SIGVIEW v2.6
User Manual

SIGVIEW v2.6 User Manual

Table of Contents

Part I

Introduction

1. Introduction

1.1. General

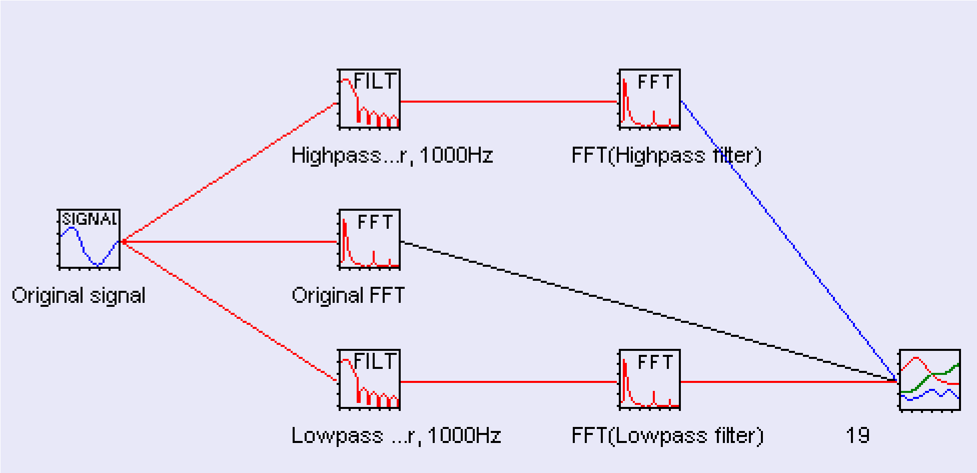
SIGVIEW is a real-time signal analysis application with the wide range of spectral analysis tools, statistical functions and comprehensive graphical solutions for 2D and 3D graphics. You can use *SIGVIEW* to analyze offline or live signals, freely combine all included analysis tools and concentrate on analysis logic instead on program usage.

SIGVIEW is distributed as a shareware software; you can download a 21-days evaluation version from http://www.sigview.com . For use after a 21-days trial period you must purchase *SIGVIEW* license.

1.2. Basic concepts

Unlike most other applications, *SIGVIEW* does not provide some firmly defined user interface for all purposes. It is designed as a collection of tools which can be combined in many different ways to create a user interface specific to your analysis. Once you are finished with the creation of your tool chain, you can save your application-specific “workspace” or "tool" and reuse it later. To be able to use the full power of *SIGVIEW*, it is very important to understand several basic concepts:

Analysis tree: All signals or signal analysis results in *SIGVIEW* are connected to other windows to form a tree of functional blocks: file input, data acquisition, FFT, instruments…You can imagine signals or analysis results to “flow” through that tree from one tool to another one following your analysis path. This analysis tree could look like this:



Visible part of the signal: If you perform any analysis on a signal window, it will be performed only on the currently visible part of the signal. Consequently, if you change the visible part (by moving through signal, zooming in or out,…), all child windows will recalculate and redraw its content.

Every array of values is a “signal”: Each sequence of X/Y values is considered to be a "signal" (audio signal, spectrum, etc.). Consequently, each sequence has its sampling rate, even if it was not actually created by digitally sampling some analog signal. Sampling rate of an FFT is simply number of its values (bins) in one Hz. Following this logic, it is possible to perform, for example, FFT analysis on a FFT result. Even if it seems like it does not make sense, it can sometimes be interesting to see the results. The important thing is that *SIGVIEW* does not create artificial rules, telling you what you should and what you should not do. You are free to experiment and to find you own way to get the results you want.

Signal graph/Linked windows : If one window is created as a result of the analysis from any other window, then those two windows are *linked*. The network of linked windows works like a complex analysis machine where output of each window is the input for its child windows. It means that all changes in one window will cause its child windows to recalculate and redraw their content. A good example is a part of the signal and its FFT; if you slide the part of the signal through the whole signal, FFT will recalculate on each move to show FFT result of each new segment.

Moving through signals: You will usually work and perform analysis on a small part of a longer signal. If your signal is zoomed-in to a smaller part, you can use arrow keys or VCR-like commands to move through the signal and observe changes in analysis results. One of the standard procedures for working with signals could be: Load one long signal and zoom-in to some power-of-2 length segment. Then perform some analysis, for example FFT, then "Track changes as 3D graphics" of that FFT, and finally return to the signal and use left/right arrow keys to move the segment through the whole signal. That way you will be able to observe spectral changes in a signal through time.

Data acquisition/monitoring : You can work with data acquisition window as you would with any other static signal window. The only difference is that the signal you record will change (if data acquisition is started) in regular intervals, depending on your data acquisition settings. That will cause your signal analysis system to recalculate and redraw all windows connected with data acquisition window. If your PC is fast enough, you will be able to create or change your analysis system while data acquisition is running; otherwise it might be better to create the system first and than start with the data acquisition.

Control window : Working with many signals and analysis windows at the same time can become quite confusing. In those situations, you can use Control window to display your signal analysis system in a tree-view form where you can easily understand signal flow, perform operations on multiple signals, hide and show windows and choose which signals to show as overlay.

Properties: You can change parameters for most analysis windows even after they are created. If this is possible, Edit/Properties menu options will be enabled in the main menu or in the context menu for specific window. By using this option, you will be able to change window properties like cutoff frequencies for filtered signal, spectral analysis settings for FFT and Time-FFT, scale limits for instruments, etc.

Part II

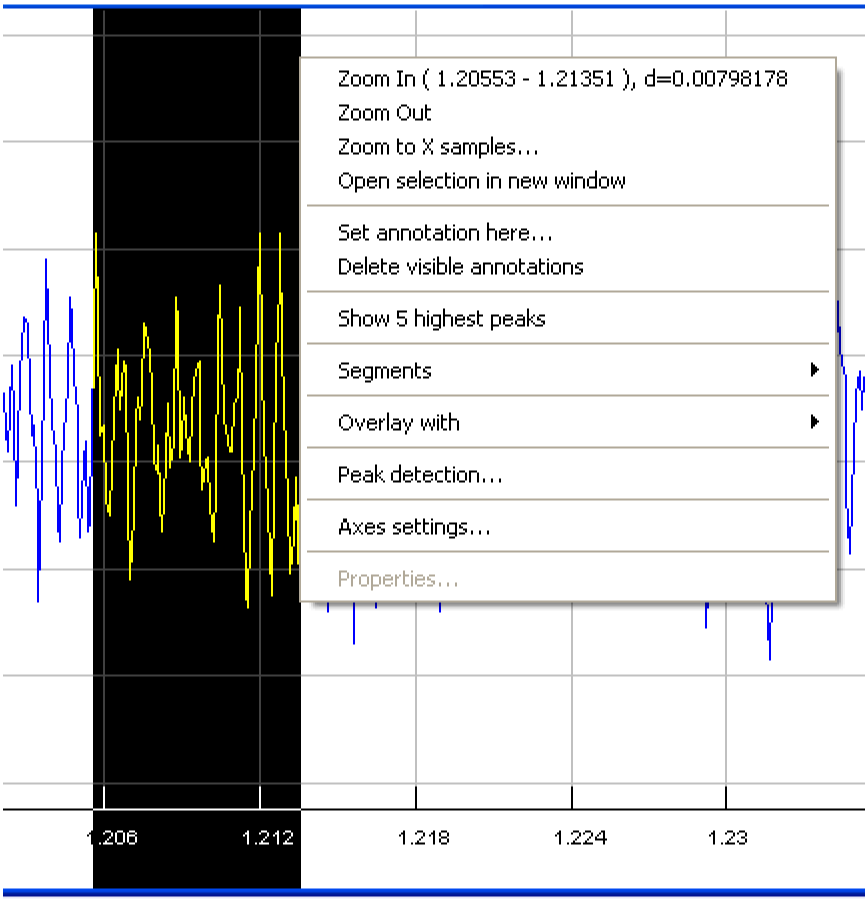
Basic signal operations

2. Basic signal operations

2.1. Selecting part of the signal

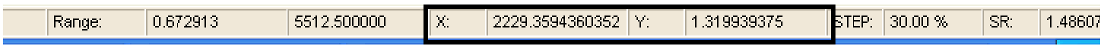
You can select a part of the signal by holding the left mouse button pressed and moving the signal ruler across the signal with your mouse. A selected part will be shown with inverted color (black background). If you want to extend the selection to the beginning or to the end of the signal, you can press *Home* or *End* **keyboard** keys during the selection.

If you select some part of a signal as explained above and press the right mouse button, you will see the exact position and length of the selected signal part in the context menu. This can be used to measure a part of a signal, for example the difference between two FFT peaks or the duration of a signal part.



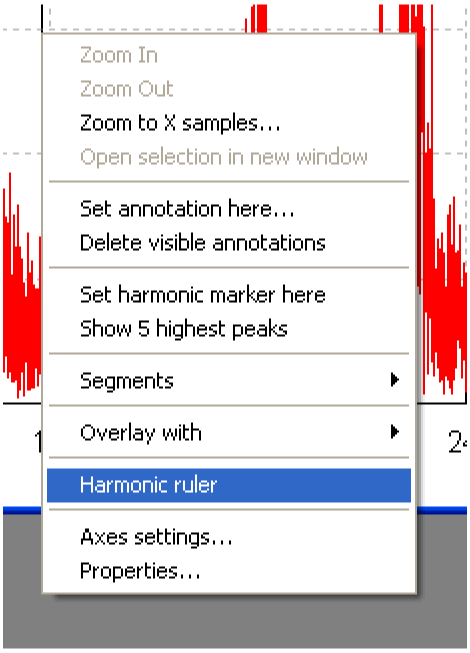
2.2. Signal ruler

A Signal ruler is a vertical line which moves over the signal following the mouse pointer. The current position of the ruler and the corresponding value from the signal at that position are available on the status bar in the bottom on the screen.



Range: 0.672913 5512.500000 X: 2229.3594360352 Y: 1.319939375 STEP: 30.00 % SR: 1.48607

By turning *Harmonic ruler* option on in the context menu of FFT and signal calculator windows, the ruler will be shown on the current position of the cursor as well as on all harmonics of the current frequency (current frequency x2, x3, x4, etc.)

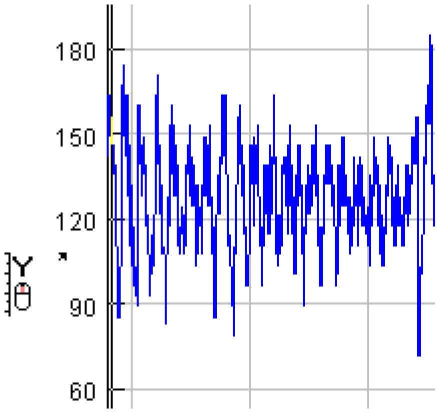


2.3. Signal axes settings

There are two ways to set Y-axis range and position: a fast method by using mouse-wheel directly on a signal and a standard method by using Axes settings dialog from the context menu.

Fast axes settings by using mouse-wheel

To use the fast Y-axis setting method, move your mouse to the area between the left edge of the signal window and Y-axis display. The mouse pointer will change to show a symbol as on this image:

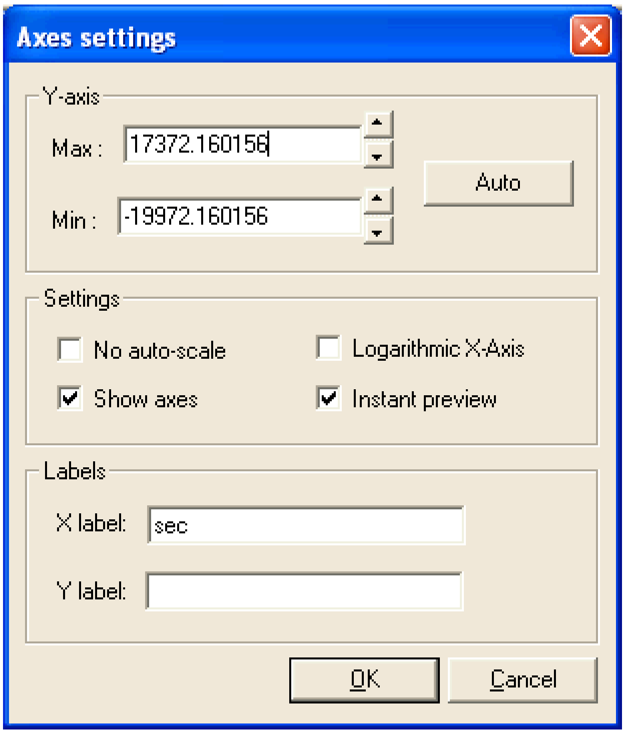


Turning a mouse-wheel forward will zoom-in the Y-Axis around the current position of the small arrow in the cursor pointer. Turning the mouse-wheel in the opposite direction will zoom-out the Y-Axis. Again, the center position for this operation will be position of the small arrow in the mouse pointer.

If you press Ctrl button during mouse-wheel moving, Y-axis will move up or down instead of zooming-in or out.

Axes settings by using a dialog

Axes settings for signals can also be performed in Axes settings dialog, which appears after selecting *Edit/Axes settings* option or choosing Ctrl+A on active signal window. There are several fields in this dialog:



Axes settings

Y-axis

Max : 17372.160156

Min : -19972.160156

Settings

□ No auto-scale

□ Logarithmic X-Axis

☑ Show axes

☑ Instant preview

Labels

X label: sec

Y label:

Max & Min fields are used for setting Y-axis range. You can enter your own values manually in corresponding edit boxes or you can change the values by pressing up/down arrow keys on the right side.

Auto button sets values in min/max fields to actual min/max values from the signal.

No auto-scale check box: unselect it if you want to keep Y-axis settings for this signal even when signal values change. If you select this check-box, *SIGVIEW* will not calculate new min/max values for the graphic's Y-axis automatically each time the signal changes.

Instant preview check box: select it if you want signal axes to redraw while you change minimum or maximum values in this dialog

Show axes check box: select it if you want to see coordinate system with X and Y axes and labels. Drawing signal only without the axes can speed-up the signal redrawing in cases where performance is very important.

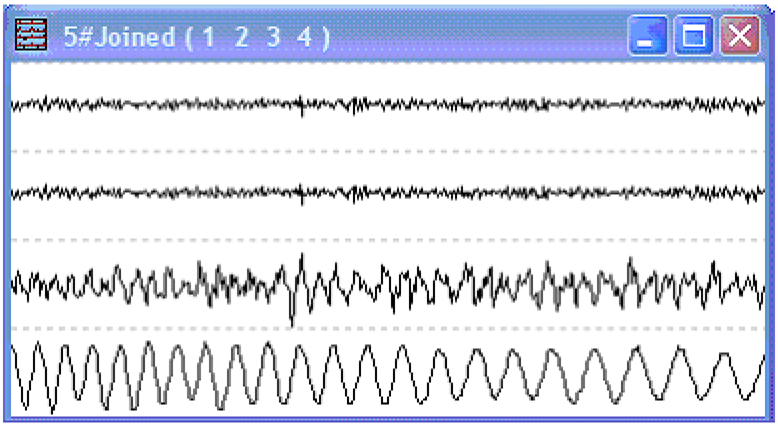
X/Y Labels: a text which appears at X and Y axes (usually units like Hz, s,...)

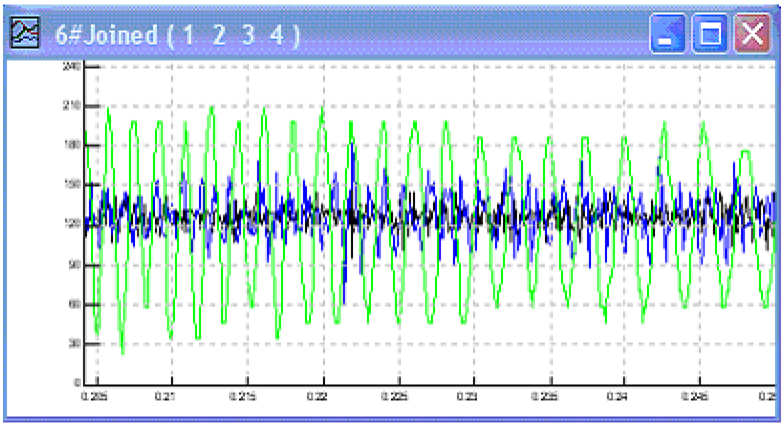
Logarithmic X-Axis: If this option is selected, the X-Axis will be shown as logarithmic axis, i.e. the distance between data points will increase logarithmically from left to the right side of the displayed part of the signal. By using this view on an FFT result, lower frequency components will take more space, and the higher frequencies will be shown "compressed" at the right side of the graphics. Showing logarithmic Y-axis is not possible in general. If you need to show your spectrum values on a dB (log) scale, you can try displaying FFT result in decibel.

2. Basic signal operations

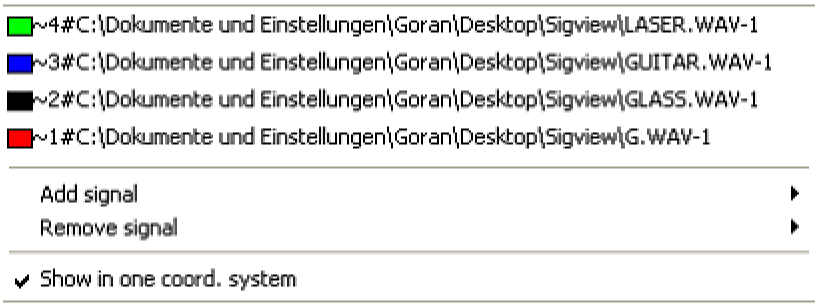
2.4. Overlays

If you want to show more than one signal in the same window or even in the same coordinate system, *SIGVIEW* offers two different views: “parallel” view where signals are shown in the same window on top of each other (for example, for typical EEG signal view), or real “overlay” view where signals are shown in the one coordinate system.





There are two ways to create these views: the first one is to selected several windows in Control window and then choose one of “Show selected in one window” or “Show selected as overlay” options from the context menu (right mouse click in the empty part of the Control window). The second one is to use the context menu of the signal or overlay window to simply add other signals to the overlay. Each of the signals shown as overlay in one coordinate system will be represented with a different color. You can distinguish the color of different signals if you look at the control window. The color of the link between signal window and Overlay window will always be the same as the color of that signal in Overlay window.


🟩~4#C:\Dokumente und Einstellungen\Goran\Desktop\Sigview\LASER.WAV-1
🟦~3#C:\Dokumente und Einstellungen\Goran\Desktop\Sigview\GUITAR.WAV-1
⬛~2#C:\Dokumente und Einstellungen\Goran\Desktop\Sigview\GLASS.WAV-1
🟥~1#C:\Dokumente und Einstellungen\Goran\Desktop\Sigview\G.WAV-1
Add signal ▸
Remove signal ▸
✓ Show in one coord. system

You can change the order of signals in the Overlay window by pressing **Ctrl+Z** keyboard shortcut. In “parallel” view, the order of signals will be shifted; in “overlay” view colors of the signals will be shifted.

You can switch between “parallel” and “overlay” view anytime by pressing **Ctrl+S** keyboard shortcut.

Note! If windows and their signals have different length or units, *SIGVIEW* will try to show all of them in the same coordinate system in overlay window - of course, this does not mean that the units shown will be valid for all signals. The coordinate system will always be taken over from the first signal, the red one. By shifting the signals order with Ctrl+Z keyboard shortcut, you can see the coordinate systems of other signals too.

2.5. Moving through signal and zooming

For advanced zooming and moving through a signal by using a mouse-wheel please see Using mouse wheel for zooming and moving through signal.

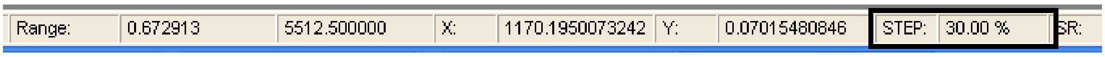
Zooming signal in & out: Select a part of the signal you want to zoom in and press the zoom-in button on the toolbar or choose a Zoom-in option from the context menu. You can repeat this as many times you want, and *SIGVIEW* will memorize your position in signal for every zoom step you make. You can use the up & down arrow keys or and buttons in a toolbar to move through your zoom history i.e. zoom in and out of the signal.

Even a faster zoom method is to use your mouse wheel.

Zooming signal to arbitrary length: It is often useful to zoom in a signal to a part which length (in samples) is equal to a power of 2. It will have great effect on FFT computation speed and precision. You can do it by pressing 2^n button on a toolbar or choosing *Edit/Zoom to X samples/values* in the main menu. A dialog will appear with all possible power of 2 lengths that your signal can be zoomed-in to. Choose the wanted length end press OK. You can also use this option to zoom-in to some arbitrary, non power-of-2 signal length. Simply enter the wanted length in the corresponding edit-box.

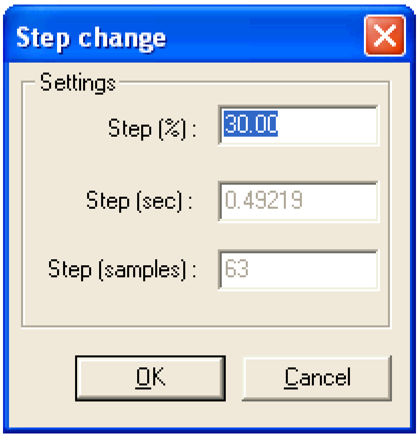
Moving through signal: After you zoom in a part of the signal, you can use it to slide through the whole signal by using left & right arrow keys. The speed depends on the *step* value, which represents the percent of the visible part of the signal you move each times you press the arrow key. You can see this value for the currently active signal on the status bar.

Even faster method for moving through signal is to use your mouse wheel.



Range: 0.672913 5512.500000 X: 1170.1950073242 Y: 0.07015480846 STEP: 30.00 % SR:

Step can be changed for the current window by choosing *Play&navigate/Step change* option in the main menu or by pressing Alt+Up or Alt+Down keys while signal window has focus.



Step (%): 30.00

Step (sec): 0.49219

Step (samples): 63

2.6. Using mouse wheel for zooming and moving through signal

Starting from *SIGVIEW v2.0*, you can use a mouse-wheel for faster zooming in/out and moving through signal.

X-Axis

If you simply position signal ruler somewhere in a signal and turn your mouse-wheel forward (away from you), the signal will be zoomed-in to a 20% of its original length. The ruler will remain on the same signal position. Zooming-out can be performed by turning a mouse-wheel in the opposite direction. Please note that these zoom steps will not be saved in a zoom history.

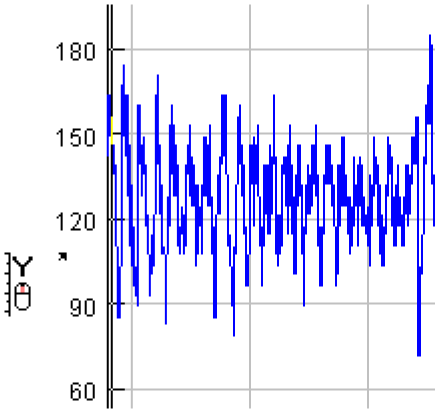
There are also two keyboard modifiers which can be used to achieve a special zooming and moving functions in combination with mouse wheel:

Ctrl key + mouse wheel: Instead of zooming, you will move through the signal by turning mouse wheel. Turning forward will be equal to pressing the left arrow key and turning backwards will be equal to pressing the right arrow key.

Alt key + mouse wheel: Zooming in/out will be performed as described above, but the length of a signal segment in samples will always be a power of 2 (i.e. 1024, 2048,4096,...) for each step. It is especially useful for zooming in a signals where FFT function will be performed afterwards.

Y-Axis

To use a mouse wheel to control Y-axis, move your mouse to the area between the left edge of the signal window and Y-axis display. The mouse pointer will change to show a symbol as on this image:

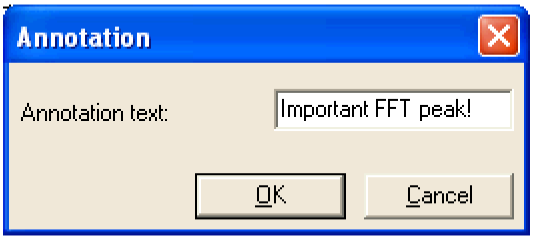


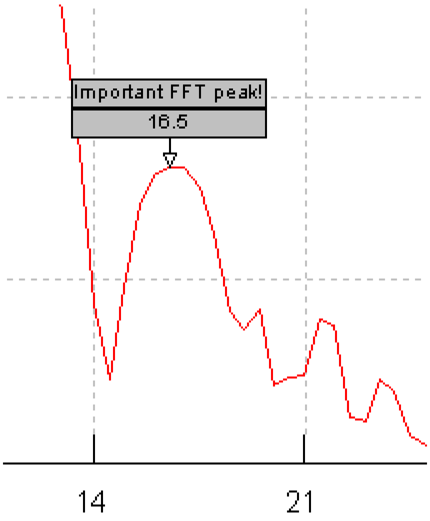
Turning a mouse-wheel forward will zoom-in the Y-Axis around the current position of the small arrow in the cursor pointer. Turning the mouse-wheel in the opposite direction will zoom-out the Y-Axis. Again, the center position for this operation will be position of the small arrow in the mouse pointer.

Ctrl key + mouse wheel: Instead of zooming, you will move the complete Y-Axis up or down.

2.7. Signal annotations

To label some interesting point in signal or analysis result, you can use signal annotations. Those will be displayed as markers containing some user-specified text and pointing to some signal location.





You can create annotations by choosing the corresponding context menu option on the signal. Simply position a signal ruler on the location where you would like to define an annotation, press the right-mouse button and choose *Set annotation here…* menu option. A dialog will appear where you can enter your annotation text of up to 32 characters length.

These annotations will be saved in a workspace file and reloaded later when you load SWS file. This feature is especially interesting if you want to exchange your signal or analysis results with somebody and would like your comments to remain visible.

Signal annotations can be deleted by choosing *Delete visible annotations* option from the context menu. If you would like to delete single annotations, simply zoom-in until that annotation is the only one visible and then choose *Delete visible annotations* option from the context menu.

2.8. VCR-style commands for playing and moving through signal

For a fast navigation and replaying the signal, *SIGVIEW* provides VCR-like signal navigation functions. All functions are accessible from the *Play&navigate* menu item or in a toolbar*:*

⏪ **Rewind**: moves to the beginning of the signal

▶ **Play (no sound)**: Moves through the signal just as when you press and hold the right arrow key. All child windows are recalculated and repainted on each step. There is no sound being played. The step for moving through the signal will be a current step for that signal. *SIGVIEW* will adjust a replay speed to the sampling rate of the signal, i.e. replaying will be performed in a real-time. If you would like to process signal as fast as possible, select *Play&Navigate/Play as fast as possible* menu option. This option works globally, i.e. will be applied to all signals after you change it in the main menu.

🔈 **Play with sound:** Plays signal on your sound card and moves through it at the same time. For example, if you zoom in to a block of 4096 samples length in the signal and use this function, *SIGVIEW* will play the visible part of the signal, move 4096 samples further, play the next block etc. until the end of the signal. It is very useful for audio signals because you can observe the analysis and hear the sound at the same time.

If you try to play the signal and there is no sound, maybe the signal amplitude is simply not high enough. *SIGVIEW* expects signal amplitude values in a 16-bit range, i.e. -32767…32767. If you would like *SIGVIEW* to automatically adjust signal volume before playing, you can turn *Play&navigate/Adjust volume automatically* menu option on.

If you would like *SIGVIEW* to start playing a signal from the beginning when one of replay options arrives signal end, please select *Play&navigate/Play repeatedly* menu option.

🔈 **Play visible segment only:** Similar as previous function, except that *SIGVIEW* will only play the visible part of the signal and then stop playing.

