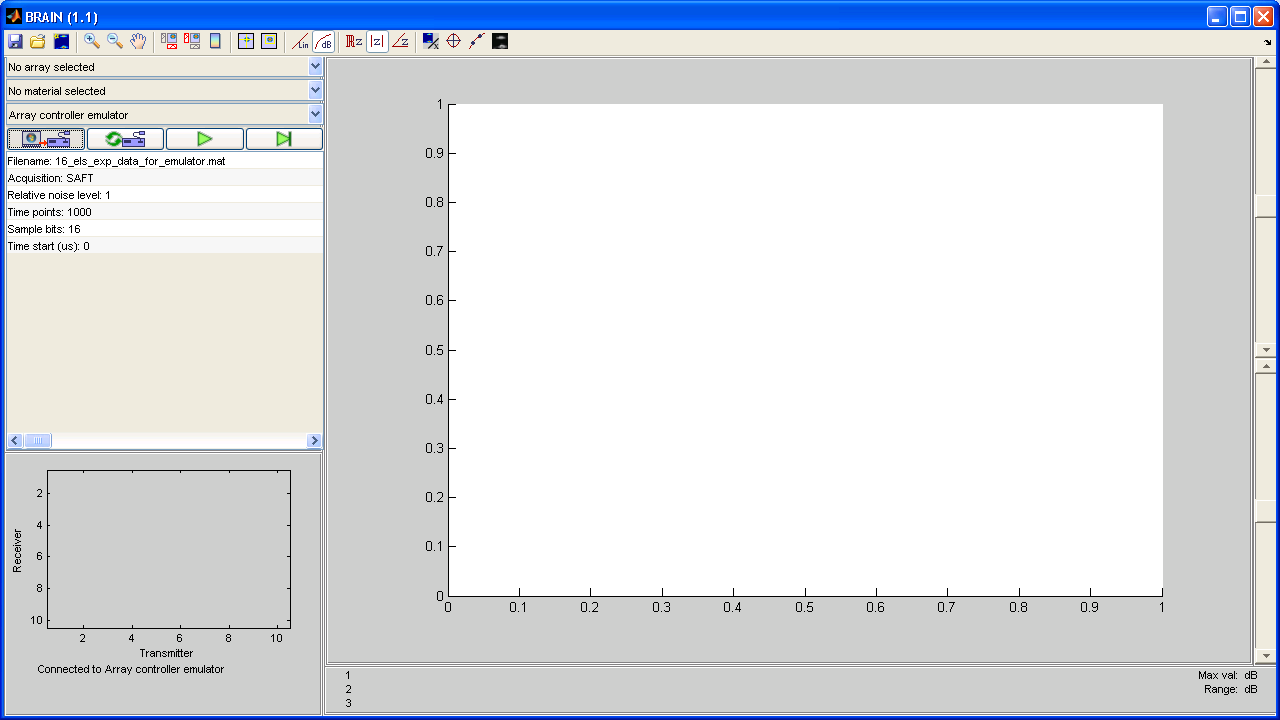


Fig. 1 Main BRAIN window for raw array data acquisition and display.

#### Array controller operation

The detail of this part of the main window is shown in Fig. 2. To connect to an array controller, select the desired device from the pull-down list and click on the connect button, . If the connection is successful, the text at the bottom of the window in Fig. 2(b) will change to Connected to *array controller name*, the table below this button will update to reflect the available parameters for the chosen array controller and the other three buttons in the row will be enabled. Alter the parameters in the table as required (see Section 3.1 for details of how to change parameters in tables and Section 4 for details of the parameters for specific instruments) and then press either play button, , or the play once button, . If play is selected, the icon on the button changes to a red stop sign and successive frames of data are captured and processed continuously until the button is pressed again. If play once is pressed, one frame of data is captured and processed. The reset button, , stops data collection if it is running and sends a reset command to the array controller.

When a frame of data is acquired, the graphic at the bottom left of the main window shown in Fig. 2(b) gives a summary of the recorded time-traces in the full matrix data. Coloured (rather than grey) values indicate that the corresponding time-trace has been recorded and the colour indicates the peak amplitude according to Table 1.

|  |  |
| --- | --- |
| **Colour** | **Meaning** |
| Red | >80% full scale |
| Yellow | <20% full scale |
| Green | >20% and <80% full scale |

Table 1 Raw data status colour meanings.

If the data cursors are active in the raw-data display pane, then the peak amplitude is based only on the part of the time-traces later than the current time-curser position. In general, the gain of the array controller should be adjusted so that no part of any signal in the time range of interest is red to avoid potential clipping.

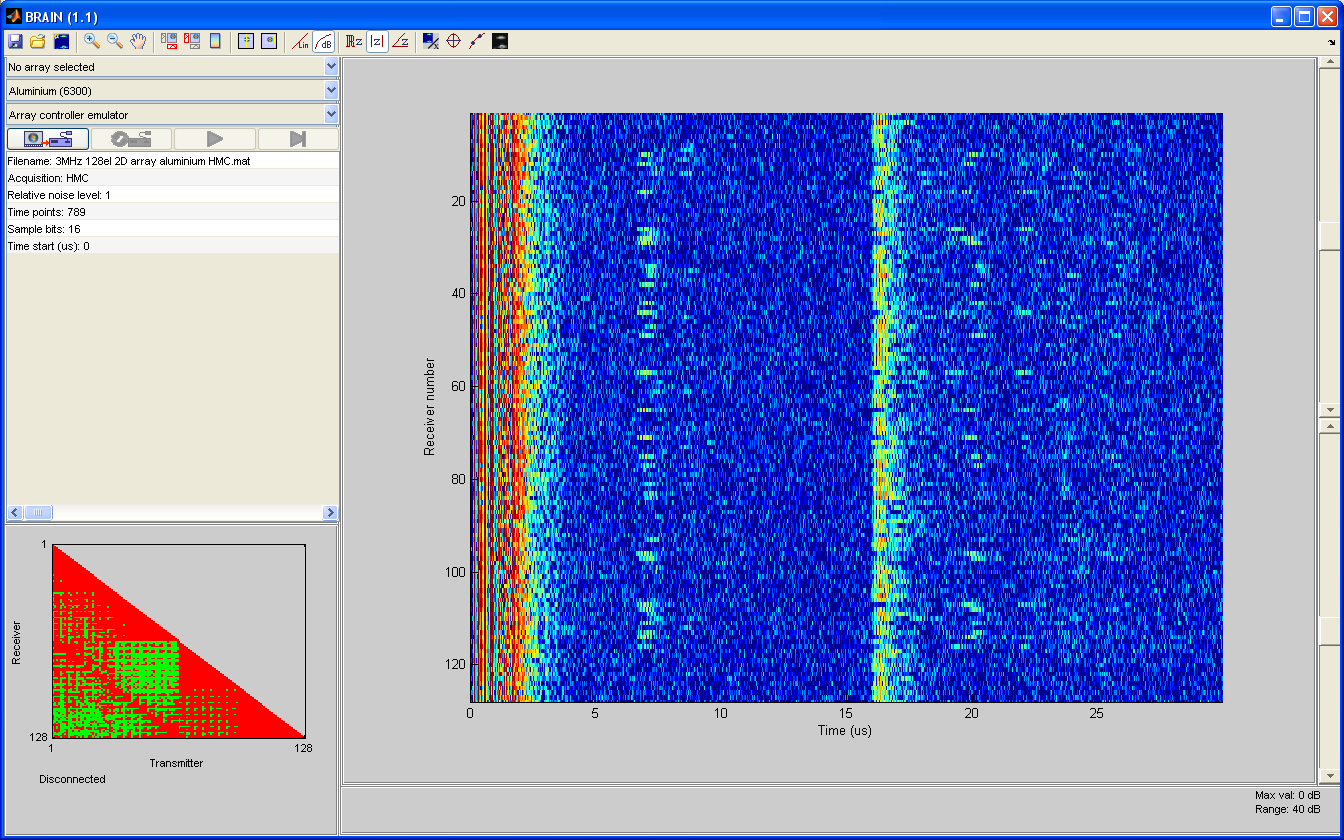
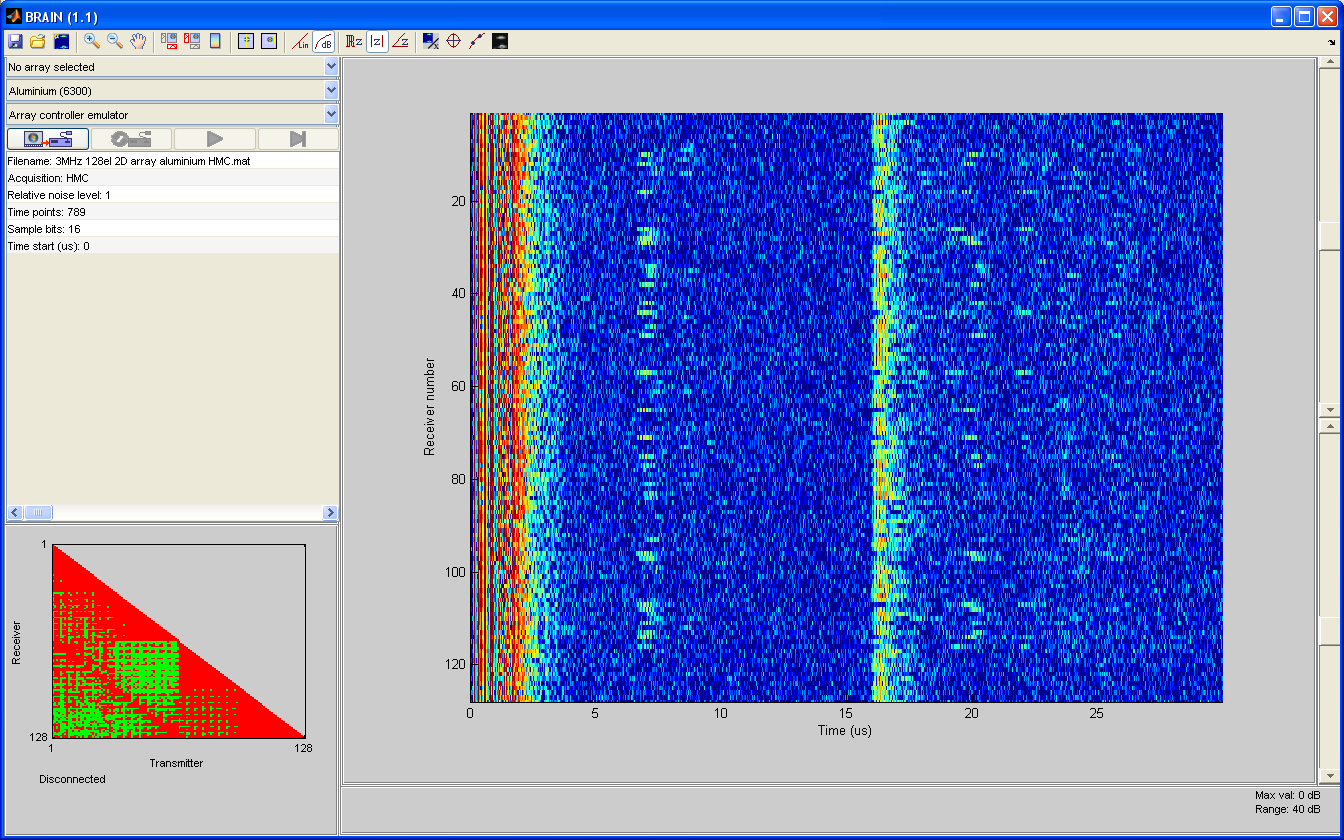