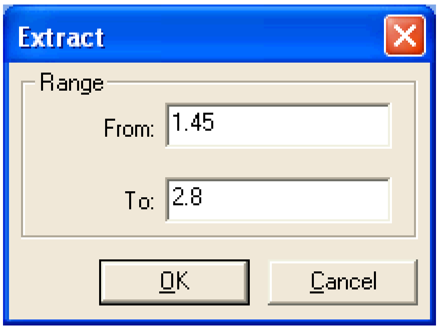
⏩ **Fast forward**: moves to the end of the signal

STOP **Stop:** stops any of the running Play functions

2.9. Extracting parts of the signals

There are two ways to extract a part of the signal into a new window:

1. *Edit/Extract signal part (from-to)* option from the menu. A dialog appears where you define limits of the part you want to extract (the units are the same as X-axis units of a signal). A new window with extracted part of the signal will appear. It will always contain part of origin signal (for example, 1000-2000Hz part from FFT sequence) that you defined in the Extract dialog. It is useful for extraction of one part of the spectrum in separate window. You can use it to observe several frequency ranges in separate windows and calculate their statistics values.



From: 1.45

To: 2.8

2. *Edit/Open selection in new window* option from the menu. Select part of the signal and choose this option from the menu or press *Alt+e*. A new window with extracted part of the signal will appear. It will always contain part of origin window which was selected in the moment of creation. It is useful for quickly extracting some visible part of the signal. For example, you can select a middle 1/3 of the window. No matter how the content of the origin window changes (zoom in/out, recalculation), extracted part will always contain the middle 1/3 of the origin window, regardless of units or current values on the X-axis.

2.10. Changing signal values

Although *SIGVIEW* is mainly being used for a signal analysis without changing original signal values, it is sometimes necessary to remove some unwanted artifacts from the signal. *SIGVIEW* offers several functions for altering signal values:

First, you must select the part of the signal you want to change.

- By pressing Ctrl+M, all values in the selected part of the signal will be set to the average (mean) value of the visible part of the signal.
- If you press Ctrl+0, all values of the selected part of the signal will be set to zero.

2.11. Exporting window content as image

If you want to save the graphical content of a window as image, use *Edit/Copy picture to clipboard* from the main menu.

The image will be transferred to the Windows Clipboard as bitmap, and you can paste it to any other picture editing program by choosing its *Edit/Paste* menu option.

2.12. Working with clipboard

You can use a clipboard to cut, copy or paste parts of the signals, the same way you would do it with the text in any text-processor.

- Cutting will be enabled only if you select a part of the signal. If you choose *Edit/Cut* option from the main menu or press ✂ button on the toolbar, the selected part of the signal will be moved to the clipboard (and removed from the signal). You can also use a standard Windows shortcut Ctrl+X for this operation.

- If you choose *Edit/Copy* option from the main menu or press button on the toolbar, a selected part of the signal will be copied to the clipboard. You can also use standard Windows shortcut Ctrl+C for this operation.

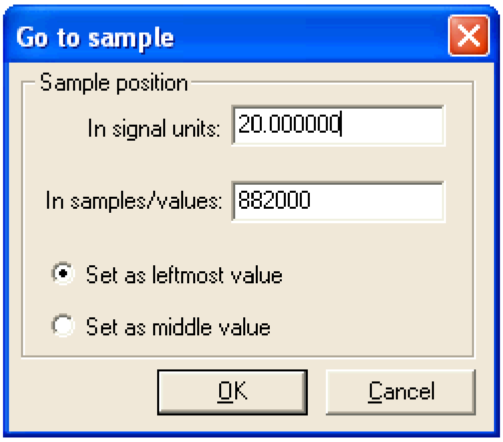
- Pasting signals from clipboard will be enabled only if some signal part already exists on the clipboard. Just put the signal ruler on the position in the signal where you would like to paste the content of the clipboard and press *Ctrl+V.* The signal part will be inserted on that position.

These clipboard functions are meant to be used only internally in *SIGVIEW*. It is not possible to use them to exchange signal data with other applications. There is another menu option, *Edit/Copy data to clipboard* which enables you to copy signal values to the clipboard in a text format. These data can be pasted as text in any other application, for example MS Excel or MS Word. The data format is the same as in files exported by using ASCII file export options.

2.13. Go to sample function

Analyzing signals with *SIGVIEW* includes often zooming-in to a smaller part of the signal and moving that part through the whole signal to observe changes in the time or frequency domain. For better navigation in a longer signal, *SIGVIEW* includes "Go to sample" function which enables you to position your currently visible part of the signal to exact absolute position in the whole signal. Please note that this function requires the selection of a smaller part of the signal - it will not zoom-in or change the visible part in any way.

You can start this function from the main menu, *Play&Navigate/Go to sample...* option.The following dialog will appear:



Go to sample

Sample position

In signal units: 20.000000

In samples/values: 882000

● Set as leftmost value

○ Set as middle value

You can define signal position in signal units (for example seconds, Hz, etc.) or in a number of samples/values. These two values are mutually depending and both will be synchronized by *SIGVIEW* each time you change one of them.

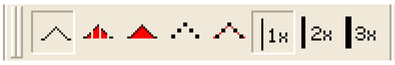
You can also choose if the value entered in above two fields should be used to define leftmost or center value of the visible signal part.

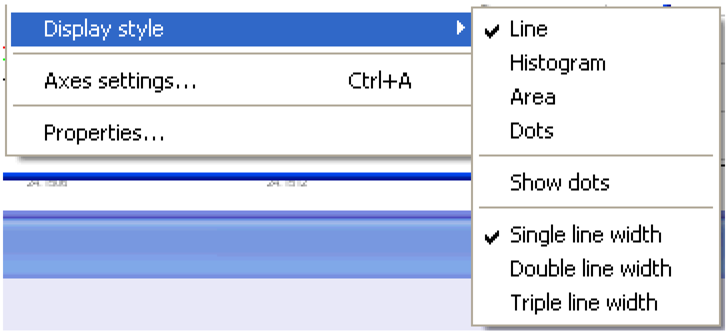
This function can also be used from the Control Window to set multiple signals to the same position, i.e. synchronize them before starting some analysis operation on multiple signals. To use this feature, simply select one or more signals in Control Window and choose "Go to sample..." function from the context (right-click) menu. *SIGVIEW* will use the sample rate of the first selected signal to convert signal units to samples in the dialog, but will position each signal to its correct position in signal units.

If requested position is invalid for the actual signal, for example 5.2 seconds for a 5 seconds long signal, *SIGVIEW* will try to move signal part as much as possible in the requested direction, i.e. will move to the end for the signal in this case.

2.14. Signal display options

SIGVIEW includes various options for signal displaying. Those are accessible for each signal window from its context menu or from a toolbar.

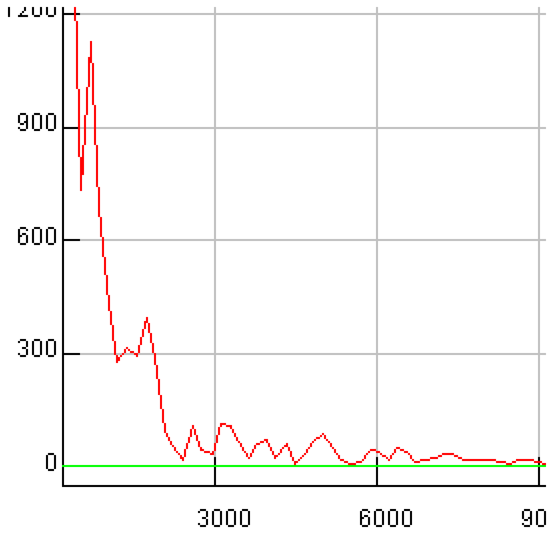




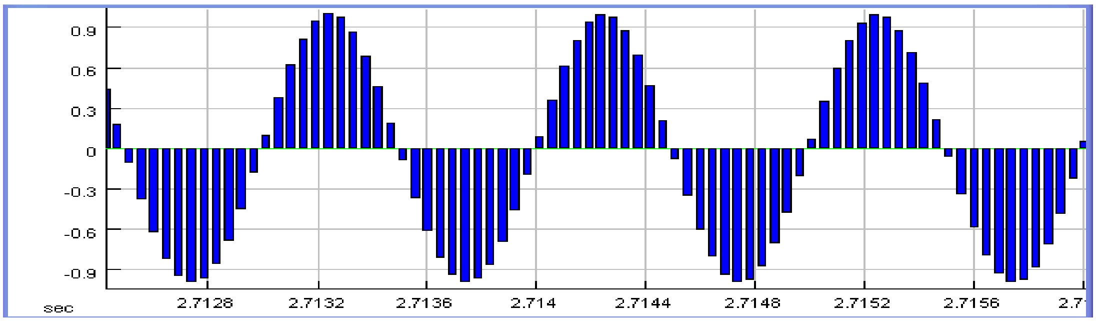
Default signal display type (line of single width) is by far the fastest way for displaying signals and should be used always if speed is important.

There are 4 different display types:

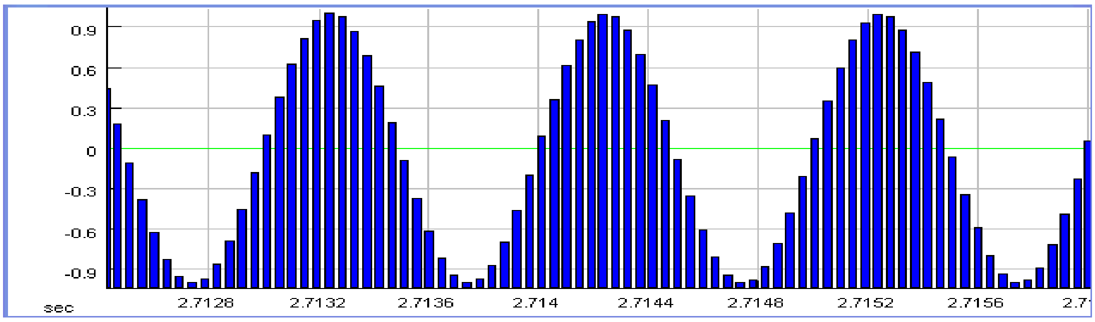
Line: signal will be shown as a line connecting actual signal values



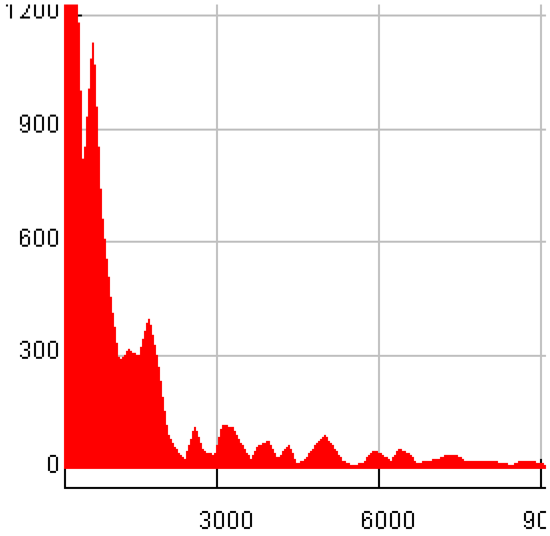
Histogram (Zero reference): For each signal value, a vertical rectangle will be drawn starting on the value of 0. This is useful if signal does not have too many values, for example for a probability distribution graph.



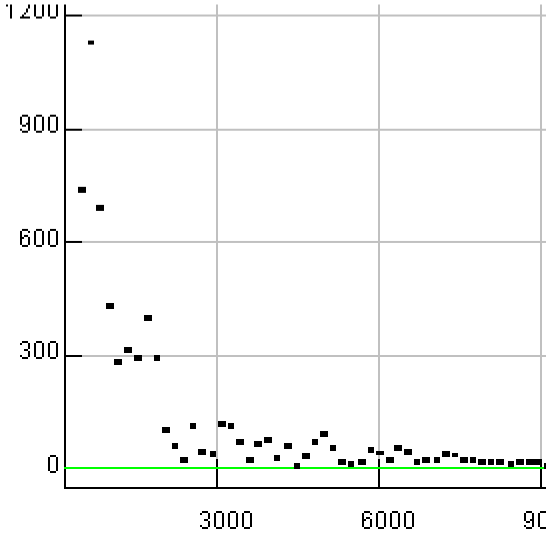
Histogram (Min reference): Very similar as Zero reference histogram graphics, but each rectangle will be drawn from the minimal value of the signal and not from the 0.



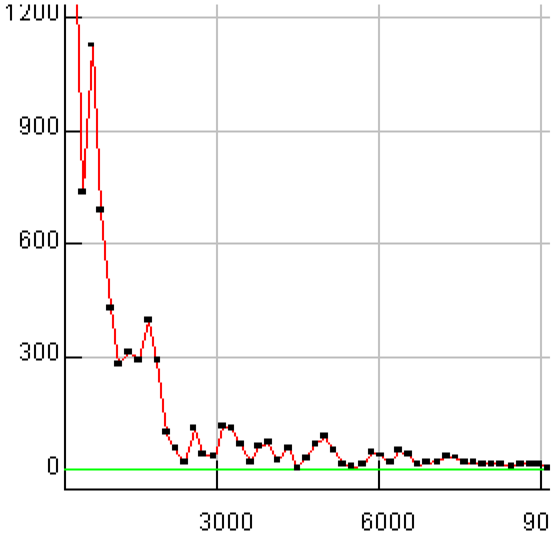
Area: With this display option, the area between a signal line and zero line will be filled.



Dots: A single dot will be drawn on a position of each signal value



Show dots option can be used in a combination with Line, Histogram and Area display type to show dots on samples positions additionally to a standard display:



As a combination with the "Line" display type, a **line width** can be chosen as single (default), double or triple line-width.

2. Basic signal operations

Part III

Loading and saving

3. Loading and saving

3.1. Loading and saving signals

Loading signals: After you select *File/Open signal* option from the menu or the 📂 button from the toolbar, the Open signal dialog will appear. In this dialog, you can choose a file type and the name(s) of the file(s) you want to load in *SIGVIEW*.
You can also use drag 'n drop to drag one or more files from Windows Explorer and drop them on *SIGVIEW* window. The effect will be the same as if loading these files through Open file dialog.

Loading compressed signals: Compressed signal file formats like MP3 or WMA require a special decoder component to be opened (codec). These components are not included in *SIGVIEW,* but are most probably already installed in your system*. SIGVIEW* uses Microsoft DirectX technology and codecs already installed in your system for file decompression. That ensures that any file playable on your computer (for example in Windows MediaPlayer) will most probably also be readable by *SIGVIEW*.

Loading very long signals: Since *SIGVIEW* loads the complete signal in RAM memory of your computer, a maximal size of one signal will be around 300-500MB. This is usually the limit for allocating one continuous memory block under MS Windows. This is equal to ~30-40min of 16-bit signal at 48kHz. If you try loading a longer signal, *SIGVIEW* will report that it can not allocate enough memory for it and offer you to load a smaller part of the signal which can fit in memory.

Saving signals: You can save a signal if you choose *File/Save signal as…* option or the 💾 toolbar button. *SIGVIEW* can save signals in WAV format (16-bit integer or 32-bit float) or export them as ASCII or raw binary file. Saving signals to a WAV file is possible only for signals with integer sampling rate. Also, please note that a 16-bit WAV format supports only integer sample values - it means that all values from your signal will be rounded before saving those in a WAV file. If you do not want this to happen, you can save your signal in a 32-bit floating point WAV format. You can save only the visible (currently zoomed-in) part of the signal by using “Save visible signal part as…” menu option. For saving multi-channel signals, please see Saving multi-channel signals chapter.

Replacing the file: If you want to perform the same analysis on several similar signals, there is an easy way to do it: you can create the complete analysis system for the first signal and reuse it for all others. Just replace the signal with some other signal by choosing *File/Replace with...* option from the menu and observe the results. The only restriction is that both original and replacement files must have the same length and the sample rate**.**

ASCII files: By using main menu options *File/ASCII files/...* You can also export/import signals in a standard tab, comma or semicolon separated ASCII format. Here is an example of TAB delimited file:

X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
....

There are also a separate menu options for loading ASCII files using dot as decimal separator (default) and comma (used in Europe).

When you export your signal to a ASCII format from *SIGVIEW*, there are two options available: *Export signal (X/Y values)* will export both X and Y values and *Export signal (Y values)* will export only Y values*.*
When loading ASCII files, *SIGVIEW* will try to analyze the first data column to see if it can be used as X axis values (if values are equidistant). If yes, you will be asked if you would like to use the calculated sampling rate for the loaded signal. If the calculation was not 100% correct because there were not enough data in the file, you can still accept this value and change it later through the *Edit/Sample rate change* menu option.

For saving multi-channel ASCII files, please see Saving multi-channel signals.

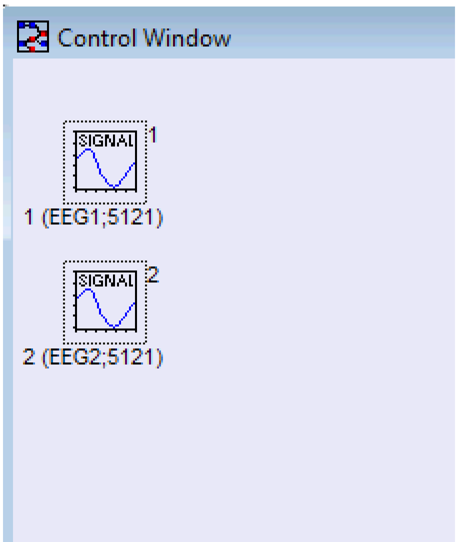
Please note that *SIGVIEW* saves only a visible (zoomed-in) part of the signal. If you want to save the whole signal, use the zoom-out option first. If you have your data already loaded in some other application, for example Microsoft Excel, the ASCII file format would be the perfect file format for data exchange. Just save your data in the other application as tab-delimited TXT/ASCII file, and import it in *SIGVIEW* afterwards.

Exporting 3D graphics values in file: By using main menu option *File/ASCII files/Export 3D-graphics*, you can export all values from the visible part of the 3D graphics in one text file. There are 3 options: export in a file with X,Y,Z triples in 3 columns (tab delimited), export a matrix where first row contains X values, first column Y values and the rest of the file are Z values (tab delimited) and finally inverted variant of the second matrix format (with switched rows and columns). All of these file types can be loaded by, for example, MS Excel for further analysis.

3.2. Saving multi-channel signals

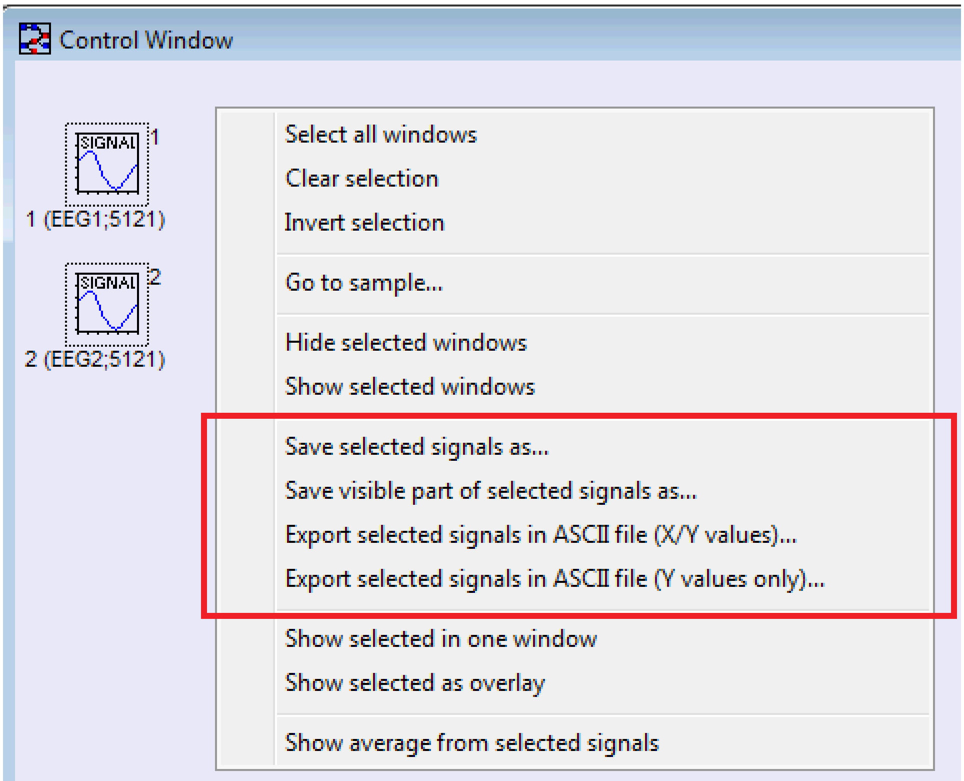
Saving multi-channel signal can be performed by using Control Window.

First step is to find icons in Control Window corresponding to the channels of the signals you would like to save and selecting them in the right channel order. Next to the each selected icon, a selection order will be shown (1,2,...). It will correspond to the channel order in a saved file.



After selecting all channel in the right order, open context menu (right mouse button in the empty part of a Control Window).

3. Loading and saving


Control Window
SIGNAL 1
1 (EEG1;5121)
SIGNAL 2
2 (EEG2;5121)
Select all windows
Clear selection
Invert selection
Go to sample...
Hide selected windows
Show selected windows
Save selected signals as...
Save visible part of selected signals as...
Export selected signals in ASCII file (X/Y values)...
Export selected signals in ASCII file (Y values only)...
Show selected in one window
Show selected as overlay
Show average from selected signals

The following options will be available if all selected signals are suitable for saving in one file (see below):

- **Save selected signals as**: Saves selected signals in a WAV file (16-bit integer or 32-bit float)

- **Save visible part of selected signals as:** Saves only a currently visible part of each selected signal in a file

- **Export selected signals in ASCII file (X/Y values)**: Saves selected signals in an ASCII file including X/Y values. X-axis values from the first channel will be used.

- **Export selected signals in ASCII file (Y values only)**: Saves selected signals in an ASCII file. Only signal amplitude values will be saved.

In order to save multiple channels in one file, all channels must have the same sampling rate (i.e. distance between samples). Furthermore, if the signals have different lengths, the longer ones will be saved only up to the length of the shortest one.

3.3. File formats

See also general topics about loading and saving signals.

Standard *SIGVIEW* version can read signals stored in several different file formats:

***.WAV files**

Standard Wave audio format (8, 16, 24 or 32-bit) including compressed WAV files if corresponding codecs are installed in the operating system.

Compressed file formats

Use *File/Open signal…* menu option. You will be able to open most compressed file formats like compressed WAV files, MP3, WMA, ASF etc. *SIGVIEW* will use Microsoft's DirectShow and installed codecs for the file decompression. That ensures that any file playable on your computer will also be readable in *SIGVIEW*. Please note that file decompression can be a rather slow operation, and the resulting signals can be very long.

Audio Interchange File Format (.AIF, .AIFC, .AIFF)

SIGVIEW uses DirectX services for loading these files. Therefore, a Windows Media Player v7 or higher is required.

Sun Microsystems and NeXT audio files (.AU,.SND)

SIGVIEW uses DirectX services for loading these files. Therefore, a Windows Media Player v7 or higher is required.

ASCII files

Available through ASCII Import/Export options in File menu. The ASCII files have to be in a following format (choose appropriate option if your files use comma as decimal separator):

X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>
X value<TAB>Y value channel1<TAB>Y value channel 2<TAB>...Y value channel N<new line>

Raw binary files

Available through File/Raw binary files/Export and Import menu options. Raw binary files are simple binary arrays of samples, saved in one of these supported formats:

8-bit signed
8-bit unsigned
16-bit signed
16-bit unsigned
32-bit float

Only one signal (channel) can be loaded from one raw binary file.

EDF (European File Format)

The European Data Format (EDF) is a simple and flexible format for exchange and storage of multichannel biological signals. EDF was developed by a group of European medical engineers and published in 1992 in Electroencephalography and Clinical Neurophysiology 82, pages 391-393. Since then, EDF became a de-facto standard for EEG and PSG recordings in commercial equipment and multicenter research projects. You can find further information about this format at: http://www.hsr.nl/edf

3.4. Loading and saving workspaces

By using *“Load Workspace...”* and *“Save workspace...”* options in a *File* menu*,* you can save all currently opened *SIGVIEW* windows and their relations (a "Workspace") in a file and reload it later.

Concept

SIGVIEW does not save the actual content of each signal or analysis result – it saves only a structure of your analysis system and a file names for loaded files. For example, if you load a signal, perform an FFT on it and save that workspace, *SIGVIEW* will not save the actual values from the signal or values from the FFT result. Only the name of the signal file will be saved along with the information that you performed FFT on it with certain parameters. Therefore, if you change the data in the original signal file and reload *SIGVIEW* workspace, you will get changed signal data and the FFT from it. Generally, a workspace file contains the information you can normally see in a Control Window plus properties for each window including axes properties and zoom info.

Files

SIGVIEW is saving its workspace in a file with extension SWS (*SIGVIEW* WorkSpace). This file is a plain text file with the structure similar to Windows *.INI files. Its structure is quite simple and easy to understand or even to edit manually. You can open SWS file with any text editor and perform some changes if you need it. You can even edit its content automatically from another application to control *SIGVIEW* functionality.

For all loaded signals, *SIGVIEW* saves their full path names in workspace file. When opening that workspace file later, *SIGVIEW* searches for those files on their original location first (for example c:wav). If file does not exist, *SIGVIEW* tries to load the file with the same name from the folder where SWS file is. Therefore, if you want to distribute workspace file with all the signals needed, you just have to be sure that the SWS and signal files will be in the same folder on target computer.

Information about window location and size will also be saved for every window. This information is relative to the size of the main *SIGVIEW* window, so you can be quite sure that the loaded workspace will look the same way in every screen resolution or *SIGVIEW* window size.

It is also possible to create a workspace file without defining exact signal file names in it. When opening such workspace file, *SIGVIEW* will ask you for a file name for each signal file used in this workspace. To create such workspace file, just save it once normally – with file names, then open the SWS file with any text editor program and replace all file names in it (all *FileName=....* keys) with “*FileName=choose”*.

You also use drag 'n drop to drag one or more Workspace files from Windows Explorer and drop them on *SIGVIEW* window. The effect will be the same as if loading these files by using "*Load Workspace...*" menu option.

Using Workspace file as a command line argument

It is possible to give an SWS file as a command line argument when starting *SIGVIEW*, for example:

SIGVIEW32.EXE c:\myworkspace.sws

SIGVIEW will start and open this workspace. It is also possible to define all needed signal file names in the command line as well. You have to create the SWS file where each signal file name is replaced with "#X", where X=1,2,3,..., for example "*FileName=#1".* Then you can start *SIGVIEW* with

SIGVIEW32.EXE c:\Analysis1.sws c:\file1.wav c:\file2.wav

Every appearance of "#1" in the Analysis1.sws will be replaced with "c:\file1.wav", every appearance of "#2" with "c:\file2.wav", and so on....

Saving file names in a Workspace

If you are saving a Workspace (SWS file) containing windows with file-based signals (for example, loaded WAV files), *SIGVIEW* will offer you two options:

1. Save full file names in a file: Each time you load the workspace, your files (if they still exist) will be automatically loaded. The result will be exactly the same as the workspace you saved if those signal files did not change in the meantime.

2. Do not save file name information in a file: Saved workspace will be used as a template for operations on any files. Each time you try to load the workspace, *SIGVIEW* will offer you a file load dialog to choose a file which should be loaded for each window from the workspace which contained file-based signal. This can be used to speed up the analysis you have to perform often on different files, similar to Custom tools.

Examples

Several example workspace files are installed in "Examples" subdirectory of *SIGVIEW's* application data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\) and can be accessed directly through *Help/Examples...* option in the main menu.

3.5. Creating and reusing custom tools

By using *“Save window as custom tool”* and *“Apply custom tool...”* options from the *File* or *Signal tools* menu, you can save parts of your analysis graphs and reuse them later as your custom tools.

Creating new tools

There are two options for saving a tool: First one is to save only one window as a tool with all its properties. For example, if you performed FFT from a signal and changed some FFT properties: smoothing, removing linear trend.... You can save all those settings as a single FFT tool by clicking on FFT window and choosing “*Save window as custom tool -> Window only...”* option from the menu. Save dialog appears where you can define a name for your tool file – for example “MyFFT.swt”. Only the information about the tool type (FFT) and its properties will be saved. Now, you can load any other signal and choose *“Use custom tool>”* from the menu. If you have saved your tool in a default “Sigview” folder, its name will appear in a submenu. Otherwise, choose “From file…” option, find “MyFFT.swt” file and open it. FFT will be performed on your signal, exactly with the properties you saved earlier. The same principle is applicable to all *SIGVIEW* functions including 3D analysis and instruments.

Second option when saving tools is to use “*Save window as custom tool -> Window and its subtree...”.* This option will save active window and all its child windows as a one tool. For example, you can perform FFT analysis, and then use “*Instruments... Maximum position”* function to display dominant frequency from the FFT. If you save the FFT window with its subtree as a new tool, the instrument window will also be saved. If you apply this saved tool to some signal, you will get its FFT with originally saved settings, and an instrument showing the maximum value from the FFT. With this option, you can save very complex tools including dozens of connected windows.

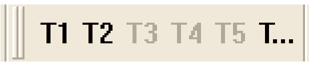
Default folder for saving and loading custom tools is “Tools” folder located under the folder where
SIGVIEW is installed. After you install *SIGVIEW*, there are already few example tools there.

Using tools

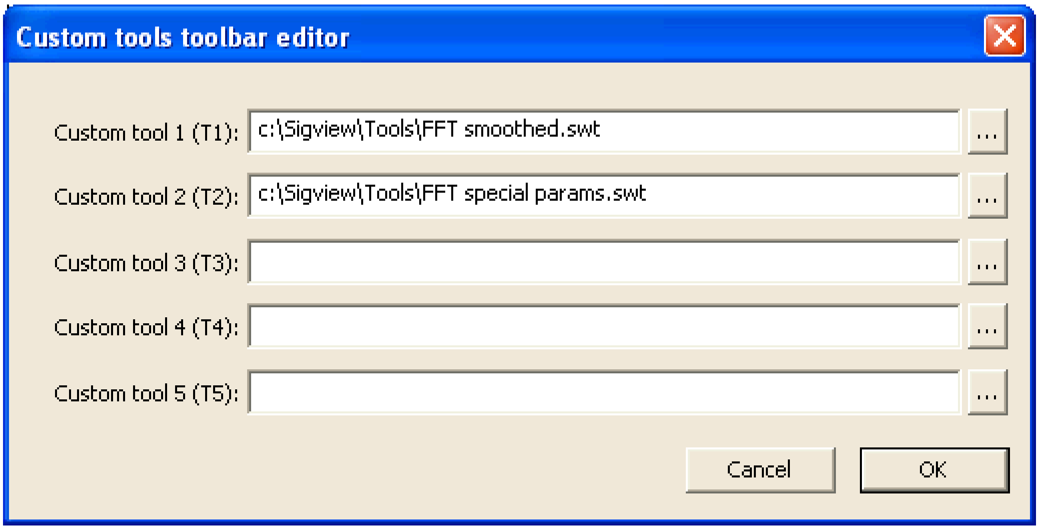
When using a custom tool, *SIGVIEW* will try to check if the tool is applicable to the currently selected window. For example, if you extract a part of some signal between its 5th and 6th second and save that extraction window as a tool, it will not be applicable to the signals with only 3 seconds of length.

You can also use drag 'n drop to drag SWT files from Windows Explorer and drop them in *SIGVIEW*. Those will be applied to the currently active *SIGVIEW* window.

You can also include your custom tools into the *SIGVIEW's* toolbar and assign them to dedicated toolbar buttons. If you click on the "T..." button in the "Custom tools" toolbar part, a dialog will open allowing you to assign a tool (SWT file) to each of 5 custom buttons labeled T1...T5.



After you assign a tool to a button, the button will become enabled and allow you to apply the corresponding tool with a single click instead of searching and opening the SWT file first.



Custom tool 1 (T1): c:\Sigview\Tools\FFT smoothed.swt

Custom tool 2 (T2): c:\Sigview\Tools\FFT special params.swt

Custom tool 3 (T3):

Custom tool 4 (T4):

Custom tool 5 (T5):

Further reading

Also, please see the chapter about Drag 'n Drop in Control Window. It describes some similar alternative functions for reusing parts of the analysis without saving those into SWT files first.

For more detailed examples on custom tools usage, please see How-To example.

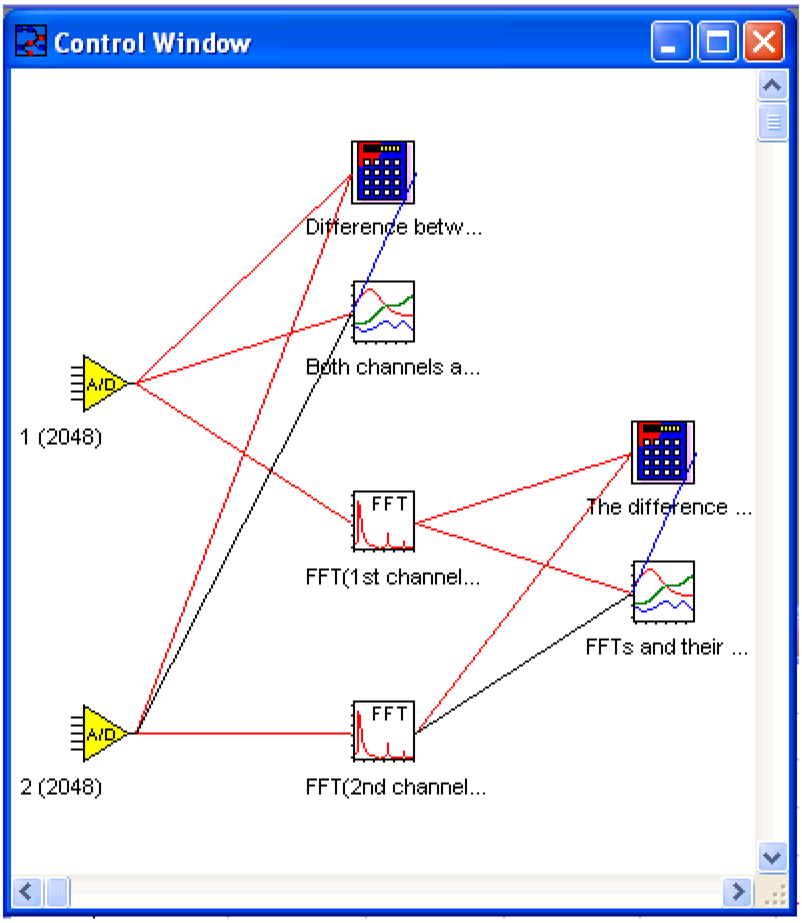
Part IV

Control window

4. Control window

4.1. Control window basics

Control window enables you to work easily with many signals or analysis windows at the same time. Each signal, analysis window, 3D graphics or instrument created while working with *SIGVIEW* is represented with its icon in a Control window.



Icons are connected with solid lines to show a signal flow through the system. A dotted line shows that signal link is not active and changes in the parent will not cause child window to repaint.

Selecting in Control window

You can select or unselect icons by clicking on them with the left mouse button. Every function you perform in the Control window (for example FFT, Time-FFT, Extract from-to,...) refers to selected icons, i.e. windows they represent. For faster selecting or unselecting, click right mouse button in the empty part of the Control window and choose some of the selecting options from context menu (Select all, Unselect all, Invert selection,...).

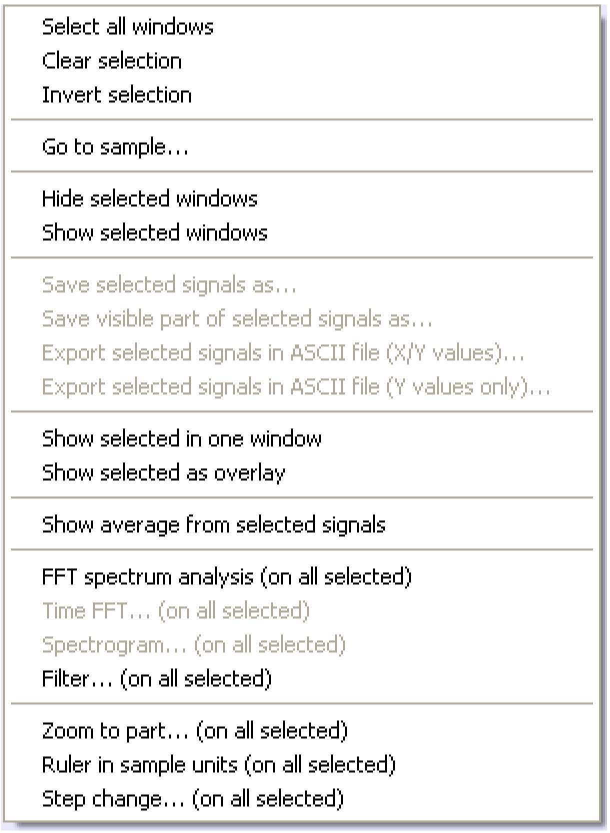
Operations on single windows

Click right mouse button over some icon and a context menu will appear with some important functions for the window it represents. If you choose one function from the menu, the result will be just as if you have chosen that option for the window itself.

4. Control window

Operations on multiple windows

After you select any number of icons you will be able to perform some of the available actions on all of them (if they all support the function). the fastest way to do so is to open a context menu in the empty part of the Control Window. It will show you various applicable functions which can be perform on all currently selected windows.


Select all windows
Clear selection
Invert selection
Go to sample...
Hide selected windows
Show selected windows
Save selected signals as...
Save visible part of selected signals as...
Export selected signals in ASCII file (X/Y values)...
Export selected signals in ASCII file (Y values only)...
Show selected in one window
Show selected as overlay
Show average from selected signals
FFT spectrum analysis (on all selected)
Time FFT... (on all selected)
Spectrogram... (on all selected)
Filter... (on all selected)
Zoom to part... (on all selected)
Ruler in sample units (on all selected)
Step change... (on all selected)

Hiding/Showing windows

Click right mouse button over the empty part of the window and context-menu will appear with Hide and Show options. Choosing one of these options will hide or show all selected windows. Hidden windows will have a "H" letter near its icon in the Control window. This can be helpful if you need some temporary windows in your analysis but you don’t want them to take place on the display.

Overlay signals

If you select more than one icon in the Control window, a two additional options will appear in a context menu: *Show selected as overlay* and *Show selected in one window*. Both operations will create a new window where selected signals will be shown in the same coordinate system (as overlay) or in separate coordinate systems in one window For general information about this feature, see Overlays.

Playing multiple signals simultaneously

You can also use VCR-like functions on multiple signals selected in the Control window. It will allow you to play more signals at the same time and simulate real-time comparison or combination between them. If your sound card supports that feature, you will be able to hear both signals mixed at the same time.

Calculating Average from two or more signals

There is also a special functionality of the Signal averager window when used from the Control window. To use it, select two or more signals in the Control window and choose Signal tools/Averager context menu option. A new signal window will be created, representing average value from all selected signals.

4.2. Linking & unlinking windows

Two windows are "linked" in *SIGVIEW* if changes in one of them influence the changes in the other one. For example, if you calculate an FFT sequence from some part of the signal, that FFT will recalculate and redraw each time you move through the original signal, zoom it in or out, change signal values, ...

All windows except Time FFTs are by default linked to its parent windows. You can disable that link by choosing *System control/Unlink window from parent* menu option or button in a toolbar. You can establish that link again by choosing *System control/Link window to parent menu option* or button in toolbar.

You can see the current state of links between windows in the Control window. If two windows are linked, they will be connected with a solid line. If they are not linked, the line will be dotted. You can also perform linking/unlinking from the Control window's context menu.

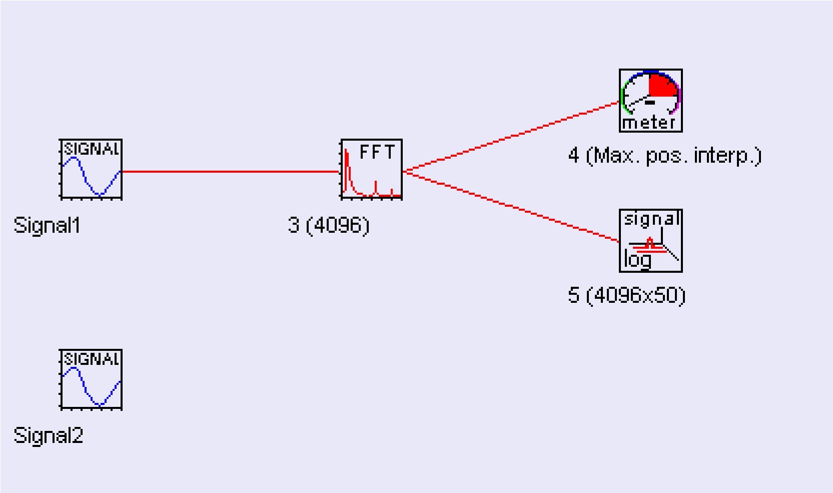
4.3. Drag 'n Drop functions

The fastest way to reuse/copy parts of your current Workspace or to apply an existing analysis sequence to some existing window is to use Drag 'n Drop functions in the Control Window. Here are some usage examples:

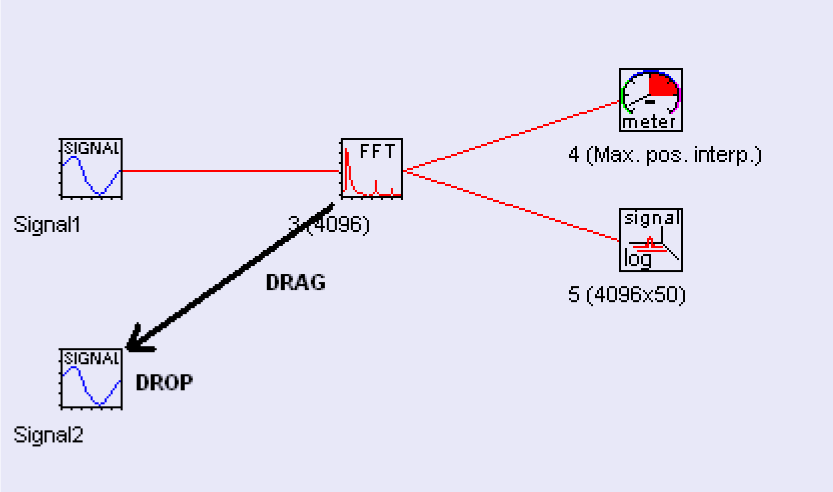
Reusing or moving part of analysis

By dragging an analysis window icon to some other target window icon in the Control Window, the complete analysis tree starting from the dragged window icon will be copied and applied to the target window. If Ctrl-key is pressed, the analysis tree will be moved, otherwise it will be copied. The cursor shown during dragging will indicate the current mode by displaying "COPY" or "MOVE" text next to the graphics.

And here is a simple example. The starting point would be this workspace:

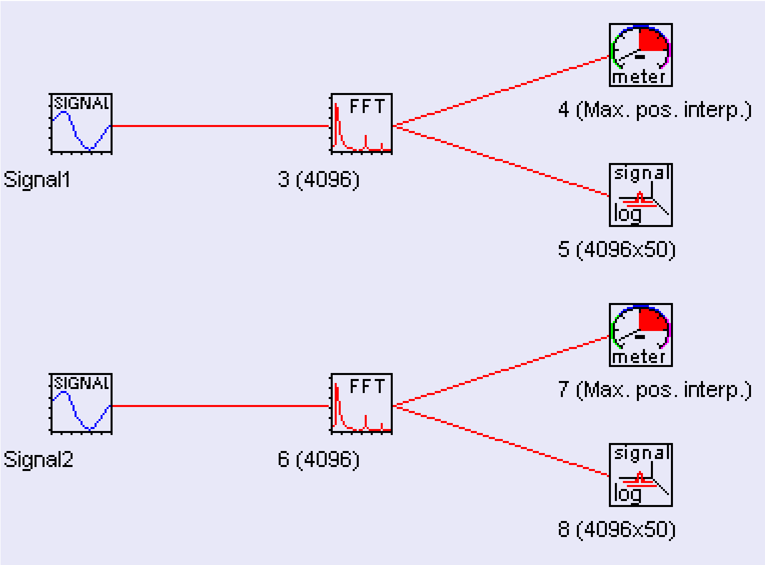


We perform drag operation of the FFT window to the new Signal2:



And the result is that the copy of the analysis is applied to the Signal2:

4. Control window



The current drag context can be recognized on the cursor mouse cursor shape:

COPY Currently dragged window and its subtree will be copied - there is no specific target to apply it to. Applicable only to root windows (without parent).

COPY Currently dragged window and its subtree will be copied and applied to the window icon currently under mouse cursor.

MOVE Currently dragged window and its subtree will be moved - there is no specific target to apply it to. Applicable only to root windows (without parent).

MOVE Currently dragged window and its subtree will be moved and applied to the window icon currently under mouse cursor.

Duplicating existing analysis parts

By dragging the icon of the root window (window without parents) to the empty space in the Control Window, the copy of that window and its analysis subtree will be created.

Dragging Tools file (*.SWT) onto Control window

It is also possible to use drag 'n drop between Windows Explorer and Control Window. If you drag a saved Tool file (*.swt file) from Windows Explorer to the Control Window inside

SIGVIEW, that tool will be applied to all currently selected windows in the Control Window.

Part V

Signal analysis

5. Signal analysis

5.1. FFT

Most of spectral analysis tools in *SIGVIEW* are based on the FFT algorithm. Its main purpose is to transform the signal from its time-domain representation into the frequency-domain representation. For the detailed mathematical description of the algorithm and its interpretation, please refer to the following internet resources:

http://en.wikipedia.org/wiki/Fast_Fourier_transform

http://www.fftw.org/links.html

http://astronomy.swin.edu.au/~pbourke/analysis/dft/

http://www.eptools.com/tn/T0001/INDEX.HTM
……..

And a wonderful web page with a free book on digital signal processing:

http://www.dspguide.com/

By preparing a signal before using the FFT algorithm, changing FFT calculation parameters or post-processing its results, many different variations of the FFT transformation can be applied. One of the most important features in *SIGVIEW* is the possibility to change most of these parameters and easily observe and compare the results.

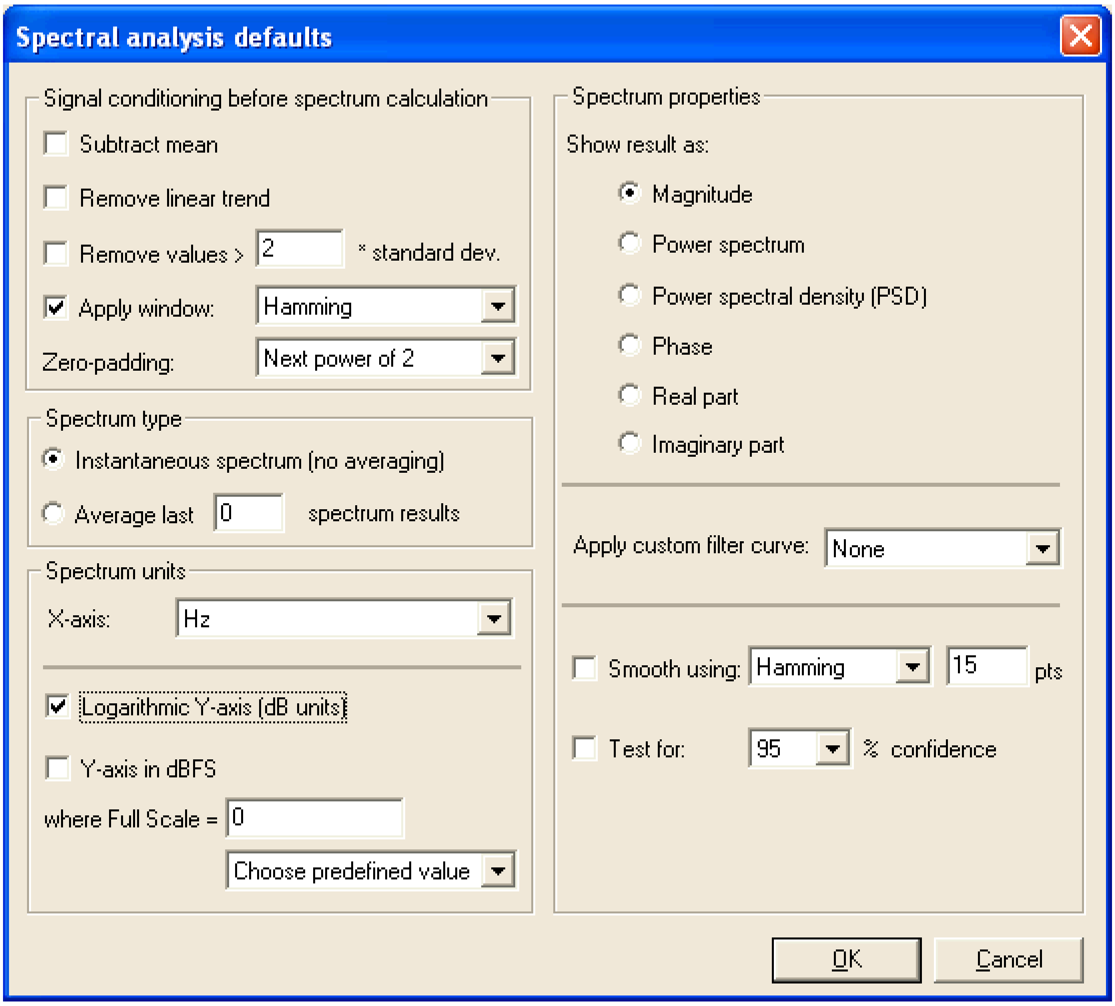
To apply FFT transformation to your signal in *SIGVIEW*, select some window containing a signal and choose *Signal tools/FFT* option. An FFT result window will be created containing the FFT result calculated according to your current Spectral analysis defaults. You can change the parameters for the already calculated FFT sequence anytime by choosing *Edit/Properties* option from the main menu or *Properties* option from the context menu. The dialog for the editing of FFT parameters is the same as the Spectral analysis defaults dialog – you can find the detailed description of its fields here.

5.2. Spectral analysis defaults

Most of spectral analysis tools in *SIGVIEW* are based on the FFT algorithm. There are many different parameters which can be applied to the signal before the FFT analysis is performed, to the FFT calculation itself or its result.

If you want to change the default settings for spectral analysis calculation in *SIGVIEW*, use *Signal tools/Spectral analysis* defaults option from the main menu. These settings will be applied to all new FFT-based calculations, including FFT, Time-FFT, cross spectral analysis, etc.

Once the FFT or Time FFT is already calculated according to your current spectral analysis defaults, you can edit those anytime by choosing *Edit/Properties* menu option for calculated window (FFT, Time FFT,…). The same option is also available in the window’s context menu. These settings will apply only to that single window.



Spectral analysis defaults

Signal conditioning before spectrum calculation

□ Subtract mean

□ Remove linear trend

□ Remove values > 2 * standard dev.

☑ Apply window: Hamming

Zero-padding: Next power of 2

Spectrum type

● Instantaneous spectrum (no averaging)

○ Average last 0 spectrum results

Spectrum units

X-axis: Hz

☑ Logarithmic Y-axis (dB units)

□ Y-axis in dBFS

where Full Scale = 0

Choose predefined value

Spectrum properties

Show result as:

● Magnitude

○ Power spectrum

○ Power spectral density (PSD)

○ Phase

○ Real part

○ Imaginary part

Apply custom filter curve: None

□ Smooth using: Hamming 15 pts

□ Test for: 95 % confidence

The following parameters can be changed for each FFT based spectral analysis operation in *SIGVIEW*:

Subtract mean check box: Select it if you want to normalize the signal before processing. It simply subtracts the mean value of the signal from each sample

Remove linear trend check box: If linear trend appears in signal (i.e. the whole signal raises or falls monotonously), it can affect evaluation of low frequency components in the FFT. This option removes linear trend by subtracting linear least squares approximation from the signal

Remove values > : This option automatically removes values from the signal that are not in range (mean -N*StDev , mean + N*StDev) where StDev is a standard deviation of the signal and N is a user-defined coefficient entered in the corresponding edit box. Values are removed by replacing them with the mean value.

Apply window: To avoid some undesirable effects of discrete Fourier transform, it is recommended to lower the signal values near the end of the signal by multiplying it with some appropriate weighting function. This technique is called "windowing". *SIGVIEW* supports several standard weight functions: Hann, Haming, Blackman, Triangle, Tukey. Choosing "Rectangle" window has the same effect as turning windowing option off.

Zero padding: Due to the nature of the FFT algorithm, the fastest calculation will be performed for signals having power-of-2 length (for example 128, 256, 512,…). If your signal or the part you would like to analyze have some other length, there are several options you can choose:

- **Never** use zero padding: FFT or slower DFT algorithm will be applied on your signal without altering or extending it with zeros. This is the slowest calculation method, but the advantage is that your signal is not changed in any way. For prime number signal lengths the calculation will be performed by using very slow DFT algorithm.
- **Optimal** method: Your signal will be expanded with zeros to the next possible length allowing the usage of the FFT algorithm instead of very slow DFT.
- **Next power of 2**: Your signal will be expanded with zeros until the next power-of-2 length. This will enable the usage of fastest FFT algorithm.

To increase the precision of the spectral analysis without taking longer signal segments, you can also expand existing signal segment with zeros, even beyond the next power-of-2 length. That will not add essentially new information in the FFT result, but will increase its precision, i.e. will reduce the size of the one frequency bin. You will simply get FFT result with more points **-** from the same signal segment. (If signal has 1024 samples and you choose expanding by factor 8, resulting spectrum will have 4096 pt. instead of 512 without using option). The result will be comparable to interpolation of the normal FFT result. To use this feature, you can choose **2x, 4x or 8x** zero padding option.

Spectrum type:

- **Instantaneous spectrum (no averaging)**: Each change in the signal will cause the FFT to recalculate. This is the default behavior.

- **Average last X spectrum results**: This option is useful only for the FFT of the live input signal or other fast changing signals. Last X complex FFTs will be averaged to calculate the final result. That will decrease the influence of the noise in the signal and is usually recommended for all spectral analysis functions on the live signal.

Show result as: After applying the FFT algorithm to the signal, the result will be an array of complex numbers. These radio-buttons determine the way these complex numbers will be converted to real values and displayed as FFT graphics.

- *Magnitude*: value generally regarded as “spectrum”; calculated as sqrt(R^2 + I^2)
- *Power spectrum*: Magnitude squared.
- *Power spectral density (PSD)*: Power spectrum is calculated first and its values are divided with the width of the frequency bin. This “normalization” makes comparisons between FFT sequences from different signals with different sampling rates possible.
- *Phase:* show signal phase angle (in degrees)
- *Real & Imaginary part*: show only real or only imaginary part of the spectrum

Apply custom filter curve: You can apply custom filter curve to change the values of the spectrum. See Custom filter curves section for more information. This option is applicable only to magnitude and power spectrum result types.

Logarithmic Y-Axis (dB units) option can be applied only to Magnitude, Power spectrum and PSD spectrum results. It shows a logarithmic values for all FFT result points, causing Y-axis to be logarithmic.