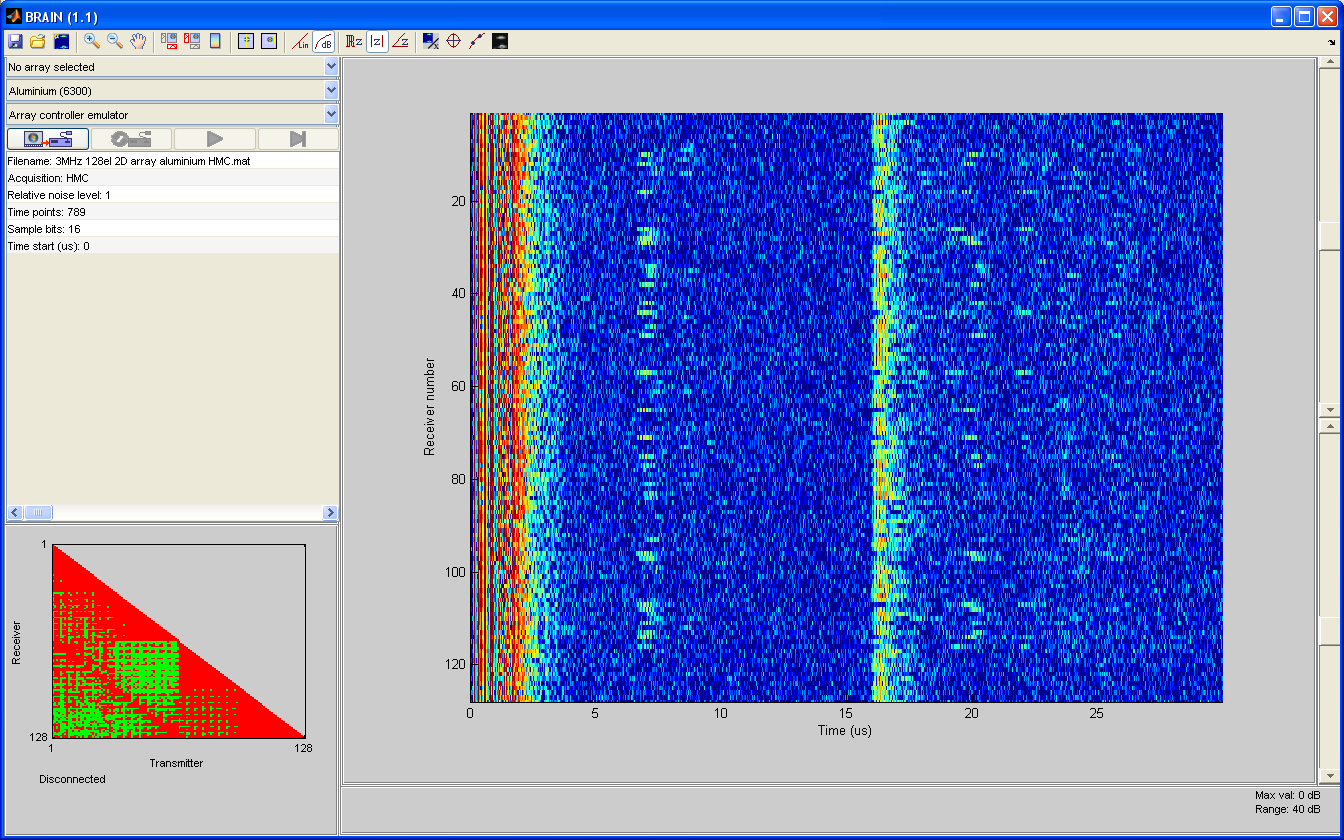
(a) (b)