Y-axis in dBFS: This option can be applied only if Logarithmic Y-Axis is used. It will calculate dB values relative to some maximal, i.e. full scale value. You can freely define this Full-scale value in signal units (for example 32767 for 16-bit sound card or max. voltage for NiDAQ device). There is also a combo-box with the choice of some predefined values for the maximal amplitude. If you set the full scale value to 0, dB values will always be calculated relative to the currently highest FFT value so that the highest value is always equal to 0 dB and all other FFT values are negative.

X-Axis units combo box: allows you to switch between following X-axis units: “Hz (Cycles/sec)”, “CPM (Cycles/minute)”, “KHz”, “MHz”.

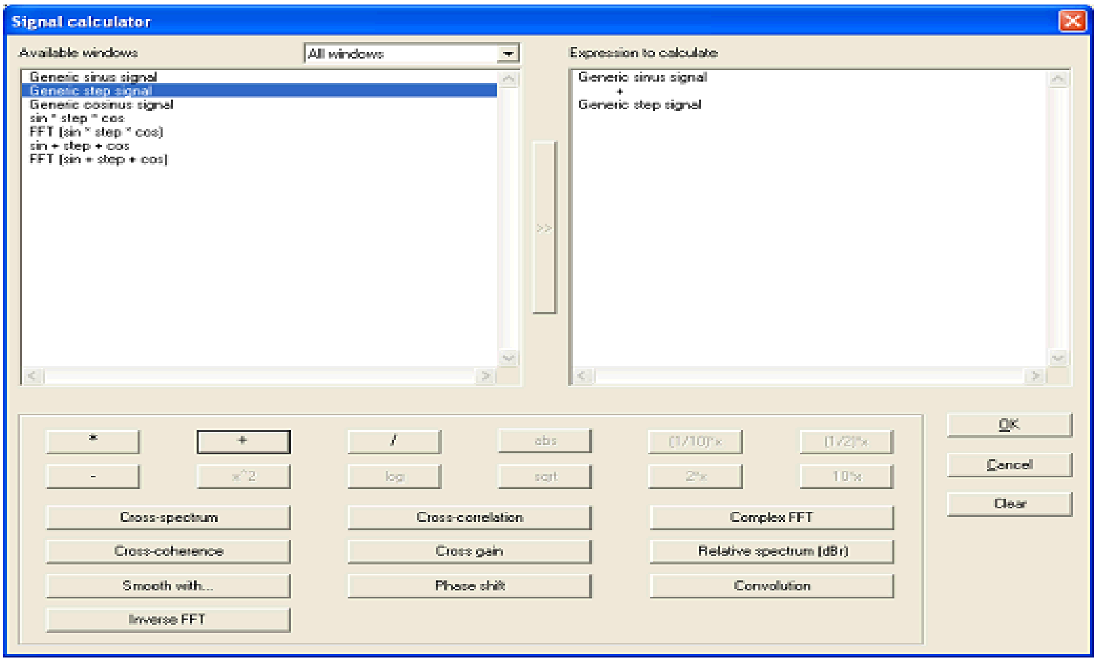
Smoothing: If the result of FFT is noisy and has many values, it can be useful to smooth it by using weighted moving average function. That will remove small noisy details from the FFT and give you a better overview of the important spectrum events. You can choose some of standard weighting functions here and determine the length of a smoothing window (longer window means more smoothing).

Test for confidence: Siegel’s test for confidence of spectral peaks tells you if some peak in spectrum is statistically significant or it could be product of some random fluctuation of the signal. If you turn this option on, only significant peaks will be shown, while all other values in the spectrum will be set to zero. You can choose two levels of confidence for this test: 95% or 99%. This test is purely statistics; it does not use any artificial intelligence methods.

5.3. Signal calculator

By choosing *System control/Signal calculator* option from the main menu, you can start one of the most useful tools in *SIGVIEW*, the Signal calculator.

Just as a standard calculator is used for calculating expressions with numbers, *SIGVIEW*'s calculator can be used to combine signals or instruments in different arithmetic or signal analysis expressions. You can use it, for example, to add or subtract two signals or spectrums or to perform different cross-spectrum calculations.



Signal calculator

Available windows: All windows

Generic sinus signal
Generic step signal
Generic cosinus signal
sin * step * cos
FFT (sin * step * cos)
sin + step + cos
FFT (sin + step + cos)

Expression to calculate

Generic sinus signal
+
Generic step signal

In the upper left box, the names of all existing signal or instrument windows (loaded signals, FFT results, etc.) in your *SIGVIEW* workspace will be shown. You can choose to display all windows, only instruments or only signal-like windows (containing sequences of values).

The list box on the right side will contain the new expression for the evaluation.

At the bottom of this dialog, various buttons with all available calculator functions are shown. Only functions which are available in the given context will be enabled. For example, only after you add a signal window to your expression, a binary function buttons like "+", "-", "/",... will be enabled.

You can move signals from the left to the right box (include them in the expression) by double-clicking their names or selecting them and pressing the '>>' button. For example, to subtract two signals, simply select the name of the first one, press '>>' button to add it to the expression, then press '-' button, and finally add the second signal to your expression.

Important note: Longer arithmetic expressions will be calculated in the order listed in the expression box and not according to the mathematical operator precedence. For example, the expression **sqrt(A) + log(B) * C** will be calculated as **(sqrt(A) + log(B)) * C.** The expression **A+B*C+D+E*F** will be calculated as **((((A+B)*C)+D)+E)*F.**

5. Signal analysis

When you create the complete signal expression, click OK to create the result signal. It will be linked to all windows included in the expression and will recalculate and redraw each time they change. In case that one of expression windows is deleted, the calculator window will be frozen and will keep its last values.

Unary functions (can be applied to the on signal or instrument window)

- **sqrt** (square root)
- **log** (log10)
- **x^2** (squared)
- **abs** (absolute value)
- **2*** (multiply with 2)
- **10*** (multiply with 10)
- **(1/2)*** (multiply with 1/2 i.e. divide with 2)
- **(1/10)*** (multiply with 1/10 i.e. divide with 10)

Binary functions (can be applied to two signal or instrument windows)

- ***** (multiply)
- **/** (divide)
- **+** (plus)
- **-** (minus)

- **Cross-correlation**
- **Convolution** (convolutes first signal with the second one)
- **Smooth with** (smooth first signal by using the second one as a weighting function)
- **Cross spectrum**
- **Cross coherence**
- **Cross gain**
- **Phase shift**
- **Relative spectrum (dBr) Complex-FFT**
- **Inverse-FFT**
- **Complex-FFT**

If only a simple arithmetic binary functions (*, /, +, -) are used, you can freely mix instrument and signal windows in the expression.

If you include two signals of different lengths in a binary operation (for example you subtract one signal from another one), *SIGVIEW* will use only a part of the longer signal (starting from its first sample) which has the length of the shorter signal.

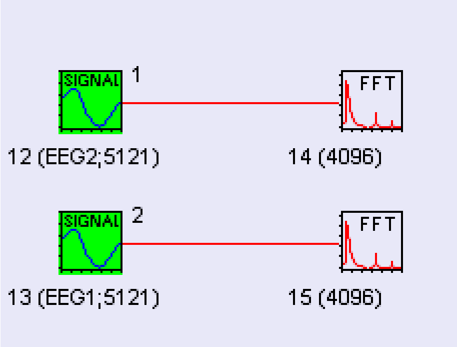
Examples

Here are some examples of calculator expressions.

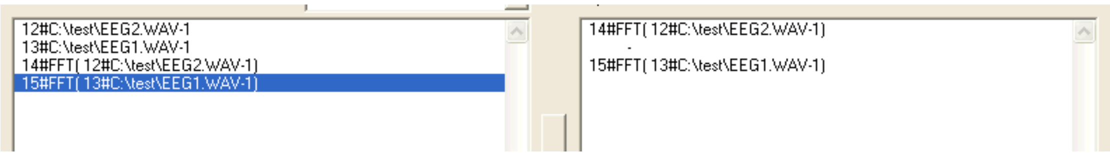
5. Signal analysis

1. Simple arithmetic operation on spectrums.

You have loaded two signals, calculated their spectrums and you have the following workspace:



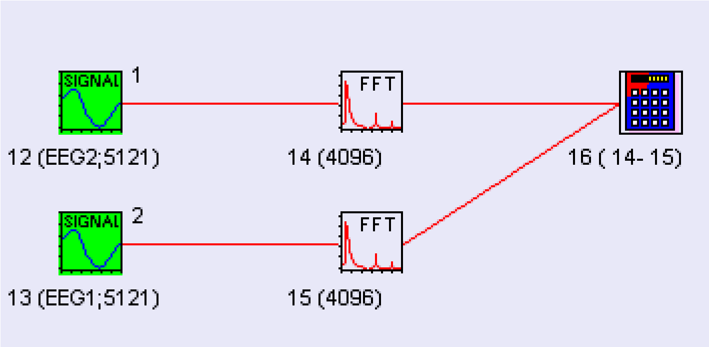
Now, you would like to subtract two calculated spectrums to get some information about their differences. To do it, create the following expression in the Signal calculator:



```
12#C:\test\EEG2.WAV-1
13#C:\test\EEG1.WAV-1
14#FFT( 12#C:\test\EEG2.WAV-1)
15#FFT( 13#C:\test\EEG1.WAV-1)
```

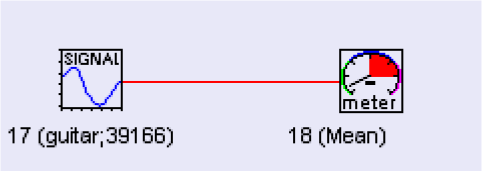
```
14#FFT( 12#C:\test\EEG2.WAV-1)
-
15#FFT( 13#C:\test\EEG1.WAV-1)
```

The resulting window will contain th requested difference between spectrums.

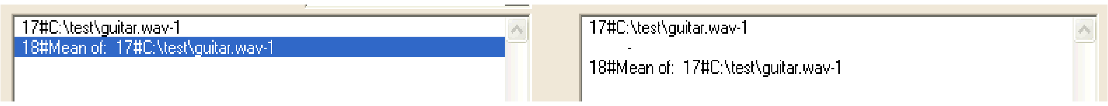


2. Combining instruments and signals in one expression

You would like to subtract mean value of a signal from all signal values (normalization). First, you would calculate signal mean by using the appropriate instrument:



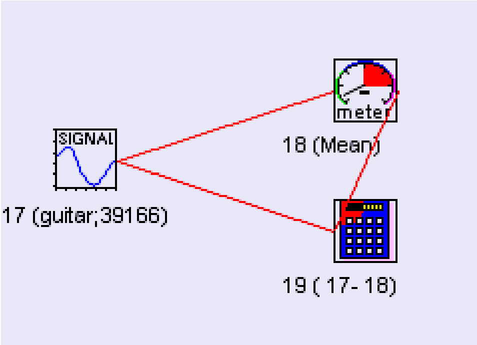
Then, you would create the following expression in the Signal calculator:



17#C:\test\guitar.wav-1
18#Mean of: 17#C:\test\guitar.wav-1

17#C:\test\guitar.wav-1
-
18#Mean of: 17#C:\test\guitar.wav-1

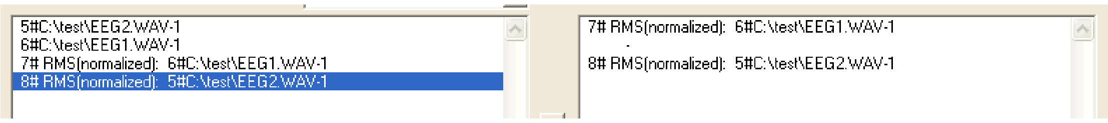
The result will be a new signal calculated as requested:



2. Instrument-only expressions

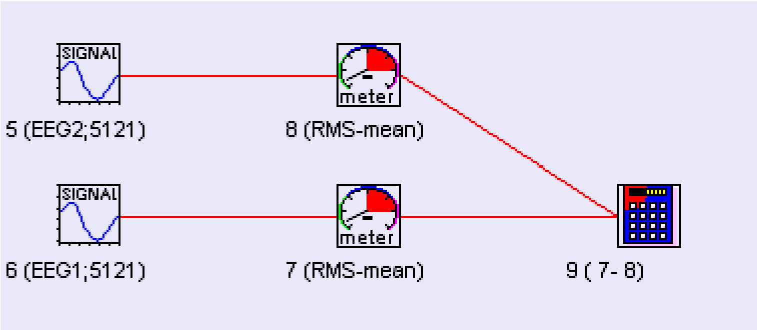
You can also have expressions containing only instrument windows. The result of such expressions will also be a new instrument window.

In this example, we calculate RMS of two signals and then create a new instrument containing their difference:



5#C:\test\EEG2.WAV-1
6#C:\test\EEG1.WAV-1
7# RMS(normalized): 6#C:\test\EEG1.WAV-1
8# RMS(normalized): 5#C:\test\EEG2.WAV-1

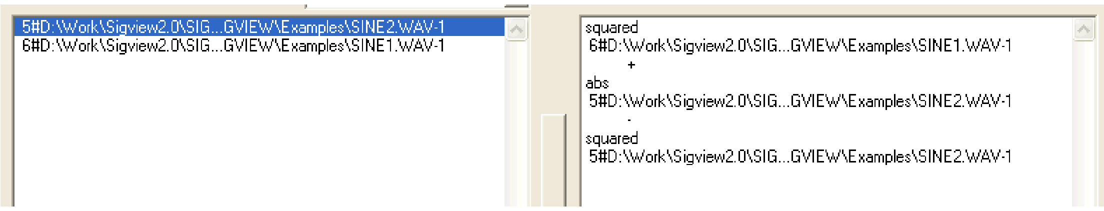
7# RMS(normalized): 6#C:\test\EEG1.WAV-1
-
8# RMS(normalized): 5#C:\test\EEG2.WAV-1



3. Complex expressions

There a practically no limitations for creating calculator expressions. Here are some examples:

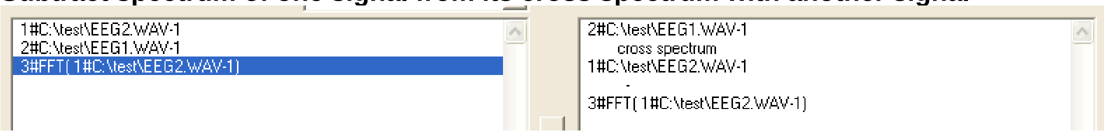
f = A^2 + abs(B) - B^2



```
5#D:\Work\Sigview2.0\SIG...GVIEW\Examples\SINE2.WAV-1
6#D:\Work\Sigview2.0\SIG...GVIEW\Examples\SINE1.WAV-1
```

```
squared
 6#D:\Work\Sigview2.0\SIG...GVIEW\Examples\SINE1.WAV-1
          +
abs
 5#D:\Work\Sigview2.0\SIG...GVIEW\Examples\SINE2.WAV-1
          -
squared
 5#D:\Work\Sigview2.0\SIG...GVIEW\Examples\SINE2.WAV-1
```

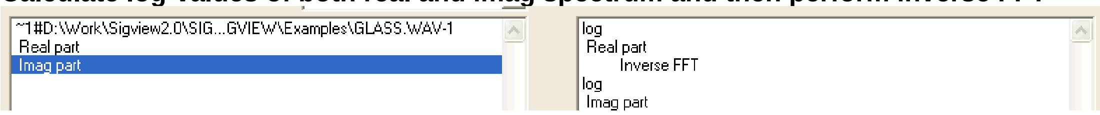
Subtract spectrum of one signal from its cross spectrum with another signal



```
1#C:\test\EEG2.WAV-1
2#C:\test\EEG1.WAV-1
3#FFT( 1#C:\test\EEG2.WAV-1)
```

```
2#C:\test\EEG1.WAV-1
        cross spectrum
1#C:\test\EEG2.WAV-1
          -
3#FFT( 1#C:\test\EEG2.WAV-1)
```

Calculate log-values of both real and imag spectrum and then perform inverse FFT



```
~1#D:\Work\Sigview2.0\SIG...GVIEW\Examples\GLASS.WAV-1
 Real part
 Imag part
```

```
log
 Real part
         Inverse FFT
log
 Imag part
```

5.4. Autocorrelation

Accessible through Signal tools/Autocorrelation menu option.

Autocorrelation is a tool used frequently for the analysis of time domain signals. It is the cross-correlation of the signal with itself. Autocorrelation is useful for finding repeating patterns in a signal, such as determining the presence of a periodic signal which has been buried under noise, or identifying the fundamental frequency of a signal which does not actually contain that frequency component, but implies it with many harmonic frequencies.

The same function can be performed in Signal calculator by performing the cross-correlation of the signal with itself.

5.5. Filters

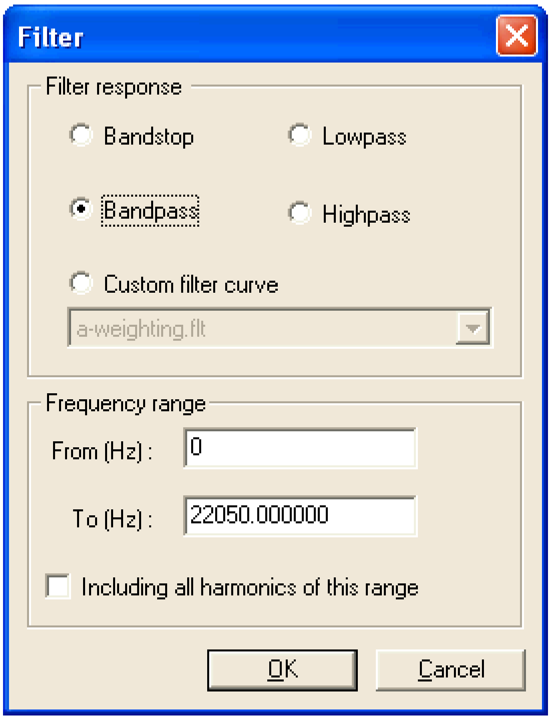
Filtering is the operation which removes or changes specific frequency components from the signal. In *SIGVIEW*, you can simply apply a filter to the signal by defining frequency segment to be removed, or the segment you want to leave in the signal. That way, you can create bandstop, bandpass, highpass and lowpass filters.

When you choose *Signal tools/Filters* option from the main menu, a dialog appears where you can enter segment boundaries (in Hz) and determine if you want to remove that frequency segment (bandstop), leave only that frequency segment (bandpass) or you want to remove all frequencies up to (highpass) or above some frequency (lowpass).

For Bandstop and Bandpass filter types, you can also include all higher harmonics of defined frequency segment. If the defined segment is [x,y], it will also include all segments [N*x, N*y], where N=2..Nmax.

There is also an option to use custom filter curves for signal filtering. For details, please see Custom filter curves section.

After pressing OK, a new window with filtered signal will appear. You can change filter’s frequency boundaries later by choosing *Edit/Properties* on filtered signal window from the menu. You will be able to see instantly how these changes affect the filtered signal.



Filter

Filter response

○ Bandstop ○ Lowpass

● Bandpass ○ Highpass

○ Custom filter curve

a-weighting.flt

Frequency range

From (Hz) : 0

To (Hz) : 22050.000000

□ Including all harmonics of this range

SIGVIEW uses FFT-based filtering algorithm. It is a combination of the FFT calculation, changes in the FFT result (filling some frequency bands with zeros or applying custom filter curve) and the inverse FFT calculation. This method is relatively slow compared to other filtering methods, but is usually very accurate.

5.6. Signal averager

Signal Averager is a signal that calculates average values of all past changes in its parent signal.

For example, try to make Averager window of some existing spectrum window. When Averager is created, it contains exactly the same data as the current spectrum. Now, start your analysis system and make spectrum values change. Averager ‘catches’ all changes of the spectrum and calculates their average. This way, you will get a spectrum average through time. You can apply this option to any kind of signal.

If you want to reset all averager values and start with the averaging again, choose *Reset data* from its context menu.

If you want simply to stop or start averaging process, choose unlink or link option from the main menu or toolbar.

5.7. Resampling signals

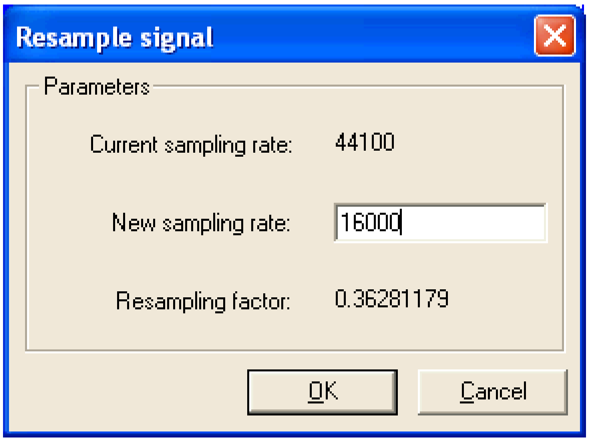
Accessible through Signal tools/Resample menu option.

Resampling is a process of changing sampling rate of an existing signal. A new signal should keep as much as information contained in the original signal as possible.

A resampling dialog in *SIGVIEW* lets you choose a new sampling rate for the signal and shows you a resampling factor (a ratio between new and old sampling rates). This factor must be in the range [0.1, 10], i.e. you can increase (up-sampling) or decrease (down-sampling) sampling rate by the factor of 10.

For some combinations of the current and new sampling rates, *SIGVIEW* will not be able to achieve exactly the requested new sampling rate. For example, when converting 44.1 kHz signals to 48 kHz signals, *SIGVIEW* will generate a signal with the sampling rate of 48000.00114 Hz. This is caused by the underlying algorithm.

As all *SIGVIEW* functions, resampling is performed only on visible part of the origin signal. Any change in the origin signal, for example zooming in/out will cause the resampled signal to be recalculated.



Resample signal

Parameters

Current sampling rate: 44100

New sampling rate: 16000

Resampling factor: 0.36281179

5.8. Remove DC offset

Accessible through Signal tools/Remove DC offset menu option.

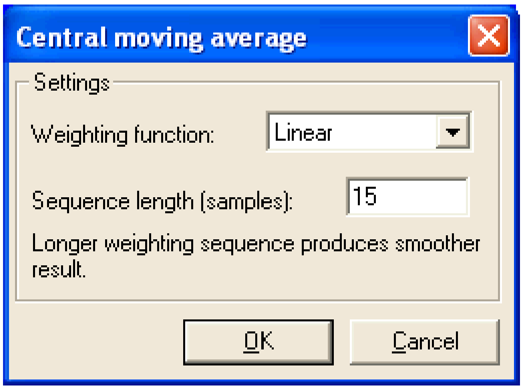
This option normalizes your signal by subtracting signal's mean value from all signal's amplitude values. The result is a signal with the mean equal to zero.

This operation should always be performed before using *Signal tools/Integrate* option.

5.9. Moving average smoothing

Smoothing is the operation that can very often help to extract some useful information from a noisy signal or indistinct FFT sequence. The overall effect of smoothing is similar to filtering with the lowpass filter.

In *SIGVIEW,* smoothing is performed by using weighted central moving average method. In Smoothing dialog you can determine the type of weighting function to be used and its length. Longer function will provide stronger smoothing effect on the signal.


Central moving average

Settings

Weighting function: Linear

Sequence length (samples): 15

Longer weighting sequence produces smoother result.

If you want this operation to be performed on each FFT result after its calculation, you can turn it on in Spectral analysis defaults .

You can change the smoothing parameters for a smoothing window if you open *Edit/Properties* menu option from its context menu.

5.10. Removing linear trend

If linear trend appears in the signal (i.e. the whole signal raises or falls monotonously), it can affect the calculation of low frequency components in the FFT. With the help of *Signal tools/Remove linear trend* option, you can transform the origin signal to a new signal without linear trend. That signal can be used for further analysis. Linear trend will be removed by subtracting signal's linear least squares approximation from the original signal.
If you want this operation to be performed before each FFT calculation, you can set it in Spectral analysis defaults .

5.11. Cross-spectral analysis

Cross-spectral analysis gives you information about relations between two signals in their frequency domain. All cross analysis functions are accessible through Signal calculator . Several functions are available:

Cross-correlation: measure of similarity of two signals, commonly used to find features in an unknown signal by comparing it to a known one. It is a function of the relative time between the signals. It can also be used to detect similarities in possibly time-delayed signals in applications like echo detection etc. Y-axis units of the result are not normalized, i.e. they depend on original signal amplitudes. Please note that amplitude values of the cross-correlation result are not fully normalized, i.e. maximal correlation will not have a value of 1. The result of the cross correlation function will show only one half of the common cross correlation plot (positive or negative depending on the order of signals used in expression). To see the other side, simply switch the order of signals used in your expression (B cross corr. A instead of A cross corr. B).

Cross spectrum: product of spectrums for two signals. Similarities in their frequency domains will be emphasized.

Cross coherence: measure of correspondence between spectrums for two signals. Y axis units are normalized to [0,1]. The value of 1 means that this frequency component is very similar in both signals, the 0 means that there is no similarity.

Cross gain: measure of contribution of signal1 frequency component in cross spectrum of signal1 and signal2. Higher amplitude means that the first signals contributes more in the cross spectrum for that specific frequency. To take full advantage of this analysis you should perform both “signal1 cross gain signal2” and “signal2 cross gain signal1”

Phase shift: phase shift between signal1 and signal2 on different frequencies. Y axis units are normalized to [-180,180] degrees.

Relative spectrum (dBr): Relative spectrum relation between two signals calculated as 20*log(spectrum1/spectrum2). By using this function you can calculate spectrum of the signal relative to some reference signal, for example microphone input relatively to microphone characteristics spectrum.

5.12. FFT on selected signal segments

It is sometimes useful to extract only a ‘good’ parts of the signal and perform your analysis only on those parts. The manual extraction of those parts, analysis on each of them and combination of the results can be a very demanding job. *SIGVIEW* includes one option to make this job easier enabling you to select ‘good’ segments from the signal and to perform FFT analysis on them.

Choosing segments from the signal can be controlled through the options in signal’s context menu (right-click menu), under "Segments" section:

Start segment here: marks the beginning of the new segment on the current signal ruler position

End segment here: marks the end of the new segment on the current signal ruler position

Delete this segment: Deletes the segment currently under a signal ruler

Clear all segments: clears all selected segments from the current signal

FFT on selected segments: performs FFT on selected segments. This specific FFT implementation performs FFT on overlapping segments inside each segment, calculates average of all FFT results and displays the result in a new window.

Selected segments will be marked with green color behind the signal display. The beginning of a new (not finished) segment will be shown as a thin green line.

Each signal with segments can be saved in file and reloaded with full segments information by using “Save workspace…” or “Save as tool…” SIGVIEW option.

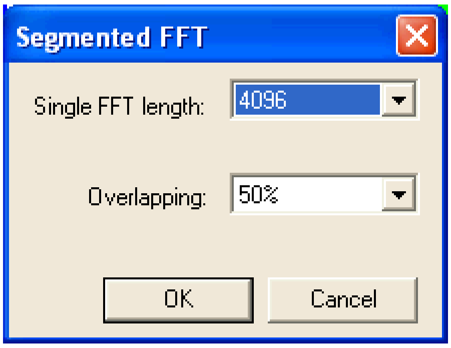
If a new segment overlaps with existing segments, SIGVIEW will simply merge both segments into a one bigger segment. The same behavior is implemented if newly selected segment contains several existing segments.

It is not possible to edit segment boundaries. You have to delete the segment and to define new segment with new boundaries.

Setting FFT parameters and performing FFT

SIGVIEW performs FFT on selected segments by calculating averaged FFT from smaller overlapping blocks in each segment, and finally averaging all results from all segments into one resulting FFT sequence.