Fig. 2 Array controller part of main window: (a) controls, (b) status.

#### Raw data display

For clarity, only *n* received signals are displayed, even though the full matrix of array data may contain up to *n*2 signals. The raw data display pane in the main window uses the same plotting routine as that for the display of 2D images from an array and the operation of both is described in Section 3.2.

## Imaging

To launch a new imaging window, press the New process button, , on the toolbar. This will bring up a window with a list of the available processing functions. Click on the appropriate one to launch a new window showing the resulting image. The left side of the window contains a table of the parameters associated with the imaging algorithm (see Section 4.1 for how to alter these) and the right hand side contains the image. Currently, there is no facility to export either the image or the underlying data other than by using the Windows Alt+PrtScrn method to copy a bitmap of the current window to the clipboard. A future version of BRAIN will provide complete image export facilities.

The image will automatically update whenever a new frame of data is acquired and the rate of data acquisition will be slowed according to how long the image takes to be formed. In general, the smaller the size of the raw data and the fewer the pixels in the image, the faster the imaging will be. Therefore, SAFT and CSM data can be used to produce images with a large number of pixels in almost real time, but to obtain images from HMC or FMC data at speed requires careful optimisation of both the acquisition and imaging parameters.

Note that changing either acquisition or image parameters does not automatically rescale the image. This is intentional.

The currently implemented images functions are described in the following sections. Both these are linear delay and sum algorithms and both use the same kernel function, which is now fairly optimised for a typical CPU on a PC. If a suitable GPU is detected, this will be used automatically unless disabled.

### Coordinates

All arrays are defined in a local Cartesian coordinate system , with the origin at the centre of the array. For a 1D array, the elements are distributed along the -axis (i.e. ), while for a 2D array, the elements lie in the plane.

For most imaging algorithms, the array coordinates are the same as those of the image, the current exception being oblique incidence TFM where the array is offset in the negative direction and rotated so that the sample surface remains in the plane at . For all imaging algorithms, is depth measured into the sample away from the inspection surface. 2D imaging with 1D arrays takes place in the plane.

### Filtering

All currently implemented algorithms have an optional digital filtering stage that is applied to the raw time-domain signals prior to processing. This is based on a filter with a Gaussian shape in the frequency domain defined by its centre frequency and fractional bandwidth (width at ‑40 dB divided by centre frequency) parameters. The centre frequency of the filter defaults to the centre frequency of the array and the fractional bandwidth to 200 %.