After selecting all segments you would like to include in your analysis and choosing “**FFT on selected segments**” option from context menu, dialog will appear with following fields:



Single FFT length: 4096

Overlapping: 50%

FFT length: length for overlapping FFTs

Overlapping: 25%, 50%, 75%, None

FFT length field will include all possible FFT lengths considering the length of a smallest selected segment in a signal (up to 131072 samples). Please note that FFT will be calculated for all samples from each of selected segments, even if overlapping needs to be slightly adjusted. It means that if you select Overlapping=50%, it could be changed to 48% or 45% for certain segment to fit integer number of FFTs into the segments.

FFT window created with this option behaves exactly as any other *SIGVIEW* FFT window. You can open its properties, set all spectral analysis parameters, or save it to by using “Save workspace…” or “Save as tool…” options.

Each time you change segment selection in a signal, FFT will be recalculated and new graphics will be shown.

You can create many FFT windows from the same signal segments at the same time but with different FFT settings. This can be useful to test how different parameters affect the result.

5.13. Probability distribution curve

Accessible through menu option *Signal tools/Probability distribution curve*.

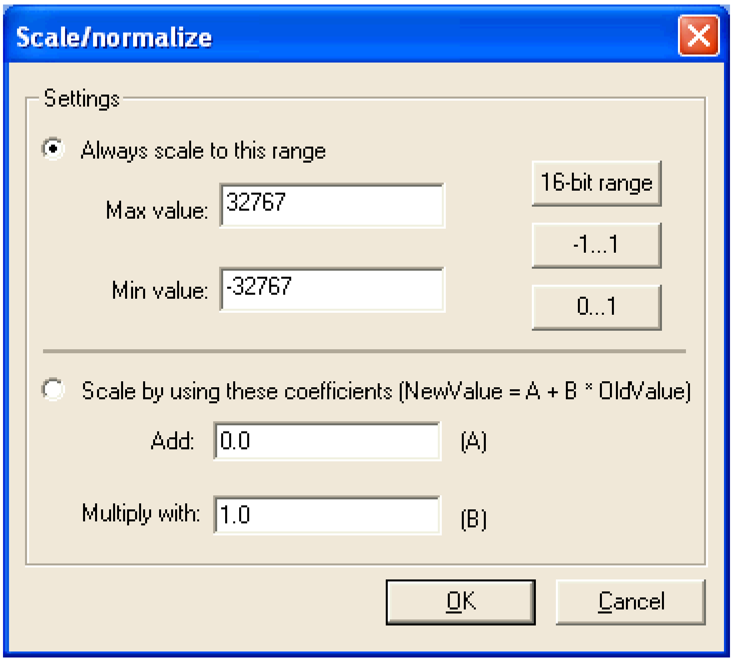
Probability distribution function divides signal's amplitude range in consecutive segments and calculates the probability for a signal value to fall into a certain segment. The resulting curve displays segments (i.e. their beginning values) at the X-axis and probability in [0,1] range on Y-axis. *SIGVIEW* automatically divides amplitude range of the origin signal into segments, where segment count is between 10 and 128.

To interpret the results of this function, the resulting curve can be compared with some standard distribution curves like normal or uniform distribution.

5.14. Scale/Normalize

Accessible through the menu option *Signal tools/Scale/Normalize*

You can use this function to scale signal values to some defined interval or to modify signal values by shifting or multiplying with a constant value.



Scale/normalize

Settings

● Always scale to this range

Max value: 32767

Min value: -32767

○ Scale by using these coefficients (NewValue = A + B * OldValue)

Add: 0.0 (A)

Multiply with: 1.0 (B)

First you have to choose which type of signal modification you would like to perform:

- **Scaling to a new range**: signal values will always be scaled to the range defined by Min/Max values you defined. It means that the current maximum value from the origin signal will become your “Max. value”, minimal value from the origin signal will become your “Min. value” and all other signal values will be scaled in between accordingly. There are few buttons to set some predefined value ranges without typing the values manually.

- **Modifying signal values** by using y = A + B*x equation: You can define A and B values and each value from the origin signal will be modified by using above equation.

5.15. Integral curve and Differentiate options

Accessible through menu option *Signal tools/Integral curve and Signal tools/Differentiate*

Integral curve

This option enables you to see how the integral of the signal changes through time. The result will be a new signal curve where n-th value in this curve will represent the integral of the origin signal from its beginning until its n-th sample.

If you need only to calculate a simple numerical integral of the signal (area under the signal curve), please use the corresponding option from *Instruments and markers* menu*.*

Differentiate

This option calculates the first derivation of the signal by subtracting consecutive signal values from the origin values and forming new curve from subtraction results. It means that the n-th value in new curve will be equal to the difference between n-th and (n-1)-th value from the origin signal. This is only a rough approximation of the first derivation and should not be used on fast changing signals. Before applying this function to such signal, you may try smoothing it first by using appropriate Smoothing option.

5.16. Sorted values curve

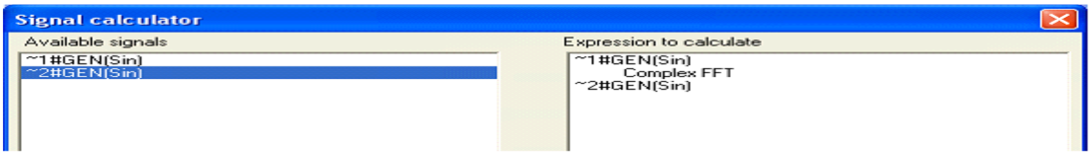
Accessible through menu option *Signal tools/Sorted values curve*

This option creates a new signal with the same length as origin signal, contain origin signal samples sorted in ascending order. For example, if origin signal included values 3, 4, 1, 6 sorted values curve will contain 1, 3, 4, 6. The purpose of this function is to give you a better understanding of the value distribution in the origin signal.

5.17. Complex FFT

Standard FFT function in *SIGVIEW* operates on real-valued sequences, i.e. real signal. If your signal is complex, i.e. contains complex samples, you can still calculate its FFT by using Complex-FFT function in Signal calculator . Good examples of such signals are IF-signals used in radio communication.

Before applying this function, you need to load both real and complex part of your signal as separate signal windows in *SIGVIEW*. Alternatively, you can perform 2-channel data acquisition where the first channel contains real values and the second one complex values. Afterwards, simply open Signal calculator window, choose real signal component, then press Complex FFT button, and then finally choose your imaginary signal component. The result should look like this:



~1#GEN(Sin)
~2#GEN(Sin)

~1#GEN(Sin)
Complex FFT
~2#GEN(Sin)

Press OK button in Signal calculator dialog and a new window containing complex FFT will appear. As expected, its frequency axis will contain both negative and positive frequencies.

If your real and complex signals have different lengths, *SIGVIEW* will use only a part of the longer signal (starting from its first sample) which has the length of the smaller signal.

5.18. Peak and Min Hold

Peak hold is a signal which stores maximum Y values for each X value from all past changes in the origin signal. *Min hold* is a similar function which stores minimum Y values.

Each time original signal changes, peak hold window will compare its new values with the old ones, and take the bigger ones. For example, you can try to create Peak-hold window of some existing spectrum window. When you calculate Peak-hold from it, it will contain exactly the same data as the spectrum in that moment. Now, start your signal analysis system and make spectrum change its values. Peak hold will store maximum values for all frequencies from spectrum. You can apply this option to any kind of signal.

If you want to reset all *peak hold* values and start from the beginning, choose *Reset data* from his context menu or press F3 button.

If you want simply to stop or start peak hold process, choose *unlink* or *link* option from the main menu or toolbar.

5.19. Time FFT statistics

Time FFT gives you many information about the changes in signal's frequency domain, and it is sometimes very useful to summarize that information. Time FFT statistics enables you to divide Time FFT result into different frequency segments and to calculate, display and transfer summarized statistics reports to other programs.

By choosing *3D Tools/Time FFT statistics from* menu, a dialog for statistics data display will appear. On the left side, 5 pairs of edit-boxes (From - To) are available which show 5 possible frequency segments. You can enter segments boundaries here and press *Calculate* button.

The calculation report will appear in big edit-box showing you max, min, sum, avg. values for each segment. By pressing *Clipboard* button, the complete report will be transferred to clipboard, so you can transfer it to some word processing program or Microsoft Excel.
If you use Time FFT statistic often, you can create templates for segments that you use, so you would not have to type them every time. You can do it by creating (or editing existing one) file called TFFTSTAT.RNG in S*IGVIEW's* application data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\). This file has the following structure:

```
from1-to1;from2-to2;from3-to3;from4-to4;from5-to5;Name1
from1-to1;from2-to2;from3-to3;from4-to4;from5-to5;Name2
................
```

where each row in a file represent a new template, fromX, toX are segment boundaries, NameX is name of template which will appear in template combo-box. Initially, this file has only one row which represent an empty template.

```
0-0;0-0;0-0;0-0;0-0;Empty
```

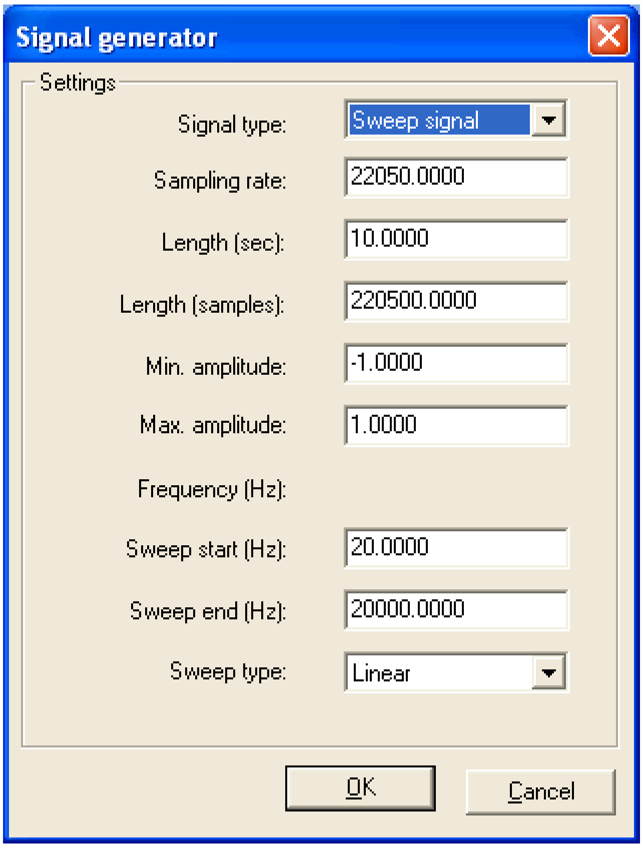
You can edit this file with any text editor.

If the file exists, its first row will be used as a default template as soon as you open Time FFT statistics window. Afterwards, you can switch to other templates by choosing them from Template combo-box.

5.20. Signal generator

If you need to generate artificial signal by using some standard mathematical function, use *Signal tools/Signal generator* option from the menu. You can choose the type of the function (sine, step, white noise,), signal length, frequency, sample rate, min & max values... *SIGVIEW* will allow only input of values which make sense for the chosen signal type (for example there is no frequency for a white noise signal) and disable all other fields.

After you set all signal parameters, click OK and a window with defined signal will be created. You can work with this signal as with any other signal or you can save it in a file for later use. If you would like to use this signal for sound output, simply choose *Play&navigate /Play signal (with sound)* option from the main menu. Choose *Repeat forever* menu option if you would like to play the signal continuously.



Signal type: Sweep signal
Sampling rate: 22050.0000
Length (sec): 10.0000
Length (samples): 220500.0000
Min. amplitude: -1.0000
Max. amplitude: 1.0000
Frequency (Hz):
Sweep start (Hz): 20.0000
Sweep end (Hz): 20000.0000
Sweep type: Linear

Following signal types are supported by the signal generator:

- **Sine, Cosine**: "pure frequency" signals generated by using corresponding mathematical functions
- **Step**: Also known as "square wave". Changes its amplitude from minimal to maximal value according to frequency setting.
- **White noise**: Random signal with a flat distribution of random values
- **Pink noise**: Also known as 1/f noise. Random signal where each octave carriers an equal

amount of noise power, i.e. signal power falls as frequency raises.

- **Sawtooth**: This signal ramps linearly from minimal to maximal value and then sharply drops to the minimal value again. Its waveform resembles the teeth on the blade of a saw.

- **Triangular**: Similar as sawtooth signals but also falls linearly from maximal to minimal value. Each period forms a triangle form.

- **Constant:** All signal vales are the same

- **Gaussian noise**: Statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution). In other words, the values that the noise can take on are Gaussian-distributed.

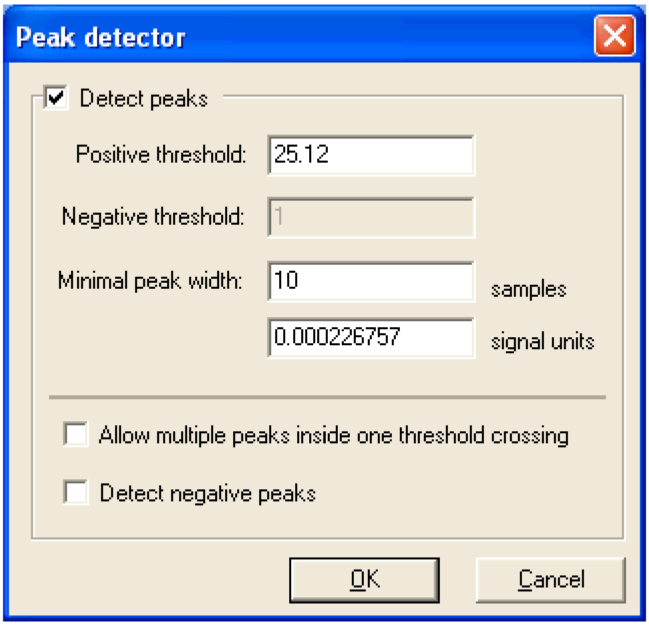
- **Exponential noise**: Statistical noise that has a probability density function of the exponential distribution. In other words, the values that the noise can take on are Exponential-distributed.

- **Sweep signal**: It is a periodic (sine-like) signal which changes (sweeps) its frequency from defined start to end frequency during signal duration There are two types of sweep signals: **linear** sweeps through frequency range at linear speed and **exponential** speeds up exponentially during sweeping so that lower frequencies has longer duration than higher frequencies.

5.21. Peak detection

Peak detection function is used to automatically detect peaks in a signal or in analysis result. It can be started for each signal window from its context menu (right mouse click) or from main menu, *Signal tools/Peak detection* option.

After choosing peak detection option, the following dialog appears:


Peak detector

☑ Detect peaks

Positive threshold: 25.12

Negative threshold: 1

Minimal peak width: 10 samples

0.000226757 signal units

□ Allow multiple peaks inside one threshold crossing

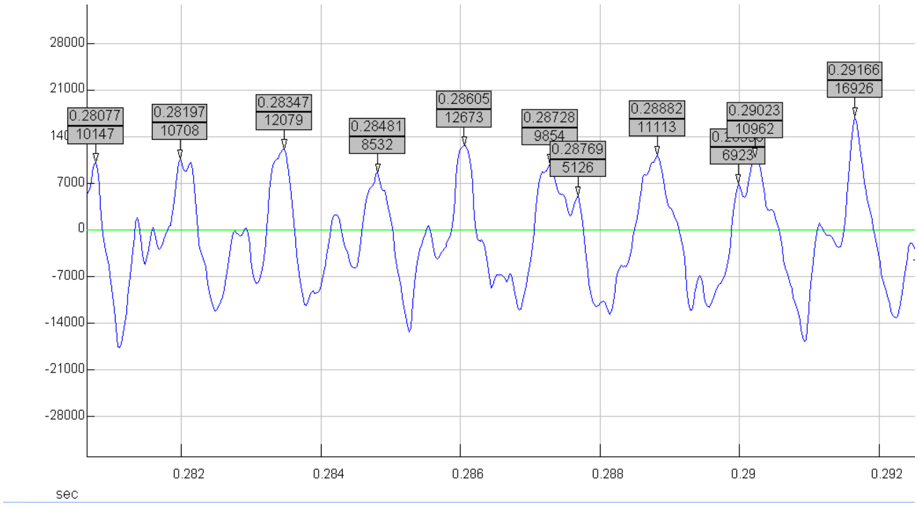
□ Detect negative peaks

To turn peak detection on for the current window, turn on the **Detect peaks** check box as shown in above image.

Following settings can be made for peak detection:

- Absolute amplitude threshold can be set in **Positive threshold** field. It means that Sigview will consider only amplitudes over that value as possible peak candidates. Default value will be set to 75% of the maximal signal amplitude.
- If you would also like to detect negative peaks, turn **Detect negative peaks** check-box on and set corresponding threshold in the **Negative threshold** field. The value should be entered as absolute amplitude so you do not need a minus sign.
- **Minimal peak width** can be defined in samples or signal units (for example Hz or seconds). Setting this property to a higher value will prevent detection of very small peaks where only few values cross the threshold limit.
- **Allow multiple peaks inside one threshold crossing** option allows *SIGVIEW* to detect local peaks inside one threshold crossing. Turning this option on will usually result in increase of detected peak count.

After turning peak detection on and closing the dialog, *SIGVIEW* will place markers on all detect peaks in the visible parts of the signal. It will look like this:



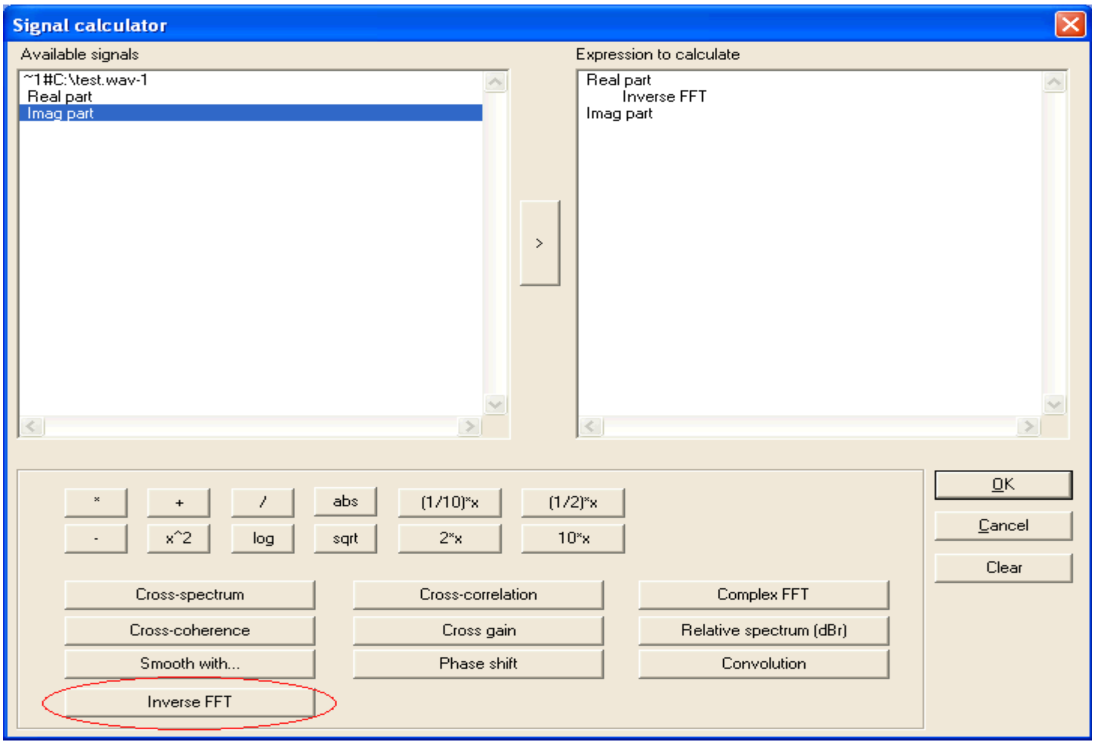
Each marker shows X-position of the peak (above) and peak amplitude (below).

By using Peak count instrument, you can automatically display or track number of peaks currently detected in the visible signal part.

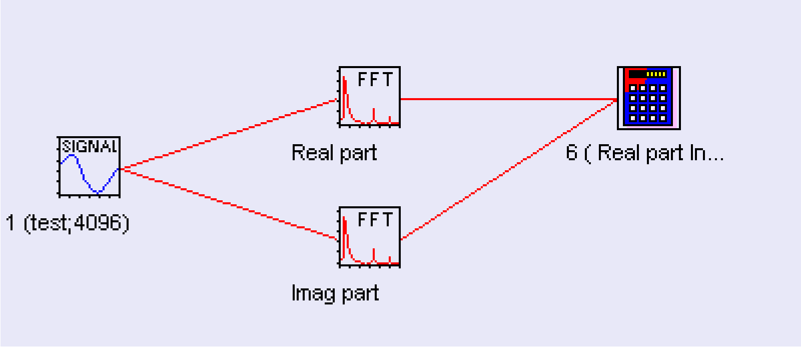
5.22. Inverse FFT

Inverse FFT is a function which converts complex spectrum in a time-domain signal, i.e. the function which reverts FFT result back in the origin signal. To apply this function, you need to provide complex spectrum with real and imaginary components. Applying Inverse-FFT directly to spectrum derivate like magnitude spectrum or power spectrum is not possible because important phase information is not available anymore.

Inverse FFT in *SIGVIEW* is included in Signal Calculator as a binary operator. It should be applied to a real and imaginary spectrum part as shown in the screenshot below:



The result would look like this in the Control Window:



Real and imaginary spectrum parts can be obtained in *SIGVIEW* in two ways*:*

1. Calculate FFT twice for a signal and choose "Real part" and "Imaginary part" in "Show

result as" field for these two spectrums respectively. For both spectrums, windowing and zero-expand options should be turned off if you wish to get exactly the same signal after inverse FFT. After calculating Inverse-FFT as shown above, resulting signal should be the same (up to the calculation precision) as the origin signal. Please note that DC component of the origin signal will also be lost, i.e. result of the Inverse-FFT will probably be translated for a constant offset on the Y-axis from the origin signal. To avoid it, you can use Remove DC offset function on the origin signal before calculating its spectrum.

2. *SIGVIEW* distribution includes ready-to-use custom tool named "RealAndImagFFT" which will automatically calculate real and imaginary FFT parts in two separate windows as described above. You can use it to prepare data for the Inverse-FFT function. since this tool uses no zero-padding when calculating FFTs, we recommend using power-of-2 signal length for best performance.

5.23. Custom filter curves

What is custom filter curve?

SIGVIEW allows you to define custom filter curves and to apply those during time-domain signal filter or for spectrum modification. Custom filter curve is actually a sequence of [frequency, modification factor] pairs which define how should a frequency component be changed.

For example, a band-pass filter with a pass-band between 1000 - 2000Hz could be defined this way:

1000	0.0
2000	1.0
20000	0.0

meaning that all frequency components up to 1000Hz should be removed (factor 0.0), all between 1000Hz and 2000Hz should remain intact (factor 1.0) and all above 2000Hz up to 20000Hz should be removed (factor 0.0).

Another example would be a standard a-weighting curve used in sound pressure level measurement. This curve is defined in dB, i.e. signal increase or decrease dB level for certain frequencies:

10	-70.43
20	-50.39
30	-40.6
40	-34.54
50	-30.27

....

meaning that frequency components around 10Hz should be reduced for -70.43dB, around 20Hz for -50.39 dB and so on...Note that we do not define upper frequency limits here as in first examples but center frequencies of each "bin".

What is included in SIGVIEW

SIGVIEW allows you to define all those filter types manually (by editing text files) or from existing spectrum curves and to apply them to signals or spectrums. All filter curves used by *SIGVIEW* are stored in a "filters" subdirectory of the *SIGVIEW* data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\). For more information about filter file structure and creation, see bellow.

Following filter curve types are supported:

Magnitude values filter: Filter values are used as absolute magnitude values in signal units. The filter amplitude values will be subtracted from the actual signal values when filter is applied (magnitude reduction). This means that value of 5 in a filter for certain frequency will actually reduce signal amplitude on that frequency for 5 signal units (for example Volts) when filter is applied. This filter will mostly be created from existing spectrum curves. For example, you can use *SIGVIEW* to calculate your microphone characteristic response curve, save it as a filter and subtract from a spectrum of the recorded audio.

Relative dB values filter: As in second example above, this filter contains dB values to modify certain frequency components. For example, value of -50 means that frequency component will be reduced for 50dB.

Relative values filter: As described in first example above, this filter contains relative coefficients which are used to multiply frequency components on certain frequencies (for example 0.5 means that frequency component is reduced to 50%). This type of filter will mostly be created manually by editing filter file with editor.

Creating custom filter curves manually

Custom filter curves are stored in plain-text files with extension *.flt stored in a "filters" subdirectory of the *SIGVIEW* installation directory. You can create these files in any text editor and save them into "filters" directory located in *SIGVIEW's* application data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\). During next startup, *SIGVIEW* will read your file and include it in all filter functions. Each filter has 3 header lines at the beginning and looking similar like these:

```
Type=DB_ADD
Frequency=CENTER
########
10	-70.43
20	-50.39
30	-40.6
.................
```

These lines are followed by rows containing frequency in Hz/value pairs using TAB (ASCII 0x9) as value separator. For an example, please take a look at delivered "a-weighting.flt" file.

Type parameter (Type=) in the FLT file defines which type of filter is in the file. Following values are allowed:

ABS_SUB: Corresponds to "Magnitude values filter" as described above. Following filter values will be interpreted as magnitude values and subtracted from corresponding signal magnitude values.

DB_ADD: Corresponds to "Relative dB values filter" as described above. Following filter values will be interpreted as dB values defining how to modify certain frequency component (for example -50 means that signal magnitude on that frequency will be reduced for 50dB).

ABS_MUL : Corresponds to "Relative values filter" as described above. Following filter values will be interpreted as coefficients to multiply certain frequency component with (0.5 means reduction of the magnitude for 50%).

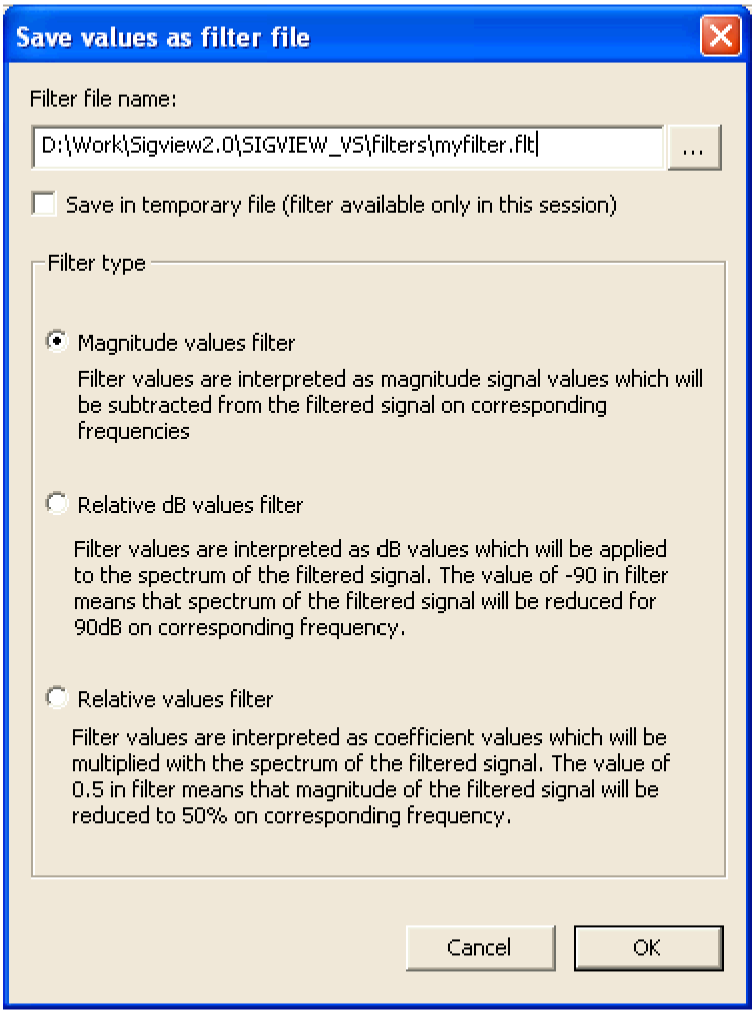
Frequency parameter defines how to interpret frequency values. Following values for this parameter are allowed:

CENTER: Frequency value is center frequency. Corresponding filter value will be applied to all frequencies ($(f_i + f_{i-1})/2$, $(f_i+f_{i+1})/2$]

MAX: Frequency value is a upper limit frequency. Corresponding filter value will be applied to all frequencies ($f_{i-1,}$ f_i]

Creating custom filter curves from SIGVIEW

You can save any signal or analysis result from SIGVIEW as filter curve. You will usually start with a magnitude spectrum or its averaged version. It can be directly saved as "Magnitude values filter". The ratio between two spectrums can, for example, be stored as "Relative values filter". In all cases, you will use *File/Save as custom filter...* main menu option on a window containing your filter sequence. The following dialog appears after opening this menu item:



Save values as filter file

Filter file name:

D:\Work\Sigview2.0\SIGVIEW_VS\filters\myfilter.flt

□ Save in temporary file (filter available only in this session)

Filter type

● Magnitude values filter

Filter values are interpreted as magnitude signal values which will be subtracted from the filtered signal on corresponding frequencies

○ Relative dB values filter

Filter values are interpreted as dB values which will be applied to the spectrum of the filtered signal. The value of -90 in filter means that spectrum of the filtered signal will be reduced for 90dB on corresponding frequency.

○ Relative values filter

Filter values are interpreted as coefficient values which will be multiplied with the spectrum of the filtered signal. The value of 0.5 in filter means that magnitude of the filtered signal will be reduced to 50% on corresponding frequency.

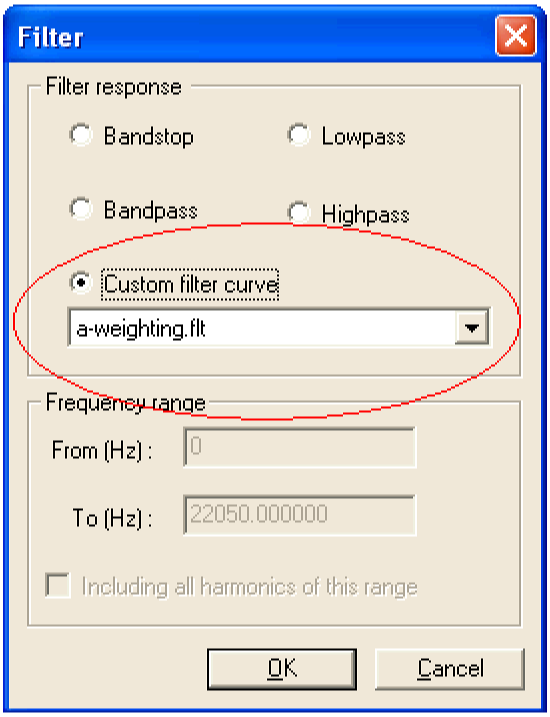
You have to define the **file name** of your filter. It has to have FLT extension and it should be stored in "filters" directory under *SIGVIEW's* application data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\). Otherwise, it will not be visible to *SIGVIEW.* You can also decide to store filter as a **temporary** file. In that case, you should only give it a name - there will be no file actually stored on disk. Of course, the filter will be lost when you close *SIGVIEW.* In **Filter type** box you have to tell *SIGVIEW* what type of filter it is, i.e. how to interpret filter values later when applying the filter. These types correspond exactly to the filter types explained above. After you press OK, filter will be saved and will be accessible from filter and spectrum properties functions.

All filters stored in this way will use **CENTER** setting to interpret frequency values (see above).

To delete filter, simply delete its *.FLT file in filters directory.

Using custom filter curves for time-domain signal filtering

You can use custom filters to directly filter time-domain data. This feature is implemented as a part of *Signal tools/Filter...* menu option. After applying your filter to a time-domain signal, the difference (or ratio) between magnitude spectrum of the original signal and magnitude spectrum of the filtered signal will correspond to the custom filter curve.



● Custom filter curve

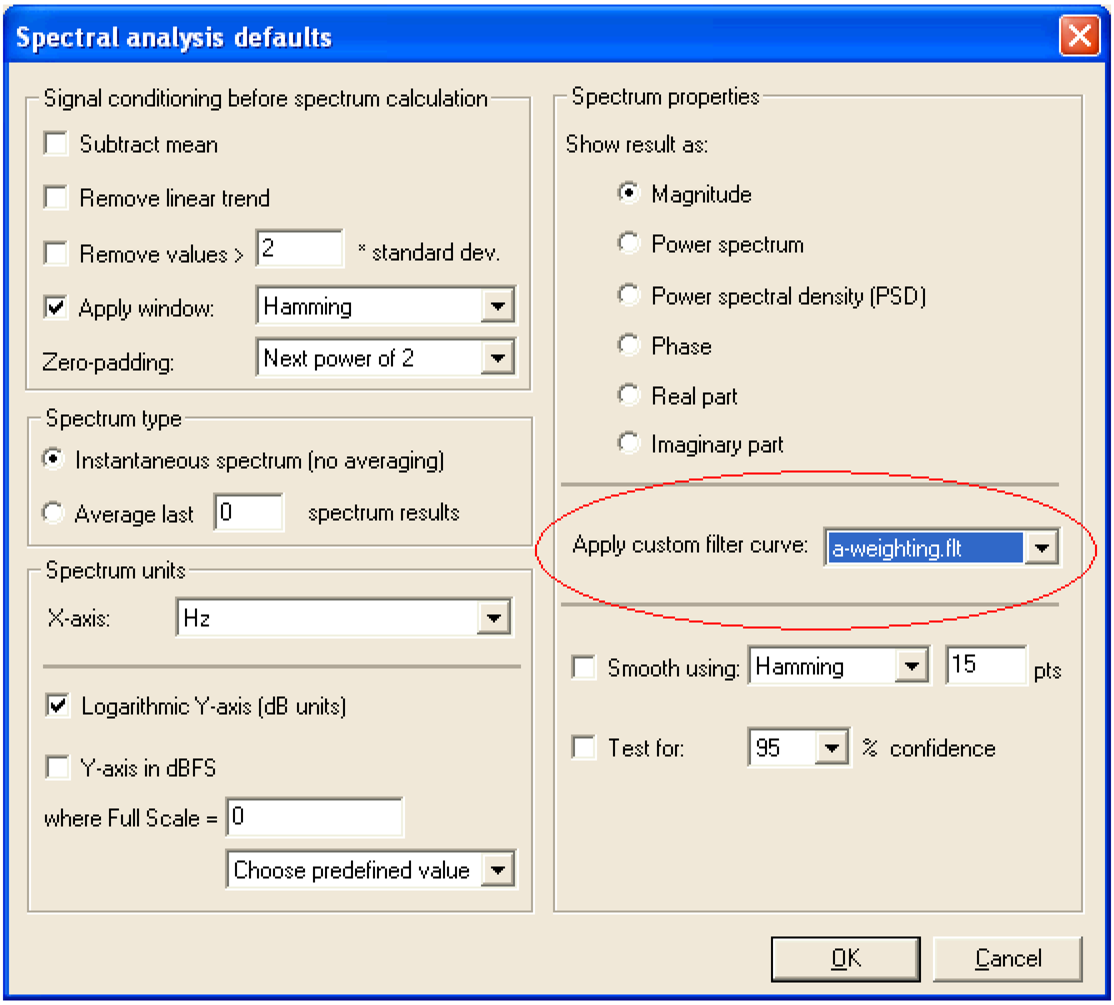
a-weighting.flt

From (Hz) : 0

To (Hz) : 22050.000000

Using custom filter curves for spectrum modification

You can use custom filters to modify calculated magnitude spectrum or power spectrum just as if the filter were applied to the origin time-domain data. This option is available from Properties dialog of any spectrum window or as general setting in Spectrum Analysis defaults dialog. After applying your filter to a spectrum, the difference (or ratio) between original spectrum and modified spectrum will correspond to the custom filter curve.



Spectral analysis defaults

Signal conditioning before spectrum calculation

□ Subtract mean

□ Remove linear trend

□ Remove values > 2 * standard dev.

☑ Apply window: Hamming

Zero-padding: Next power of 2

Spectrum type

● Instantaneous spectrum (no averaging)

○ Average last 0 spectrum results

Spectrum units

X-axis: Hz

☑ Logarithmic Y-axis (dB units)

□ Y-axis in dBFS

where Full Scale = 0

Choose predefined value

Spectrum properties

Show result as:

● Magnitude

○ Power spectrum

○ Power spectral density (PSD)

○ Phase

○ Real part

○ Imaginary part

Apply custom filter curve: a-weighting.flt

□ Smooth using: Hamming 15 pts

□ Test for: 95 % confidence

No matter what is the type of your custom filter (dB, absolute,...), *SIGVIEW* will be able to apply it to all spectrum types, i.e. dB filter will be also applicable on magnitude spectrum with linear scale values. *SIGVIEW* will perform all needed filter conversions internally.

For more information, see the Use custom filter curve for better measurement results section.

5.24. Octave plots

To determine the frequency components of a sound, *octave band analysis* in which frequencies are segmented into proportionate widths (octave bands) is used. In a standard full-octave plot, each band occupies a bandwidth that is twice as wide as the previous band and half as wide as the next band.

Octave analysis corresponds to the way how people perceive sounds because it lets you compare signal levels across broad frequency ranges. Breaking frequencies into octaves helps you measure the subjective qualities of sound.

To get a better frequency resolution, each octave band can be divided into 3 bands. This is called *1/3 octave plot* and is available as a separate function in *SIGVIEW*.

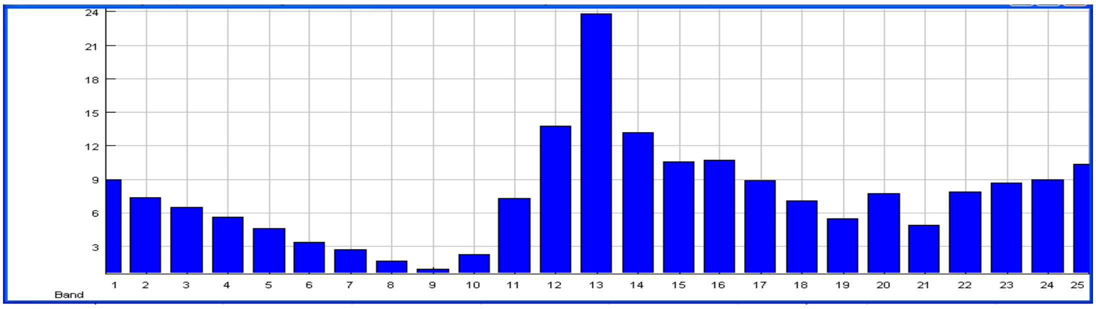
Both Octave and 1/3 Octave plot functions can be found in a "Signal tools" menu.

Implementation

Both Octave plot functions are implemented by using Custom Filter bank plot function and not by using FFT. This is required by the various technical standards defining octave plot calculations.

Filter bank definition files can be found in the "filters" subdirectory of the *SIGVIEW* application data directory. The files are named "octave_filter.flb" and "1_3_octave_filter.flb". As for all custom filter bank plots, you can freely change the frequency values in these files by using any text editor. For details, see the custom filter bank chapter.

By default, Octave plots are displayed by using histogram display type (min-reference). As for all other plots, you can change the display type by using a context menu or toolbar buttons (Display type option).



5.25. Custom filter bank plot

To create different equalizer or octave plots for audio analysis, it is often needed to define a bank of bandpass filters with different, possibly overlapping, frequency ranges. This task can be performed by using "*Custom filter bank*" option in *SIGVIEW*.

Creating filter bank files

For each custom filter bank, a definition file with the extension "FLB" is created and stored in the "filters" subdirectory of the *SIGVIEW* application data directory. Two example files will be stored there during *SIGVIEW* installation, "octave_filter.flb" and "1_3_octave_filter.flb". These files are used for Octave plots functions.

Each FLB file contains multiple rows, each representing one bandpass filter in the filter bank. For example:

```
14;18;A
18;22.4;B
22.4;28
...................
```

First row defines a filter with bandpass from 14-18 Hz. As a label for this band in a *SIGVIEW* graphics, a "A" will be shown.

Second row defines a filter with bandpass from 18-22.4 Hz. As a label for this band in a *SIGVIEW* graphics, a "B" will be shown.

Second row defines a filter with bandpass from 22.4-28 Hz. Since a label is not defined for this filter, a band index, i.e. "3" will be used as label.

Each filter bank has to contain at least 4 filters in order to be used.

Settings in filter bank files

Besides frequency limits for bandpass filters, you can define some additional settings for the calculation in the FLB file as well. All settings are added as text lines beginning with "#" character. The following settings are possible:

#X_LABEL defining label for X-axis

#Y_LABEL defining label for Y-axis

#FULL_SCALE defining the value to be considered as a "full scale" value in signal units when calculating energy of the filtered signal in dB.

This would be an example of FLB file with settings described above:

```
#X_LABEL=Bands
#Y_LABEL=dBFS
#FULL_SCALE=32768
14;18;A
18;22.4;B
22.4;28
```

....

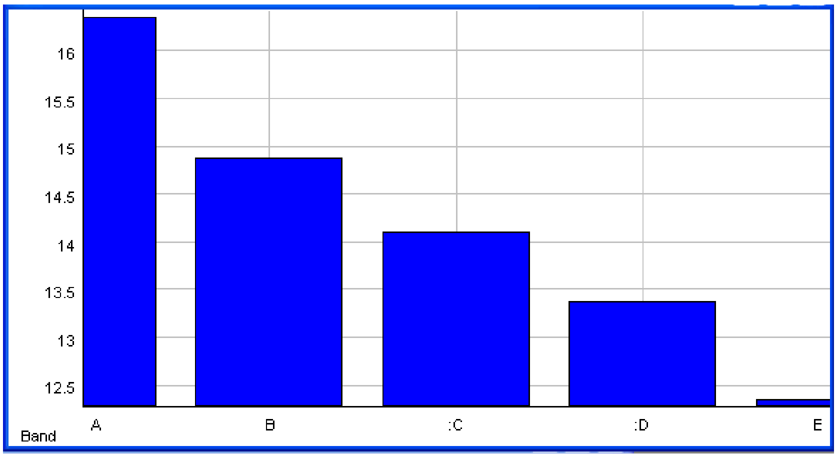
Please note that the settings names are case sensitive.

Applying filter bank function

To apply one of defined filter bank files to a signal, use "Signal Tools/Custom filter bank..." menu option and choose the FLB file you would like to apply.

After a specific filter bank file is applied to a signal, the following procedure will performed:

1. The signal will be filtered with each filter from the filter bank
2. The energy of the filtered signal (RMS) in dB (logarithmic scale) will be stored as a next value in the "Custom filter bank plot".
3. After all values for all filters are calculated, a plot will be shown. By default, these plots are displayed by using histogram display type (min-reference). As for all other plots, you can change the display type by using a context menu or toolbar buttons (Display type option).



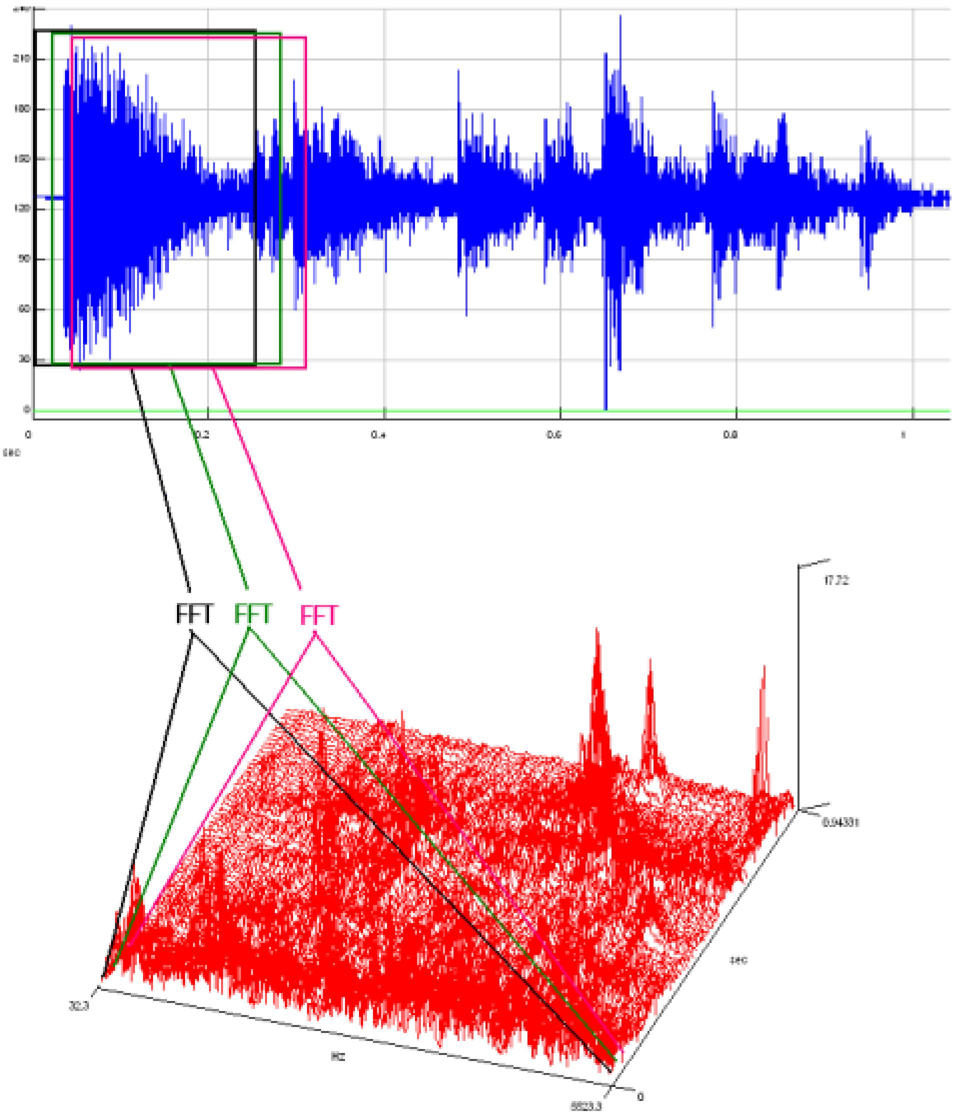
Part VI

3D graphics

6. 3D graphics

6.1. Time FFT

Time FFT function calculates change of signal’s frequency components through time. The signal is divided in smaller equally spaced segments (possibly overlapped), an FFT for each segment is calculated and all FFT sequences are used to build a 3D graphic.

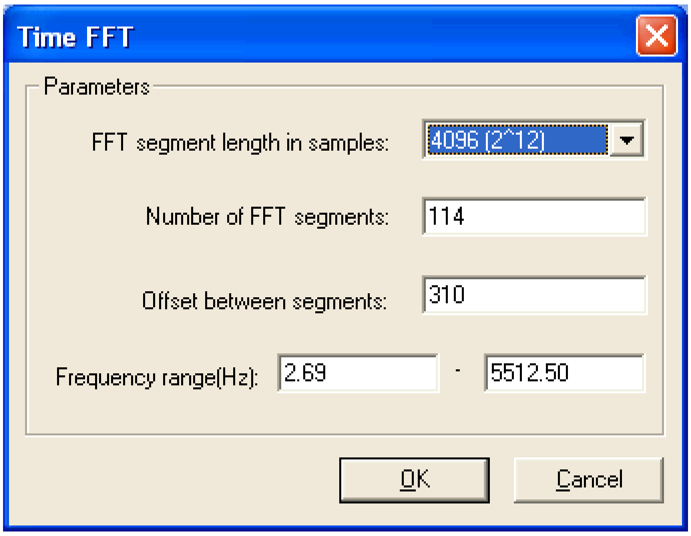


FFT calculation parameters depend on your current Spectral analysis defaults . You can change the parameters for the calculated Time-FFT anytime by choosing *Edit/Properties* option from the main menu or *Properties* from the context menu.

By choosing 3D-Tools/*Time-FFT* option from the main menu, you can set Time FFT parameters, calculate it and display 3D graphics.

Time FFT parameters:

6. 3D graphics



Time FFT

Parameters

FFT segment length in samples: 4096 (2^12)

Number of FFT segments: 114

Offset between segments: 310

Frequency range(Hz): 2.69 - 5512.50

FFT segment length in samples: Length of a signal segment for the FFT calculation

Number of FFT segments: Number of signal segments. It can be calculated from the segment length and offset parameters.

Offset between segments: Distance (in samples) between beginnings of consecutive signal segments. It can be calculated from FFT segment length and the number of FFT segments.

Frequency range: Frequency range you would like to see in your graphics.

Special case of Time-FFT analysis is Spectrogram – it is the same as Time-FFT except the 3D-graphics is shown from above and colored so that the amplitude differences are color-coded (same amplitude always has the same color). Choosing 3D *tools/Spectrogram* from the main menu brings up the same dialog as Time-FFT, except that resulting graphics is shown as Spectrogram.

Time FFT will by default not be linked to the origin signal because its calculation time can be much slower then real-time. If you explicitly link it to the signal, the step property of the signal will temporarily be changed to be the same as Time FFT step property. Now, when you move through signal, Time-FFT will change accordingly. When you unlink Time FFT from the signal, signals step property will restore its prior value. This is done to optimize Time FFT calculation, so every time you step through signal, FFT for only one new segment will be calculated.

6.2. Tracking changes as 3D graphics

It is often very important to see how the result of some analysis changes through time. If that result is a number, you can simply draw changes as function of one variable - time (use Instrument values log). But, if the result is a signal, for example some FFT sequence, you can track its changes through the time by creating a 3D graphics. Each time FFT values change, all its values will be added as one new column in a 3D graphics. You can perform this on FFT window but also for any other signal or analysis result window in *SIGVIEW*.

You can perform this function in *SIGVIEW* by choosing *Signal tools/Track changes as 3D graphics* option from the main menu. You will first have to determine the number of columns in 3D graphics, i.e. how many last signal changes will be stored.

Each time signal the origin signal changes, one new column containing this new signal will be added to 3D graphics and the ‘oldest’ column will be removed.

X-axis unit will be initially marked as ‘change’, i.e. the number of signal changes, and the Y-axis units will be the same as for signal’s X-axis. After few signals are added to the 3D graphics, X-axis unit and range will be detected automatically by backtracking through the analysis chain to the root signal window. If X values are equidistant (i.e. difference between two consecutive values is always the same), those will be display on X-axis of the 3D-graphics. If not, X-axis will switch again to a "change" unit where each new signal will simply be labeled with its index, i.e. 0,1,2,3,4,...

If the length of origin signal changes (for example by zooming it in/out), its further changes will not be added to the 3D graphics until you return the signal to the previous (original) length.

For more information see: Track analysis results through time as 3D graphics and Basic 3D operations

6.3. 3D graphics: basic operations and zooming

Rotation
If you want to get a good look on 3D graphic from all sides, use arrow keys on your keyboard to rotate 3D graphic in all directions. This is possible only if you are not in ruler mode (see below).

Switching Waterfall / Mesh view
You can switch from waterfall (lines only) to mesh (solid rectangles) view by choosing 3D Tools/Mesh option from menu or by pressing button in toolbar. You can also use F2 key as a shortcut to switch between views.

Ruler
By pressing TAB key or button in toolbar you can switch to ruler mode. Ruler will appear on the graphic and you will be able to move it with keyboard arrow keys. The ruler's X,Y position and f(X,Y) for the current ruler position will appear in status bar at the bottom of the screen.

Zooming in/out

Zooming-in part of the graphic:

In spectrogram view, simply select the part of the graphics with mouse (rounding box), and then choose *Edit/Zoom in* menu or toolbar option. If the ruler is turned on, zoom-in option will always zoom-in to an area around the current ruler position.

Edit/Zoom out option and button from toolbar will restore the zoom positions before the last zoom in operation.

Zoom-in graphical content of the window:

By pressing and buttons in toolbar you zoom in/out only graphical content of a window. This zooming will affect the displaying only – the data displayed will remain the same.

Palettes: In a 3D graphics mesh mode, the color of the rectangles is defined by a current palette. You can define it for each window separately from its context menu or from the main menu. The available palettes are: Hot (red/black), Blue (blue/black), Gray (white/black), Copper (yellow/black), HSV (all colors), White (inverted gray palette)

Inverting colors: If the 3D graphics is in a Mesh mode, you can invert it colors by choosing *Invert colors* menu option from the context or main menu. By mot palettes, it will provide the graphics with the light colors for the values around 0, making the graphics much more printer-friendly.

Z-axis adjusting
When 3D graphic is recalculated and redrawn because its parent signal has changed, min & max values are not automatically recalculated to avoid speed loses. This can cause 3D graphic to go out from visible part of coordinate system, or can corrupt the color layout. If that happens, you can select *3D Tools/Adjust Z-axis* option from menu and min & max values for Z axis will recalculate. This will solve all described problems.

6. 3D graphics

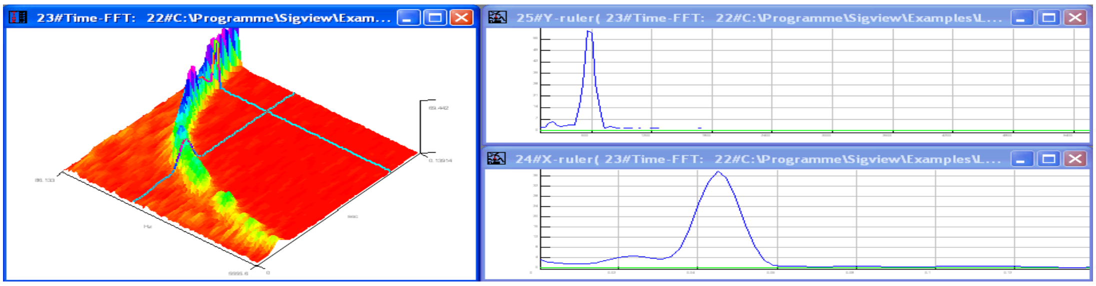
Exporting 3D graphics values in file: By using options *File/ASCII files/Export 3D-graphics*, you can export all values from the visible part of the 3D graphics in a one file. There are 3 options: export in a file with X,Y,Z triples in 3 columns (tab delimited), as a matrix where first row contains X values, first column Y values and the rest of the file are Z values in a tab delimited table and finally inverted variant of the second matrix format (with switched rows and columns). All of these file types can be loaded by, for example, MS Excel for further analysis.

Copying 3D graphics values to the clipboard: By using menu option *Edit/Copy data to clipboard,* all values from the visible part of 3D graphics will be copied to clipboard*.* The data format will be the same as when exporting data in ASCII: matrix where first row contains X values, first column Y values and the rest of the file are Z values in a tab delimited table.