### Basic TFM

This implements the basic contact TFM algorithm to the data and adapts it according to whether the data is SAFT, CSM, HMC or FMC. It automatically presents the results as either a 2D image if a 1D array is used (see Section 4.2 for details on how to use this display) or in a special 3D display if a 2D array is detected. The algorithm includes an optional Gaussian frequency-domain pre-filtering stage specified by filter centre frequency and fractional bandwidth parameters. Also an angle-limiter can be activated which limits the aperture angle at all image points (i.e. makes the image constant *f*-number).

### Contact B-scan

This implements a simple unfocused contact B-scan algorithm to the data and adapts it according to whether the data is SAFT, CSM, HMC or FMC. It automatically presents the results as either a 2D image if a 1D array is used (see Section 4.2 for details on how to use this display) or in a special 3D display if a 2D array is detected. Interpolation into the raw data to form the image can either be linear (more accurate but slower) or rounded to the nearest point.

### Contact sector scan (1D arrays only – beta version)

This implements a simple unfocused sector scan algorithm to the data and adapts it according to whether the data is SAFT, CSM, HMC or FMC. The algorithm includes an optional Gaussian frequency domain pre-filtering stage specified by filter centre frequency and fractional bandwidth parameters. Interpolation into the raw data to form the image can either be linear (more accurate but slower) or rounded to the nearest point.

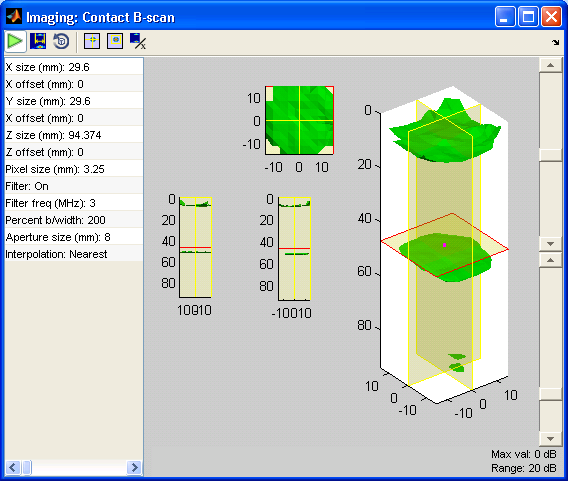
### Immersion TFM (1D arrays only – beta version)

This handles the case of a 1D array used in immersion above a sample with a surface that can be flat (aligned to the array), curved or arbitrary. For flat and curved surfaces, the user inputs the necessary geometric parameters to defined the surface (distance from the array for a flat surface; centre coordinates and radius of curvature for a curved surface). The TFM delay laws are then calculated accordingly to take into account refraction at the surface. In the case of an irregular surface and adaptive algorithm is used to first find the surface from the ultrasonic data and then to create the necessary TFM delay laws. The surface finding algorithm may require some parameter adjustment to obtain satisfactory results.

### Oblique incidence TFM (1D arrays only – beta version)

This is essentially a specialised case of the Immersion TFM algorithm for the case of an array inclined at an angle above a flat sample. As in the Immersion TFM algorithm, the code can either adaptively determine the location of the sample surface (useful for immersion inspections) or this can be specified (e.g. if array is mounted on a solid coupling wedge). In both cases, the surface of the sample is assumed parallel to the -axis and at depth and the array is moved and rotated accordingly. Either a longitudinal or shear wave velocity can be specified for the material under inspection as the algorithm is identical in both cases. The success of one or the other depends on the physical angle of the array.

## Image analysis functions

Clicking the Analyse image button, , in an imaging window brings up another window showing available image analysis functions. When an image analysis function is selected, all data acquisition and processing stops and a new window specific to the selected image analysis function will appear.

Two analysis functions are currently implemented and work only for 2D images.

### 2D direct image sizing

This implements a simple –6 dB sizing method to the currently selected region of a 2D image. The peak amplitude of the selected region is found and all pixels within –6 dB of that are identified. The algorithm then attempts to fit the smallest possible rectangular box around these points. The final dimensions and orientation of the box give some indication of the size and orientation of a crack-like defect provided the defect is larger than the diffraction limit of the imaging algorithm.