6.4. Extracting signals from 3D graphics

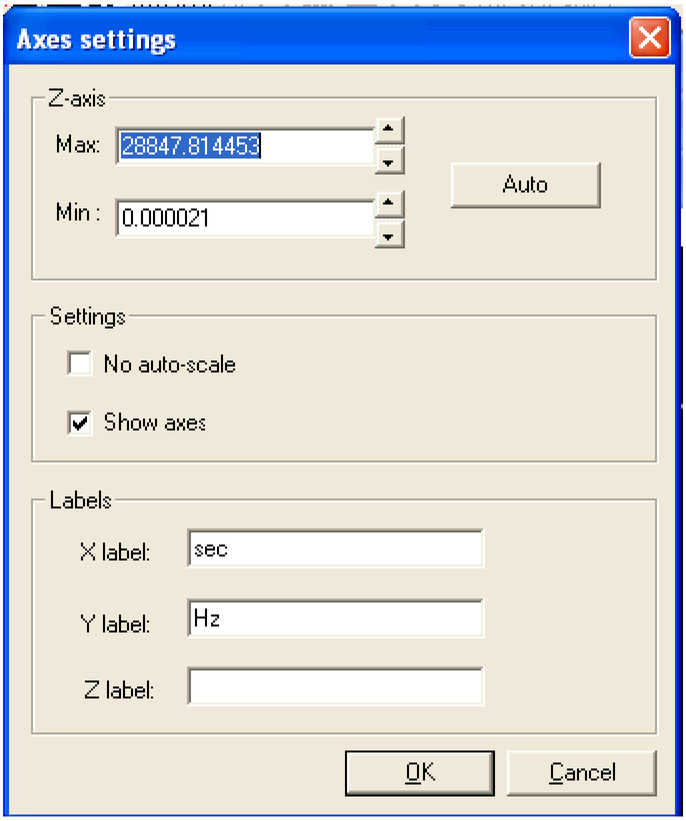
It is often very useful to see extracted slices from 3D graphics as signals. For example, you can extract all values at the same frequency from Time FFT, to see how that frequency component changes through time.

You can perform this in *SIGVIEW* by selecting *3D Tools/Extract X(Y) ruler* options from menu (3D graphic must be in a ruler mode). All values from the current ruler position will be extracted to a separate signal linked to this 3D graphic. By moving 3D ruler along the 3D graphics, this signal will redraw to show the current ruler values. Even if you turn the ruler mode off, the extracted signal will keep showing the corresponding values from the 3D graphics (from the last ruler position).


23#Time-FFT: 22#C:\Programme\Sigview\Exam...
25#Y-ruler(23#Time-FFT: 22#C:\Programme\Sigview\Examples\L...
24#X-ruler(23#Time-FFT: 22#C:\Programme\Sigview\Examples\L...

6.5. Axes settings (3D graphics)

Axes settings for 3D graphics is done through the Axes settings dialog, which appears after selecting *Edit/Axes settings* option.



Axes settings

Z-axis

Max: 28847.814453

Min : 0.000021

Settings

□ No auto-scale

☑ Show axes

Labels

X label: sec

Y label: Hz

Z label:

There are several fields in this dialog:

Max & Min fields are used for setting Z-axis range. You can enter your own values in the edit box, or you can change them by pressing up/down arrows on the right side of each edit box.

Auto button sets values in min/max fields to the actual min/max values from the graphics.

No auto scale check box: select it if you want to prevent automatic changes of the Z-axis settings when graphics values change.

Show axes check box: select it if you want to see the lines and labels of the coordinate system

Labels: text which appears at X, Y and Z axes if *Show axes* option is checked

6.6. Types of 3D graphics

3D graphics are used to display any function defined by f(x,y) where x and y can be different domains, like frequency, time,…

There are two types of 3D graphics in *SIGVIEW:*

- Time FFT: f(X,Y) is a function of time and frequency. The function value in some (X,Y) point represents the intensity of the frequency component Y (Hz) in the moment X (seconds).
- Tracking changes as 3D graphics : f(X,Y) depends on nature of signal that is being logged. Y axis will be the same as X axis from that signal and X axis will represent the number of logged signal changes (this unit is called ‘change’).

6.7. Spectrogram view

If you try to rotate 3D graphic to set viewpoint somewhere on Z axis (look 3D from above), you will find out that it is not possible. That view is implemented as a separate option called *Spectrogram view*. It has been done mainly because of scrolling optimization when tracking FFT changes as 3D graphics.

You can switch to Spectrogram view by choosing *3D Tools/Spectrogram view* from menu, or pressing button in toolbar.

All operations including ruler handling, zooming, colors changing,... are the same as with standard 3D- graphics. In addition, you can use a mouse rectangle selection to zoom-in fast to part of the spectrogram.

Spectrogram view is very useful and fast when used with 3D graphics created with “Track changes as 3D graphics” option, because a big part of the graphic can simply be scrolled, and only a new data is drawn each time data in origin window (for example, FFT) changes.

Spectrogram view makes it easy to draw *spectrograms* in *SIGVIEW.*

6.8. Palettes

3D graphics can use different colors to show different f(X,Y) values of represented function. One 3D graphics can use only one color set, called *palette*. Palette in *SIGVIEW* has 128 colors.

You can set palette used by one 3D-graphics from its context menu or from the main menu, *3D tools/Palette* menu option.

Currently available palettes are:

- gray
- hsv
- blue
- pink
- hot
- white
- jet
- rainbow
- winter
- autumn
- spring
- summer
- cool

Some palettes are better used for linear scaling and some are better suited for log scales. The best way to see palette look & feel is to try it on an example 3D graphics.

Their definitions are in simple text files with *.pal extension (hot.pal, blue.pal, etc.). Each file has 128 rows with 3 numeric values defining each of 128 colors as intensities of red, green and blue (RGB). You can freely change values in these files to define your own palettes, provided that they keep same structure and number of rows.

Part VII

Instruments

7. Instruments

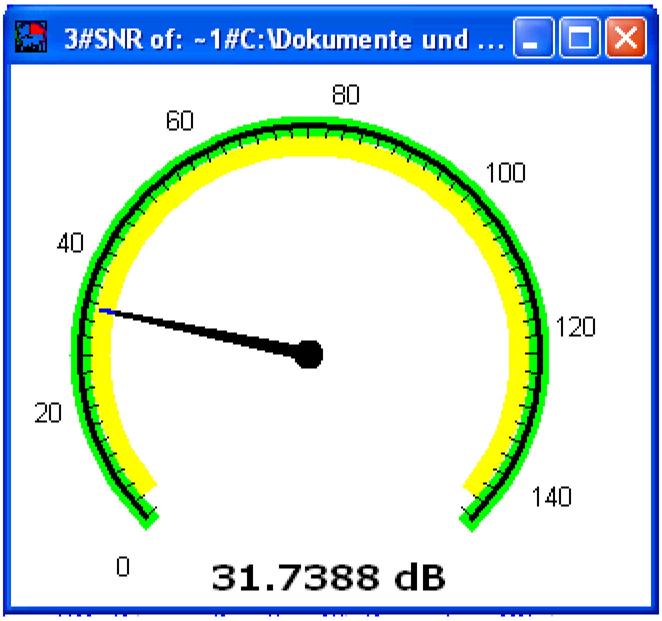
7.1. Instruments

7. Instruments

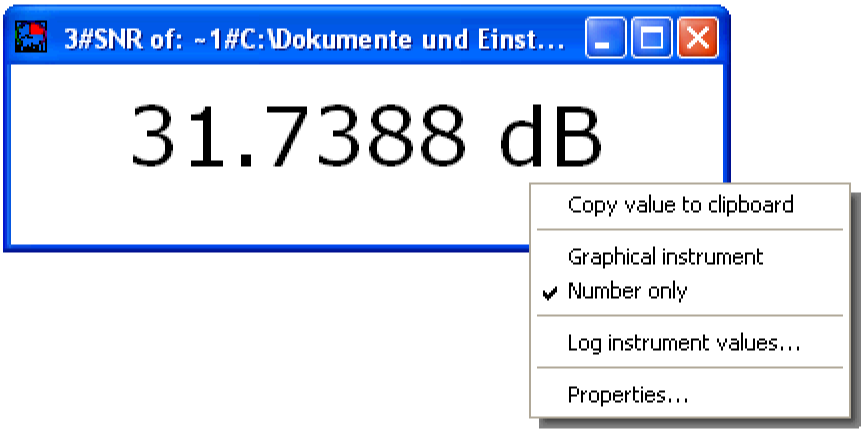
7.1. Instruments, overview

To calculate and monitor the changes of some numeric value, you can use *SIGVIEW*'s instruments. They calculate different values from the signals or 3D graphics and display them as a number or in graphical form – as round-scale instrument.

Right-click menu on the instrument window includes many useful options: copying instrument value to the clipboard, logging instrument values, editing instrument properties etc.



31.7388 dB



31.7388 dB

You can create instruments by using one of options in the *Instruments and markers* main menu section. *SIGVIEW* supports the following instrument types:

Marker at...: You can define X value from the signal and instrument will monitor Y value (for example you can monitor FFT value on certain frequency)

Marker with harmonics at...: Available for FFT windows only. Shows value at desired frequency along with markers on all harmonics of that frequency.

Mean: calculates mean value of the signal

Sum: calculates sum of all signal values

Const: Creates an instrument with a constant numerical value. It can be used for expressions in the Signal calculator

RMS (Root Mean Square): If the signal include negative values, the sum instrument will probably not be useful. In that case you can use this instrument – the *square root* of the *mean* of the *squares* of the signal values. RMS=sqrt((x1^2+x2^2+…..)/N).

RMS –normalized: The same as standard RMS function, but will normalize the signal first by subtracting its mean value from all signal values.

Standard deviation: Calculates the standard deviation of the visible part of the signal.

Variance: Measure of the amount of variation within the signal values, equal to the square of the standard deviation.

Skewness: Measure of symmetry, or more precisely, the lack of symmetry of distribution. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point. The skewness for a normal distribution is zero, and any symmetric data should have a skewness near zero. Negative values for the skewness indicate data that are skewed left and positive values for the skewness indicate data that are skewed right.

Kurtosis: Measure of whether the data are peaked or flat relative to a normal distribution. That is, data sets with high kurtosis tend to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. A uniform distribution would be the extreme case. *SIGVIEW* uses "excess kurtosis" meaning that standard normal distribution has a kurtosis of zero.

Crest factor: Calculates the crest factor of the visible part of the signal.

Integral: Calculates numerical integral over the visible signal part

Weighted mean (Mean frequency): Calculates sum of products Xi * Yi divided with sum of Yi (Xi are values on x-axis and Yi corresponding values on y-axis from origin plot). If applied to a FFT plot, it is equivalent to a mean frequency.

Weighted median (Median frequency): Xi are values on x-axis and Yi corresponding values on y-axis of the origin plot. This function calculates Xi value so that sum of Xi * Yi on both sides of the Xi is the same (or nearly the same). If applied to a FFT plot, it is equivalent to a median frequency.

Signal-To-Noise Ration (SNR): Calculates ratio between signal power and noise power in the overall signal. The higher the ratio, the less obtrusive the background noise is. Since *SIGVIEW* does not have any information about the real nature of your signal, it will not be able to deliver accurate values in all cases. Therefore, you should use it with pure tone signals (i.e. sine) signals to achieve best accuracy. SNR units are decibel (dB).

Total Harmonic Distorsion (THD) and Total Harmonic Distorsion + Noise (THD+N): THD is a measurement of the harmonic distortion present in the signal and is defined as the ratio of the sum of the powers of all harmonic components to the power of the fundamental frequency. Lesser THD allows the components in a loudspeaker, amplifier or microphone or other equipment to produce a more accurate reproduction by reducing harmonics added by electronics and audio media. You should measure it in signals containing one pure tone (sine) component. THD+N means total harmonic distortion plus noise. This measurement is much more common and more comparable between devices. Both instrument values are calculate in decibel (dB). To convert dB values to percent values, you can use one of online calculators, for example here.

Maximum: calculates maximum value from the signal

Minimum: calculates minimum value from the signal

Maximum position: calculates the position of maximum from the signal (for example frequency of the max. peak in FFT)

Maximum position (interpolated): estimates the position of the maximum value more precisely by using interpolation. It is especially useful for determining fundamental frequency in the spectrum with precision better then the frequency resolution of the spectrum.

Minimum position: calculates the position of minimum value from the signal

Sum 3D over frequency range: it is the only instrument applicable to 3D graphics (usually Time FFT). It calculates the sum of all values from the 3D graphics for one segment of the Y axis (usually frequency).

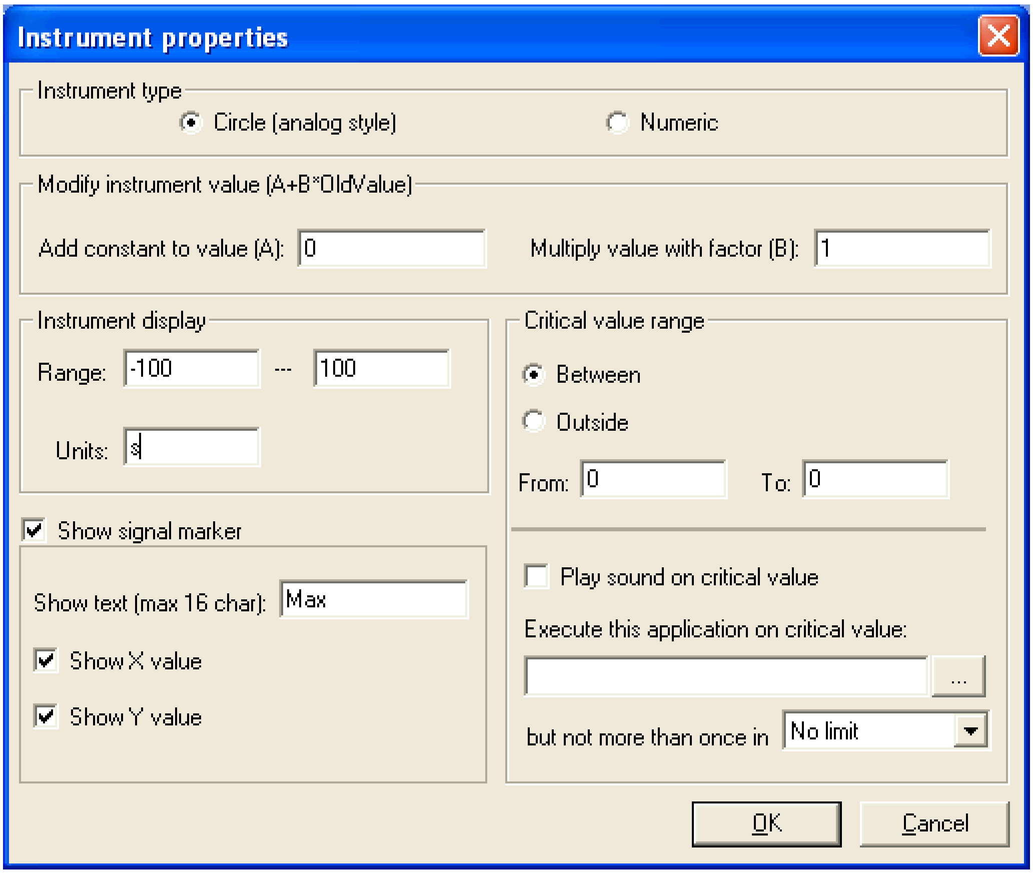
Peak count: Shows number of peaks detected by the Peak detection function

The context menu for each signal window includes “Show 5 highest peaks” option. With this option turned on, *SIGVIEW* will show 5 markers on 5 highest signal values in the visible part of the signal.

To copy the values of all visible markers to the clipboard in a TAB-separated format, you can use the context menu option "Copy markers to clipboard" on the signal containing the markers.

7.2. Instrument properties

By choosing *Edit/Properties* for instrument window, you will be able to set some of the following instrument properties:



Instrument properties

Instrument type
● Circle (analog style) ○ Numeric

Modify instrument value (A+B*OldValue)
Add constant to value (A): 0
Multiply value with factor (B): 1

Instrument display
Range: -100 --- 100
Units: s

Critical value range
● Between
○ Outside
From: 0 To: 0

☑ Show signal marker
Show text (max 16 char): Max
☑ Show X value
☑ Show Y value

□ Play sound on critical value
Execute this application on critical value:
but not more than once in: No limit

Type: There are 2 types of instruments:

- Circle with classic round scale
- Non-graphic, with numeric value only

Modify instrument value: You can define coefficients to modify the instrument value after calculation. It is useful if you need a calculation slightly different than the one included in *SIGVIEW* , for example if the unit you would like to use needs some scaling. You can define both add (A) and multiply factors (B). The resulting instrument value will be calculated from the originally calculated value by using the following expression: NewValue = A + B*OriginalValue.

Range: Value range for instrument scale

Units: Units for instrument value (Volt, meter, etc.)

Show signal marker: This option is available only for *At..., Maximum, Minimum, Maximum position and Minimum position* instruments. If you turn it on, a marker will appear in the origin signal, showing exact position for the value shown in instrument. You can define the content of the marker by changing its text, or turning X/Y value display in the marker.

Critical value range: You can choose value range which represents the “red zone”. It will be marked red on the instrument. By choosing *Outside* option in the radio box you will invert the selection, i.e. all values outside that range will be marked red.

Play sound on critical value: *SIGVIEW* will play a single beep when the value is in a critical zone.

Execute this application on critical value: You can enter complete path of an external application to be executed if the value is in a critical range, i.e. if value of the instrument enters “red-zone”. You can also define how often this application will be called. *SIGVIEW* will execute defined application and pass current instrument value as a parameter to it. For example, if your application is “C:\MyLogger.exe”, and value of 45.4432 causes alarm to goes off, *SIGVIEW* will call “C:\MyLogger.exe 45.4432”. This function can also be used to simply log analysis values outside *SIGVIEW* in some regular intervals.

As an example for the external application, *SIGVIEW* distribution includes “Logger.exe” application. You can simply copy this file from *SIGVIEW* installation folder to some other folder and enter its path to the instrument dialog. Logger.exe will log the values passed from *SIGVIEW* in a TXT file with the name SIGVIEWLog_<current date>.txt in its working directory.

Another example for such external application is ULDigital.exe, also included with *SIGVIEW* installation. This application can control digital outputs on devices compatible with Universal Library from Measurement Computing(R), i.e. can turn external devices on or off. There are following command line options: ULDigital.exe <duration in ms> <digital output>.
where <digital output> should corresponds with UniversalLibrary's FIRSTPORTA, SECONDPORTA,... values. For example, on PMD devices these values are 10,14,18,... for channels A0, A1, A2,... This application requires that Universal Library driver is installed.

7.3. Logging instrument values

After you create an instrument , you can log changes of its values through time with the *Log instrument values* option in *Instruments and markers* menu*.*

After you choose *Log instrument values* option*...* option from the *Signal tools* menu, you will have to determine the length of the log signal (in samples). A new, empty, signal window will appear.

As soon as the instrument value changes, its new value will be added to the log window - it will work just as old-fashioned paper tape recorder.

Scrolling and resetting

Log window will monitor all changes in the instrument and add all consecutive values from instrument into one signal. When this signal reaches maximum length you determined, the display will start to scroll, removing oldest values and adding new ones. By choosing *Reset data* from the Log window's context menu (or pressing F3), you can clear all value and start logging again. By using Start/Stop options from the context menu, you can stop logging and start it again.

X-axis units

Value log signal will try to determine values for its X-axis automatically by backtracking through the analysis chain to the root signal window. If those X values are equidistant (i.e. difference between two consecutive values is always the same), those will be display on X-axis of the instrument log. If not, X-axis will switch to unit called "change" where each new value will simply be labeled with its index, i.e. 0,1,2,3,4,...

Long-time logging of instrument values in a file

Since instrument log keeps its data in your computer's RAM, total storage length is very limited. For long-time recording of signal values, you can use Logging data to file option connected to a log instrument window. For example, this would be a Control Window display of a system to store max. frequency from an FFT into a file.

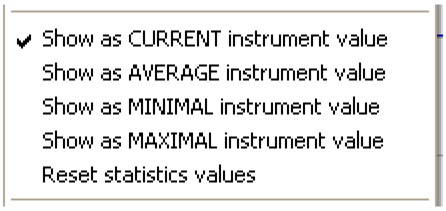
Instrument log in this system stores only a last 100 values and can be used for monitoring. File log window, on the other hand, stores all values which are added to a instrument log into a file.

For more information, see Track changes of signal parameters through time How-To section.

7.4. Instruments statistics

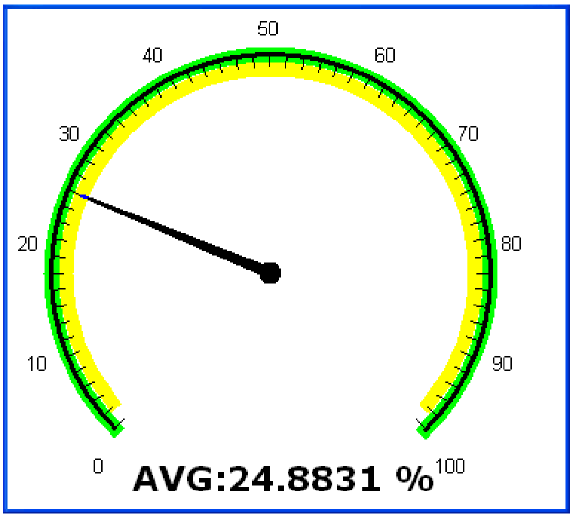
Each instrument includes statistics functions which automatically track its maximal, minimal and average values.

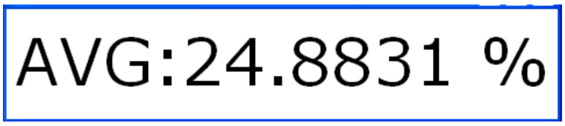
To see one of these automatically calculated statistical values, you can use instrument's context menu:


✓ Show as CURRENT instrument value
Show as AVERAGE instrument value
Show as MINIMAL instrument value
Show as MAXIMAL instrument value
Reset statistics values

As a default setting, each instrument will show its current value, i.e. "Show as CURRENT instrument value" option will be selected.

After choosing some of statistical values (AVERAGE, MINIMAL, MAXIMAL), the instrument will switch to displaying the chosen statistic value. That will be indicated by "AVG", "MIN" or "MAX" prefix to the instrument value.





By choosing "Reset statistical values" options from the context menu, you will reset all statistical values to the current value of the instrument.

After switching instrument display to some of statistical values, it will affect all windows attached to the instrument window as well, for example instrument value log.

Part VIII

Data acquisition

8. Data acquisition

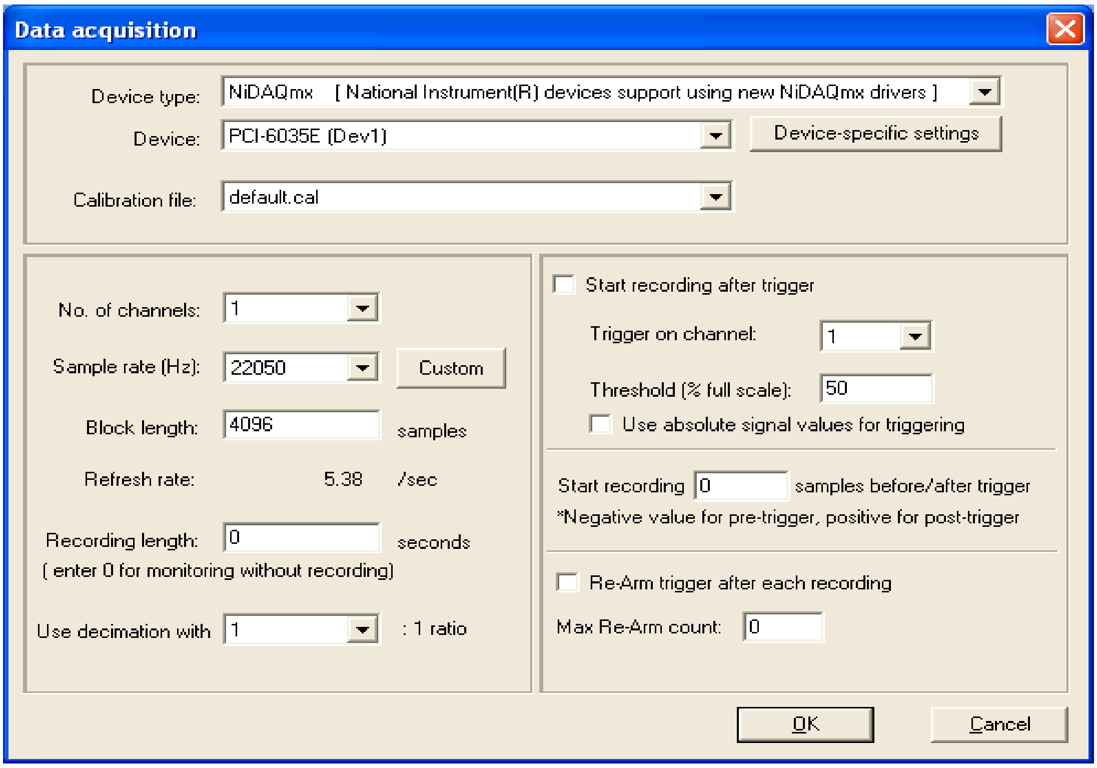
8.1. Data acquisition overview

SIGVIEW supports real-time data acquisition from the following device types:

- **Windows compatible sound cards** (up to 16-bit resolution, up to 2 channels, old-style drivers)
- **DirectSound compatible sound cards** (maximal supported sound card resolution, typically 24-bit for professional sound cards, typically all available channels)
- **ASIO sound card drivers** (by Steinberg) for low-latency and high fidelity sound card recording
- **National Instruments(R)** DAQ cards with support for **NiDAQmx drivers** (see www.ni.com)
- **National Instruments(R)** DAQ cards with support for **NiDAQ drivers** (see www.ni.com)
- **Measurement Computing(R)** DAQ cards supported by **Universal Library** (see www.measurementcomputing.com)
- Data acquisition **simulation** by reading data from specially prepared **signal file**

SIGVIEW offers a generic user interface and settings for data acquisition from all supported devices. Device-specific settings (if any) are available through additional dialogs. To start data acquisition, choose *Data acquisition/Open data acquisition* menu option. A dialog appears which allows you to choose data acquisition device and to change the data acquisition settings:

8. Data acquisition



Data acquisition

Device type: NiDAQmx [National Instrument(R) devices support using new NiDAQmx drivers]

Device: PCI-6035E (Dev1) | Device-specific settings

Calibration file: default.cal

No. of channels: 1

Sample rate (Hz): 22050 | Custom

Block length: 4096 samples

Refresh rate: 5.38 /sec

Recording length: 0 seconds

(enter 0 for monitoring without recording)

Use decimation with 1 : 1 ratio

□ Start recording after trigger

Trigger on channel: 1

Threshold (% full scale): 50

□ Use absolute signal values for triggering

Start recording 0 samples before/after trigger

*Negative value for pre-trigger, positive for post-trigger

□ Re-Arm trigger after each recording

Max Re-Arm count: 0

OK | Cancel

Device type: Device type to be used. It corresponds to the above list of device classes supported by *SIGVIEW*. If you have no devices of specific type installed in your system, you will not see the corresponding entry in this combo-box.

Device: Actual device to perform data acquisition. Only available devices from the chosen device type will be shown in the combo-box.

Device-specific settings button: Some devices allow special settings to be made, for example hardware triggering, input range, gain, etc. By pressing this button, a new dialog will open allowing you to change device specific settings. For more information about these settings, please consult your hardware manual for the data acquisition device. If there are no specific settings for currently chosen devices, this button will be disabled.

Calibration file: see calibration topic for details. Alternatively, some professional DAQ devices offer “Automatic calibration” option. This means that *SIGVIEW* will try to obtain data in physical units (usually Volts) from the DAQ device.

Channels: The number of input channels to use. *SIGVIEW* will use channels from 1...NumberOfChannels.

Sample rate (samples/sec) : this is the sampling rate for **one channel.** The maximum value is determined by your DAQ device capabilities. The combo box will show only standard types supported by most cards. If you want to try some other custom sampling rate, press “Custom” button and enter the sample rate manually. Please note that some sound cards will simply report that they support any sample rate even if they don’t support it in hardware. Those will simply try to simulate it on a driver level. If you want to avoid it, please consult the card's manual to determine which sample rates are really supported. Most devices do not allow all possible sample rate settings. *SIGVIEW* will try to find the next possible sample rate settings applicable to your device and apply it automatically. You will see the actual sampling rate in the status bar of the data acquisition window after you close this dialog.