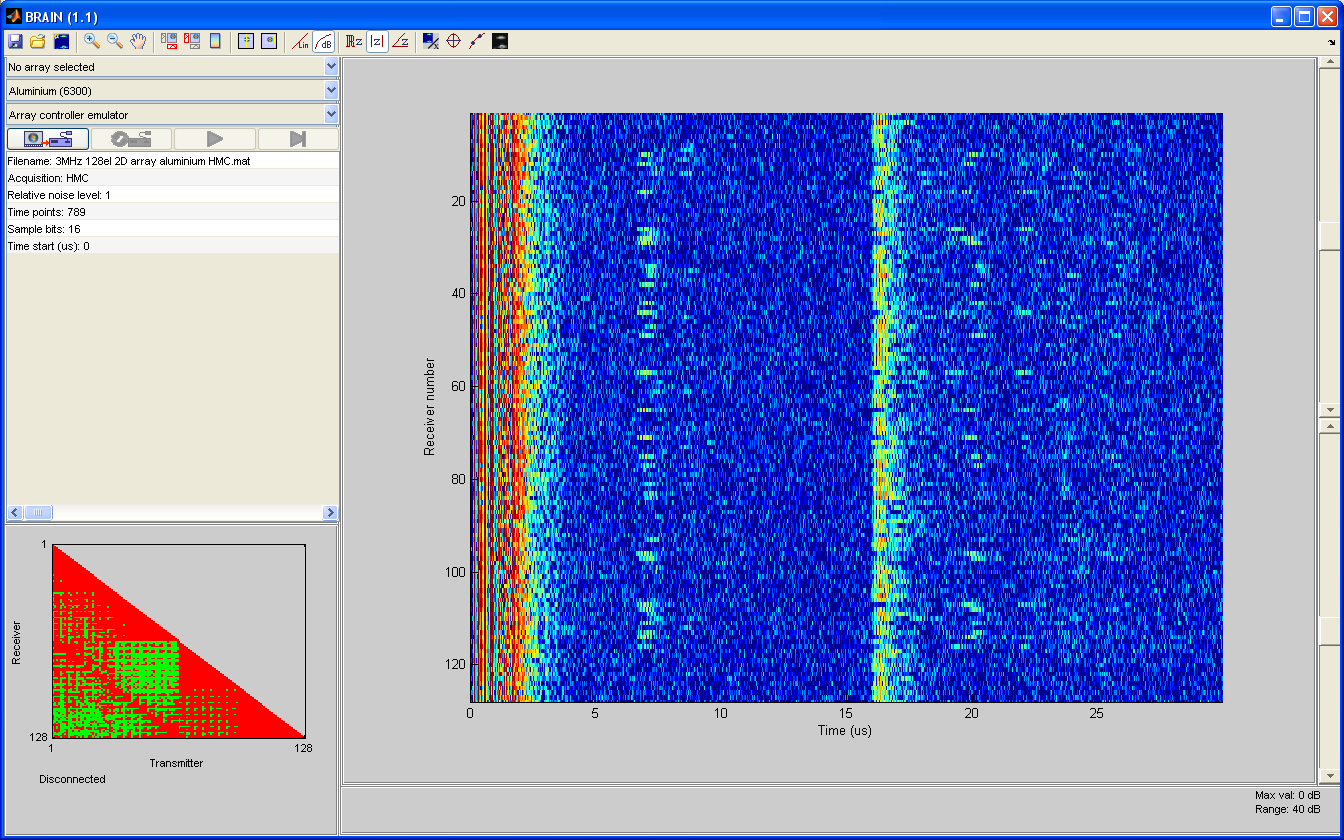
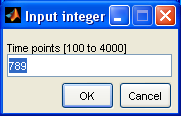
### 2D S-matrix (original method)

This extracts the S-matrix associated with the point of highest amplitude in the currently selected region of an image. The method is based on a simple moving aperture approach (not the more sophisticated and accurate technique that uses reversible imaging). Essentially, the selected point is probed from different angles and the scattered field recorded as a function of angle by sweeping transmit and receive apertures of a specified size along the array (entirely in post-processing).

## General interface notes

### Tables

Parameters for array controllers, imaging and analysis algorithms are displayed and edited from tables, as shown for example in Fig. 3(a). To edit a parameter, click on the relevant line in the table. A dialog box will be displayed, the format of which depends on the nature of the parameter. For numerical values a prompt with the available range as shown in Fig. 3(b) is displayed. Type in the desired value and press OK (or cancel to close dialog without changing value). The value will not be accepted unless it is within the range specified. For constrained parameters, the dialog box will contain a list of possible options as shown in Fig. 3(c). Click on the desired value and then press OK.

(a) (b)

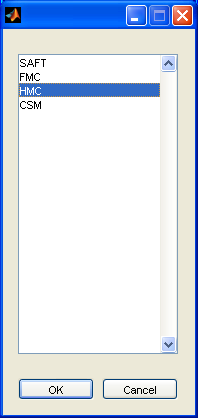
(c) 

Fig. 3 (a) Typical table of parameters, (b) numerical parameter entry dialog, (c) constrained parameter dialog.

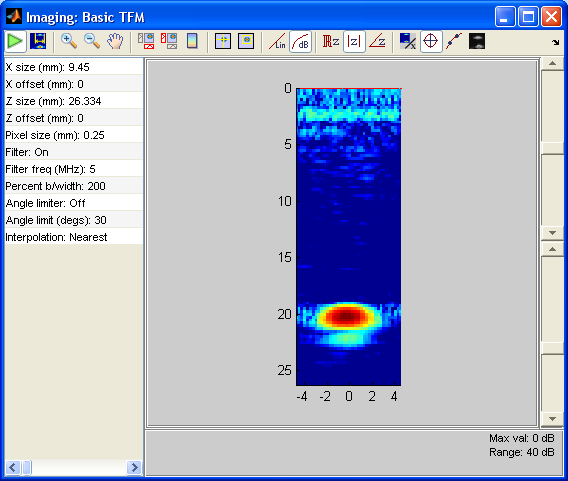
### 2D data display

Both the raw array data and the output of 2D imaging algorithms are displayed using the same routine, so the following description is common to both. When 2D data is displayed, the toolbar will contain some or all of the buttons shown in Fig. 4(a) and the display will appear with some or all of the features shown in Fig. 4(b). The operation of the toolbar buttons is summarised in Table 2.

In the data display itself, the main image is displayed in the middle, with optional cross-sectional views shown to the left and below. On the right hand side are two sliders (one if the scale is set to linear) which control the range of the image. The lower slider (log scale only) controls the range in dB relative to the maximum value. Small increments (i.e. clicks on the end arrows) are 1 dB and large increments (i.e. clicks on the bar itself) are 10 dB. The upper slider controls the maximum value displayed relative to the current normalisation value (i.e. unity on a linear scale or 0 dB on a log scale).

|  |  |  |
| --- | --- | --- |
| **Icon** | **Function** | **Description** |
|  | Zoom in | Switches to zoom mode. Click and drag on the image (or cross-section) to zoom in on a region; click once to zoom in by a factor of two and double click to zoom out. |
|  | Zoom out | Click to zoom out and set image size to boundary of data. Note that this does not happen by default when parameters that may change image size are changed. |
|  | Pan | Switches to pan mode, enabling image (or cross-section) to be clicked and dragged to pan. |
|  | Cross-sections | Turns the relevant cross section graph on or off. The cross sections are displayed based on the currently selected cross-hair position (i.e. the selected point or first point of a region selection) |
|  | Colour bar | Turns the colour bar on or off |
|  | Select point | Switch on point selection mode so that clicking on the image (or cross-section) selects a point and puts the cross-hairs through it. The status text at the bottom of the display gives the spot value at the point and the cross-hairs determine the position of the cross-sections to be displayed if the cross-section graphs are active. |
|  | Select region | Switch to region select mode so that clicking two successive points on the image defines a rectangular region and puts the cross-hairs through the first selected point. The status text at the bottom of the display gives the values associated with the image within the region and the cross-hairs determine the position of the cross-sections to be displayed if the cross-section graphs are active. |
|  | Linear | Use linear scale |
|  | Log | Use log scale |
|  | Real | Displays real part of complex data (i.e. the RF signal) |
|  | Argument | Display argument of complex data (i.e. phase angle) |
|  | Modulus | Display modulus of complex data (i.e. envelope) |
|  | Normalise | Change normalisation value of image (i.e. unity on linear scale, 0 dB on log scale). A dialog appears that depends on current selection. If neither point or region is selected, then the normalisation is with respect to global maximum of modulus anywhere in image; if a point is select then normalisation is no modulus at point; if a region is selected then normalisation is to either peak or RMS modulus value within region. |
|  | Aspect ratio | If selected, forces aspect ratio of image to unity otherwise image size is made to fit window size. |
|  | Interpolate | Performs Fourier domain interpolation of underlying complex data to improve resolution of image. This should only be used if underlying data is already adequately sampled (i.e. with a least half wavelength spacing between points). |
|  | Monochrome | Switch to monochrome display. |

Table 2 2D display toolbar button summary.

(a) 

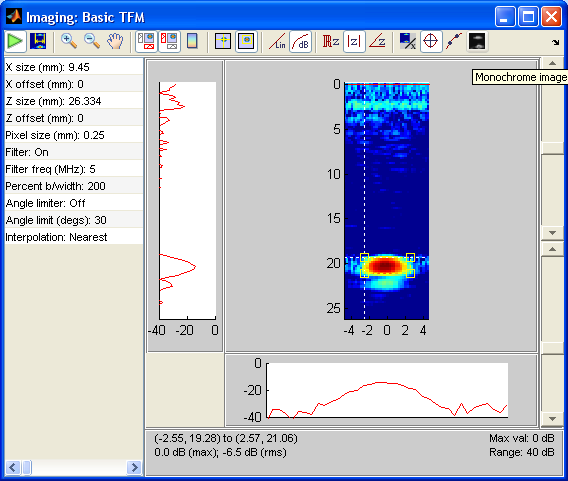
(b) 

Fig. 4 (a) Toolbar for 2D data displays, (b) example of 2D display with cross-section graphs on and image region selected.

### 3D data display

Preliminary version only – not finalised or fully documented. 3D data is displayed as an isosurface at a specified dB level. The display shows 3 elevations of the isosurface and a 3D view of the same, the latter of which can be rotated. The 3D view shows 3 orthogonal planes, the intersection of which defines a selected point. At any one time, one plane is fixed and clicks on the 3D image are interpreted as clicks on that plane, hence moving the other two planes so that the intersect at the clicked point in the fixed plane. The fixed plane is selected by clicking on one of the three elevations. 3D region selection requires 3 points; the first two define a 2D region in the current fixed plane and the third (in a new second orthogonal fixed plane) extrudes the 2D region accordingly. Normalisation functions are as in 2D data display.

## Array controllers

### Array controller emulator

This enables the operation of an array controller to the emulated for the purposes of testing algorithms. The parameters are listed in Table 3.

|  |  |
| --- | --- |
| **Parameter** | **Notes** |
| Filename | Filename of raw data to use. The files are standard array date files in the BRAIN x.y\Instruments\Emulator data folder but they should contain a complete data set (i.e. either FMC or HMC data but not CSM or SAFT data). The data output by the emulator is derived from that in the file according to "Acquisition" below, which may reduce the data to CSM or SAFT. |
| Acquisition | SAFT – pulse-echo data from each array element only ( time-traces)  CSM – fire all elements in parallel and record signals from each element ( time-traces)  HMC – fire on each element sequentially and record only linearly independent data ( time-traces)  FMC – fire on each element sequentially and record all data ( time-traces). |
| Relative noise level | Level of random white noise added to each frame of raw data when in play or play once mode. |
| Time points | How many points to record in each time-trace (range from 100 to 4000). |
| Sample bits | How many bits to use to digitise signals (selectable from 1, 2, 4, 8, 10, 12 or 16). The signals in the example files are typically acquired at 16 bit resolution, and are discretised down according to this value. |
| Time start (s) | Start of acquisition relative to transmitted pulse in microseconds (range 0 to *number*) |

Table 3 Array controller emulator parameters.

### Peak NDT Micropulse

The parameters for the Peak NDT Micropulse are listed in Table 4.

|  |  |
| --- | --- |
| **Parameter** | **Notes** |
| IP address |  |
| Port number |  |
| Acquisition | SAFT – pulse-echo data from each array element only ( time-traces)  CSM – fire all elements in parallel and record signals from each element ( time-traces)  HMC – fire on each element sequentially and record only linearly independent data ( time-traces)  FMC – fire on each element sequentially and record all data ( time-traces). |
| Sample frequency (MHz) | Sampling frequency in MHz, selectable from 25, 50 or 100 MHz. Note that changing sampling frequency causes a complete instrument reset which takes a few seconds. |
| Pulse-voltage (V) | Amplitude of rectangular transmitted pulse in volts (range from 0.1 to 100 V). |
| Pulse-width (ns) | Width of rectangular transmitted pulse in nanoseconds (range from 1 to 100 ns). |
| Time points | How many points to record in each time-trace (range from 100 to 4000). |
| Sample bits | How many bits to use to digitise signals (selectable from 8, 10, 12 or 16) |
| Gain (dB) | Gain in pre-amplifiers (range from 0 to 70 dB) |
| Filter number | Which analogue filter to use prior to digitisation (*details to be listed here*) |
| Maximum PRF (kHz) | Maximum pulse repetition frequency (range from 1 to 10 kHz) |
| Averages | Number of averages to use (1 to 256) |
| Time start (s) | Start of acquisition relative to transmitted pulse in microseconds (range 0 to *number*) |
| Instrument delay (ns) | Offset to the time axis of acquired signals to account for instrumentation delay. |

Table 4 Peak NDT Micropulse parameters.