Block length (samples): determines how much samples will be shown in a window during acquisition.

Recording length (sec): If you want only to monitor the signal without any saving choose length of zero. In that case, you will always have only the last "Block length" samples available in your data acquisition window. Otherwise, enter the duration in seconds you want to save. During acquisition, you will always see the last "Block length" samples in your acquisition window but you will see all samples after zooming-out. Please note that *SIGVIEW* stores all data in physical memory (RAM). Therefore, saving more then several minutes of data by using this method is not recommended. You can use Logging data to file feature for that purpose.

Decimation ratio: To reduce the amount data which has to be processed during analysis, you can use decimation feature, i.e. convert multiple samples from the DAQ device to one sample in *SIGVIEW*. You can define a decimation ratio by determining how many samples from the DAQ device will be converted to one sample in *SIGVIEW*. If you choose decimation ratio greater then 1, *SIGVIEW* will calculate the actual sample rate and show it as a read-only information under this field. For example, if the sample rate on Sound card is 44100Hz, and a decimation ratio is 100, then the actual sample rate in *SIGVIEW* will be 441Hz.

Start recording after trigger: If you check this option, recording will not start immediately after pressing START button – *SIGVIEW* will wait for a specific trigger (i.e. signal level) value in the signal and then start recording automatically. Trigger behavior and the definition of trigger value is determined by using the following options:

> **Trigger on channel:** Determines which of the data acquisition channels will *SIGVIEW* monitor for a trigger value. It is possible to receive data on one channel and use the other one only for triggering.
>
> **Threshold (% full scale):** This option determines which signal level is exactly regarded as a trigger value in the signal. Each sample with the amplitude higher than the threshold, by default 50%, from the maximum possible signal amplitude will be regarded as a trigger. For example, if your trigger is very strong signal impulse, you can set threshold to 90%. If you would like to use absolute signal values for triggering, select "**Use absolute signal values for triggering**" option. In that case, negative signal values will also be taken into account as trigger.
>
> **Start recording X samples before/after the trigger:** This option determines on which position, relative to the trigger appearance, will *SIGVIEW* start recording. The position is defined in number of samples. Negative value will cause *SIGVIEW* to start recording before the trigger and positive value after the trigger. Maximal number of samples you can set for pre-trigger option is equal to "Block length" setting (see above).

Re-Arm trigger trigger after each recording: After data acquisition is started after trigger and required signal length has been recorded, data acquisition would normally stop. By turning this option on, you can instruct *SIGVIEW* to automatically start new data acquisition window after the triggered data acquisition has been finished. New data acquisition window will use exactly the same settings as the previous one, i.e. will wait on trigger, record, and then start new data acquisition window. This way, you can let *SIGVIEW* automatically record multiple triggered signal events. Each recording will be stored in a separate window. To limit numbers of new windows which can be created during this operation, you can set **"Max Re-Arm Count"** option. Please note that trigger Re-Arm options make sense only if you define some specific recording length. If you use the value of zero (monitoring only), trigger will never be re-armed because recording will never stop.

For further information about the triggering, please see the How-To:
Using data acquisition triggering options

After you set all these data acquisition parameters and press OK button, one or more s*ignal windows* will appear (depending on number of channels). Now you can press REC button on toolbar or choose *Data acquisition/Start* menu option to start data acquisition. If you choose acquisition length of 0 (monitoring only), you can stop it by pressing stop button on the toolbar or choosing *Data acquisition/Stop* menu option. Otherwise, data acquisition will stop automatically after the desired acquisition length is reached. While data acquisition is running, you can work with the *signal window* as you would with any window containing static signal: for example, you can perform FFT of running signal, track its changes as 3D graphics, and you will have real 3D spectrum analyzer.

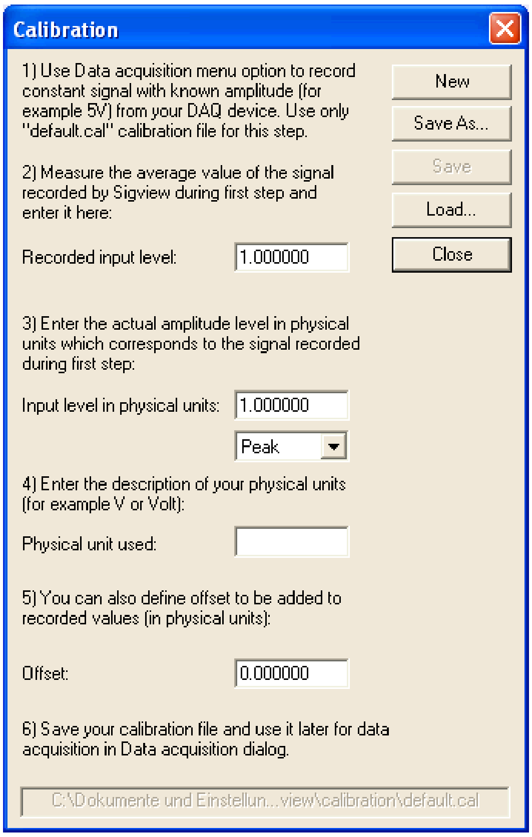
After data acquisition has finished, the signals in data acquisition signal windows are zoomed-in to show only the last recorded block. You will have to choose *Zoom-out* option to see the whole recorded signal.

If you choose ‘File’ driver type you will be able to simulate data acquisition by reading data from file in 16-bit signed binary format (You can store any data in that format by using Binary file export feature). After clicking on “Device specific options” button you will be able to choose the file name. SIGVIEW will try too keep the desired sample rate if there is enough data in the file.

8.2. Calibration

Signal data received from the A/D device is for *SIGVIEW* only a series of samples with different numeric values. For example, each sample from the sound card has usually a value between -32767 and 32767 (16-bit resolution). Of course, these values have no physical meaning. To transfer these values into meaningful physical values like Volt, meter, Celsius, etc., you have to tell *SIGVIEW* how to convert generic numeric values into real word units. This complete transformation is described in a 'calibration file'. You can create as many calibration files you want, and use them for different data acquisitions.

By choosing *Data acquisition/Sound card calibration* option from the main menu, you will open a calibration dialog where you can create a calibration file through 6 simple steps:



1) Use Data acquisition menu option to record constant signal with known amplitude (for example 5V) from your DAQ device. Use only "default.cal" calibration file for this step.

2) Measure the average value of the signal recorded by Sigview during first step and enter it here:

Recorded input level: 1.000000

3) Enter the actual amplitude level in physical units which corresponds to the signal recorded during first step:

Input level in physical units: 1.000000

Peak

4) Enter the description of your physical units (for example V or Volt):

Physical unit used:

5) You can also define offset to be added to recorded values (in physical units):

Offset: 0.000000

6) Save your calibration file and use it later for data acquisition in Data acquisition dialog.

C:\Dokumente und Einstellun...view\calibration\default.cal

1. Use Data acquisition menu option to record constant signal with known amplitude (for example 5V) from your DAQ device. Use only "default.cal" calibration file for this step.
2. Measure the average value of the signal recorded by SIGVIEW during first step and enter it in 'Recorded input level' field.
3. Enter the actual amplitude level in physical units which corresponds to the signal recorded during first step in 'Input level in physical units' field. You can also define if this value should be considered as peak or RMS value.
4. Enter the description of your physical units (for example V or Volt) in field 'Physical unit used'.
5. You can also define offset to be added to recorded values (in physical units).
6. Save your calibration file and use it later for data acquisition in Data acquisition dialog.

8. Data acquisition

All calibration files should be saved in the “Calibration” folder located in the *SIGVIEW's* application data directory (usually C:\Documents and Settings\<UserName>\Application Data\Sigview\). Each of these calibration files can be used later for data acquisition.
There is one special data acquisition file, “default.cal”, which is distributed with *SIGVIEW* and used by default. It leaves recorded data from the DAQ device just as they are, without any conversion. You should not change the content of this file.

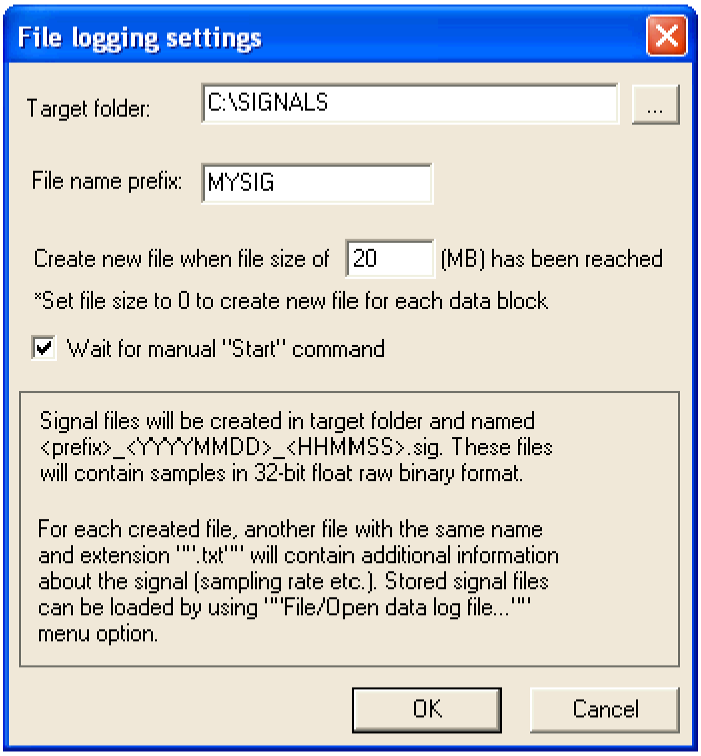
8.3. Logging data to file

Accessible through menu option *Signal tools/Log data in file or Data acquisition/Log data in file*

Data acquisition operations can produce a large amounts of signal data which can not be stored in RAM. This function can be used in such cases to store all signal data from one data acquisition window in files on the hard-disk.

Also, you can use this function for a long-time instrument values logging to a file. See Logging instrument values for details.

After choosing this menu option, following dialog appears:



File logging settings

Target folder: C:\SIGNALS

File name prefix: MYSIG

Create new file when file size of 20 (MB) has been reached

*Set file size to 0 to create new file for each data block

☑ Wait for manual "Start" command

Signal files will be created in target folder and named <prefix>_<YYYYMMDD>_<HHMMSS>.sig. These files will contain samples in 32-bit float raw binary format.

For each created file, another file with the same name and extension "".txt"" will contain additional information about the signal (sampling rate etc.). Stored signal files can be loaded by using ""File/Open data log file..."" menu option.

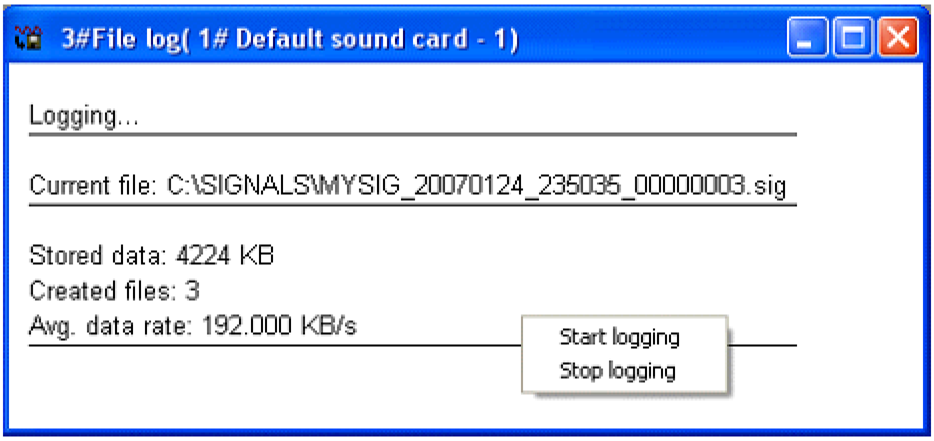
Target folder: Type or choose directory where signals should be stored. The directory should already exist.

File name prefix: To make file names recognizable, you can define prefix string which will be added to the beginning of each created file name.

Create new file when file size of …MB has been reached: You can limit the size of single files. After reaching length you defined here, SIGVIEW will create new file and continue storing data in it. If you set maximal file size to 0, SIGVIEW will create new file for each new block from the origin window.

Wait for manual “Start” command: If you turn this option on, log window will not start storing data after it has been created, but will rather wait on your command. You can manually start and stop logging process by using context menu (right mouse button) options on the logging window.

After you set all properties for signal logging and press OK button on this dialog, log window will appear:



Logging...

Current file: C:\SIGNALS\MYSIG_20070124_235035_00000003.sig

Stored data: 4224 KB
Created files: 3
Avg. data rate: 192.000 KB/s

It shows all information about the logging process including current file, amount of stored data, number of files and average data rate.

Signal files will be created in target folder and named <prefix>_<YYYYMMDD>_<HHMMSS>.SIG.
To make data storage fast enough, data is stored in a 32-bit binary float format without any header information. File extension will always be “*.SIG”. All additional data like sampling rate or start time is stored in a TXT file having the same name as stored SIG file.

To load SIG files in SIGVIEW, please use *File/Open data log file…*. This function will load SIG file and automatically read its additional information from TXT file. You can store your signal in WAV or any other supported format afterwards.

Part IX

Command-line functionality

9. Command-line functionality

9.1. Concept

SIGVIEW supports various command-line parameters enabling you to use some of *SIGVIEW* functions from external applications or batch files. Since this feature is based on simple command-line invocation, it can be used from any other programming language or environment capable of calling other applications.

Typical application examples for command-line invocation would be to perform the same analysis on many different files or to perform some live signal analysis according to a time schedule.

General syntax of command-line invocation is of the form:

Sigview32.exe <action> <parameters>

Each command invocation checks if there is already *SIGVIEW* instance running and if yes, sends command to it. Otherwise, new *SIGVIEW* instance is started and executes the command. To make calling Sigview32.exe easier, you can use environment variable **SIGVIEW_EXE** which contains full path of the *SIGVIEW* executable. For example, in a batch file it would look like this:

%SIGVIEW_EXE% <action> <parameters>

Instead of providing single function for each analysis feature in *SIGVIEW,* command-line interface is based on creating and reusing Sigview WorkSpace files (SWS). This means that you will always create a workspace file first, containing the basics of analysis you would like to control remotely. For more details about creating and using SWS files, please proceed here. First command you will use will probably be loading of SWS file, for example:

%SIGVIEW_EXE% -loadsws c:\myworkspace.sws

After the SWS has been loaded, you will be able use various commands to start/stop analysis, save data to a file etc. For example, to save content of a window to a 16-bit WAV file, you would use:

%SIGVIEW_EXE% -savewav16 window1 "c:\my dir\file1.wav"

To be able to reference single windows from your workspace, as "window1" in above example, you have to assign those a name before saving your SWS file. It is done by using "*Edit/Change Window title*" menu option. Instead of default window name, assign window a name which describes well its usage and can be used in command line calls, for example "daq_window", "myfft1" or "result".

For detailed information about command-line invocation, please see Reference and Examples chapters.

9.2. Command reference

This chapter lists all command-line actions and parameters you can use with *SIGVIEW,* as well as some general rules for using them.

1. General rules valid for all command-line actions:

- You should always use SIGVIEW_EXE environment variable instead of actual *SIGVIEW* executable file name, for example: `%SIGVIEW_EXE% -loadsws c:\myworkspace.sws.` This will make it easier to reuse your scripts on another computer or if your *SIGVIEW* installation changes. This environment variable will be defined automatically during *SIGVIEW* installation.
- For all references to a **<windowname>**, you have to use the window's name as visible in its title bar. Usually, you will use "*Edit/Change window title*" menu option to give a simple name to a window before saving a Workspace or a Tool file.
- Any command parameter which includes spaces, for example "my window" or c:\Program files\myfile.wav" has to be enclosed in quotes (general rule for Windows command line calls)
- All file paths are full paths including directory and file name, for example "c:\my dir\my file.txt"
- You should use only one command in each call. To execute multiple commends, use multiple *SIGVIEW* calls (multiple lines in the batch file).

2. Full list of command-line actions and options:

`%SIGVIEW_EXE% -loadsws <sws_file_name> [<signalfilename> ...]`
Loads Workspace file in *SIGVIEW.* If workspace has been saved without absolute file names in it, you can define those as additional parameters (full file names) in order of their appearance in workspace file. See Loading and saving workspaces for more information.

`%SIGVIEW_EXE% -hide`
Hides *SIGVIEW* window

`%SIGVIEW_EXE% -show`
Shows *SIGVIEW* window again if it was hidden with the -hide command

`%SIGVIEW_EXE% -closeall`
Closes all opened windows

`%SIGVIEW_EXE% -daqstart <windowname>`
Starts data acquisition of a window with the specified name. Of course, window must be Data acquisition window. The effect will be the same as if choosing "Data acquisition/Start data acquisition" main-menu option. If you would like to start multichannel data acquisition, it is enough to call this command for the first channel.

`%SIGVIEW_EXE% -daqstop <windowname>`
Stops data acquisition of a window with the specified name. Of course, window must be Data acquisition window. The effect will be the same as if choosing "Data acquisition/Start data acquisition" main-menu option. If you would like to stop multichannel data acquisition, it is enough to call this command for the first channel.

%SIGVIEW_EXE% -playnotone <windowname>
Plays signal in a specified window without tone output. The effect will be the same as if choosing "Play&Navigate/Play (no sound)" main-menu option.

%SIGVIEW_EXE% -playwithtone <windowname>
Plays signal in a specified window with tone output. The effect will be the same as if choosing "Play&Navigate/Play signal(with sound)" main-menu option.

%SIGVIEW_EXE% -playstop <windowname>
Stops signal playing in a specified window started with -playnotone or -playwithtone options. The effect will be the same as if choosing "Play&Navigate/Stop playing" main-menu option.

%SIGVIEW_EXE% -zoomout <windowname>
Zoom-out signal in a specified window so that the whole signal is visible. It should always be used before storing signal in a file if you would like to save the whole signal. Otherwise, only a visible signal part will be stored.

%SIGVIEW_EXE% -zoomin <windowname> <length_in_samples> [<offsett_in_signal_units>]
Zoom-in signal in a specified window to the specified length (in samples) starting from some offset (in signal units, for example seconds). If offset is omitted, signal beginning will be used. For example, `%SIGVIEW_EXE% win1 256 0.6` will zoom-in signal in window "win1" to 256 samples starting from 0.6s.

%SIGVIEW_EXE% -savewav16 <windowname> <full_file_name>
Saves visible part of the signal in specified window to a 16-bit WAV file. If you would like to save the whole signal in a file, perform -zoomout command first (see above). For example, "`%SIGVIEW_EXE% -savewav16 sig_win_1 c:\MyFiles\test.wav`".

%SIGVIEW_EXE% -savewav32 <windowname>
Saves visible part of the signal in specified window to a 32-bit WAV file (32-bit float format). If you would like to save the whole signal in a file, perform -zoomout command first (see above). For example, "`%SIGVIEW_EXE% -savewav32 sig_win_1 c:\MyFiles\test.wav`".

%SIGVIEW_EXE% -saveascii <windowname>
Saves visible part of the signal in specified window to ASCII file format. ASCII file will contain two columns: one with X values and one with Y values from the signals. If you would like to save the whole signal in a file, perform -zoomout command first (see above). For example, "`%SIGVIEW_EXE% -saveascii sig_win_1 c:\MyFiles\test.wav`".

%SIGVIEW_EXE% -savebitmap <windowname> <bitmap_filename>
Saves graphical content of the <windowname> window as BITMAP file (*.bmp). This will work only if main *SIGVIEW* window is currently visible.

%SIGVIEW_EXE% -getvalue <windowname>
Gets current value from an instrument window with specified window name. Instrument value will be converted in text format, for example "4.325" and forwarded to the standard output. You can use standard batch file syntax to forward that value to a file or to another application. For example, "`%SIGVIEW_EXE% -getvalue instr_max >> myinstrlog.txt`" or

"%SIGVIEW_EXE% -getvalue instr_max | myinstrprocessor.exe".

%SIGVIEW_EXE% -repeatforever on|off
Turns"Repeat forever" play option "on" or "off". This is a global setting valid for all windows.

%SIGVIEW_EXE% -playfast on|off
Turns "Play as fast as possible" play option "on" or "off". This is a global setting valid for all windows.

%SIGVIEW_EXE% -autovolume on|off
Turns "Adjust volume automatically" play option "on" or "off". This is a global setting valid for all windows.

%SIGVIEW_EXE% -closeapp
Closes *SIGVIEW.*

%SIGVIEW_EXE% -waitms <number_of_ms_to_wait>
Waits specified number of milliseconds and returns. It is useful as pause command if your programming language does not support it directly, for example when using batch files. Please note that time resolution of this command is very coarse - you should expect jitter of up to 100ms.

%SIGVIEW_EXE% -help
Shows list of supported commands in a message box.

For further information about using this commands, see Examples section.

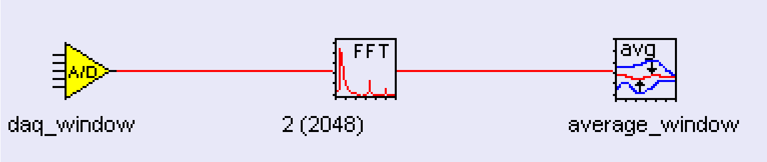
9.3. Examples

This chapter contains several examples for using command-line parameters in *SIGVIEW.* All examples are implemented as MS Windows batch files. You can find the source code of these examples in "Scripts" subdirectory of the *SIGVIEW* installation directory*:*

1. Record 10 seconds of signal and save signal to a file along with its averaged spectrum

In this example, we will create a batch file which starts 10 seconds of data acquisition, saves resulting signal in a file along with its average spectrum.

You should create a Workspace containing data acquisition window (from any device), FFT of data acquisition window and Averager of the FFT window. Control window display should look like this:



Data acquisition window has to have defined acquisition length (10s). You should rename data acquisition window to "daq_window" and Averager window in "average_window". Save this workspace in, for example, c:\test\daq_average.sws.

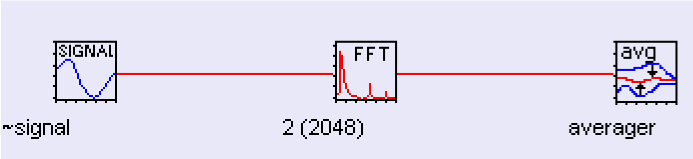
The following batch script will use this workspace to perform 10 seconds of acquisition and save recorded signal and average spectrum at the end.

```
REM close any opened windows
%SIGVIEW_EXE% -closeall

REM load workspace
%SIGVIEW_EXE% -loadsws c:\test\daq_average.sws

REM start data acquisition
%SIGVIEW_EXE% -daqstart daq_window

REM wait 11 seconds - until we are sure that acquisition is finished
%SIGVIEW_EXE% -waitms 11000

REM stop data acquisition
%SIGVIEW_EXE% -daqstop daq_window

REM save average spectrum
%SIGVIEW_EXE% -saveascii average_window c:\test\average_fft.dat

REM zoom-out data acquisition window and save it to WAV file
%SIGVIEW_EXE% -zoomout daq_window
%SIGVIEW_EXE% -savewav32 daq_window c:\test\daq_signal.wav

REM close SIGVIEW
%SIGVIEW_EXE% -closeapp
```

2. Generate averaged spectrum from a signal and store it to a file

In this example, we will create a batch file which can load any signal, generate its average spectrum and save it to a file.

Create a workspace containing a loaded file, zoomed-in to its first 4096 samples, FFT calculated from that signal part, and averager window attached to the FFT. Your workspace should look like this in Control Window:



Window title of a signal window has been changed to "signal" and window title of averager window has been changed to "averager".

Now save this workspace to a SWS file, for example named "sig_average.sws". When *SIGVIEW* asks if you would like to save actual signal file name(s) into workspace file, answer NO.

The following batch script will use this workspace to load any WAV file provided as first parameter of the batch file ("%1" variable), will move through the signal for 10 seconds (-playnotone option) and save resulting FFT average in a file which name is given as a second parameter of a batch file ("%2" variable):

```
REM close any opened windows
%SIGVIEW_EXE% -closeall

REM turn repeat forever option on
%SIGVIEW_EXE% -repeatforever on

REM load workspace
%SIGVIEW_EXE% -loadsws c:\test\sig_average.sws  %1

REM start playing
%SIGVIEW_EXE% -playnotone signal

REM wait 10 seconds
%SIGVIEW_EXE% -waitms 10000

REM save average spectrum
%SIGVIEW_EXE% -saveascii averager %2

REM close SIGVIEW
%SIGVIEW_EXE% -closeapp
```

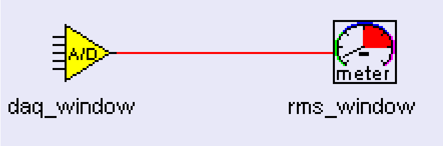
If you save this batch file as "example2.bat", you can call it with, for example, the following command line:

```
example2.bat c:\test\mysignal.wav c:\test\average_fft.dat
```

3. Start data acquisition and log signal RMS energy in file each second

In this example, we will create an endless instrument logger. It will start data acquisition, calculate continuously RMS of the current signal, and save it to a file along with the system time.

Create a workspace containing data acquisition window (acquisition length = 0, any device) and an RMS instrument. Your workspace should look like this in Control Window:



Window title of a signal window has been changed to "daq_window" and window title of RMS instrument window has been changed to "rms_window".

This would be the content of a batch file:

```
REM close any opened windows
%SIGVIEW_EXE% -closeall

REM load workspace
%SIGVIEW_EXE% -loadsws c:\test\daq_rms.sws

REM start data acquisition
%SIGVIEW_EXE% -daqstart daq_window

REM endless loop
:start

REM wait 1 second
%SIGVIEW_EXE% -waitms 1000

REM read value from rms_window and add it to text file
%SIGVIEW_EXE% -getvalue rms_window >> c:\test\values.txt

REM time information after the value
echo   %time% >> c:\test\values.txt

REM loop again
GOTO start
```

Since this is an endless loop, you will have to stop the batch file manually. The result file will look like this:

```
0.011391  20:34:30,35
```

Part X

Registration

10. Registration

10.1. Registration: How?

All *SIGVIEW* licenses cover the Windows 95/98/NT/2000/XP/Vista/win7

Procedure for ordering SIGVIEW:

Online ordering

Visit http://www.sigview.com/order.htm for further info about pricing and ordering options

or go directly to the order page:

https://secure.element5.com/register.html?productid=133912

to order *SIGVIEW* online and secure.

All major credit cards are accepted. Wire transfer or check payments are also available.

What will happen after you order Sigview license?

You will receive an e-mail containing your registration data assembled from combination of your user name, e-mail address and registration key. You should enter these values in the registration dialog (Help/Registration… in menu) and restart *SIGVIEW* once. If your registration was successful, your registration data will be shown in *SIGVIEW* About box (Help/About… in menu).

If your trial period is already expired you will be able to enter registration dialog immediately after starting *SIGVIEW*, before "Trial expired…" message.

If you have any problems or need additional info please send an e-mail to info@sigview.com .

10.2. Registration: Why?

SIGVIEW is not a free software. You are hereby licensed to use *SIGVIEW* for evaluation purposes without charge for a period of 21 days. If you want to use it after the 21 day evaluation period, a registration fee is required. For detailed info, please visit http://www.sigview.com

Why should you do it?

1. You will get your registration key, which enables you to use your copy of *SIGVIEW* after 21-day evaluation period (it will also remove that boring license dialog that appears each time you start *SIGVIEW*)

2. You will get all later minor updates completely free, with huge discount on all new major versions or new products

3. You will get a chance to order *SIGVIEW* improvements or customization

Part XI

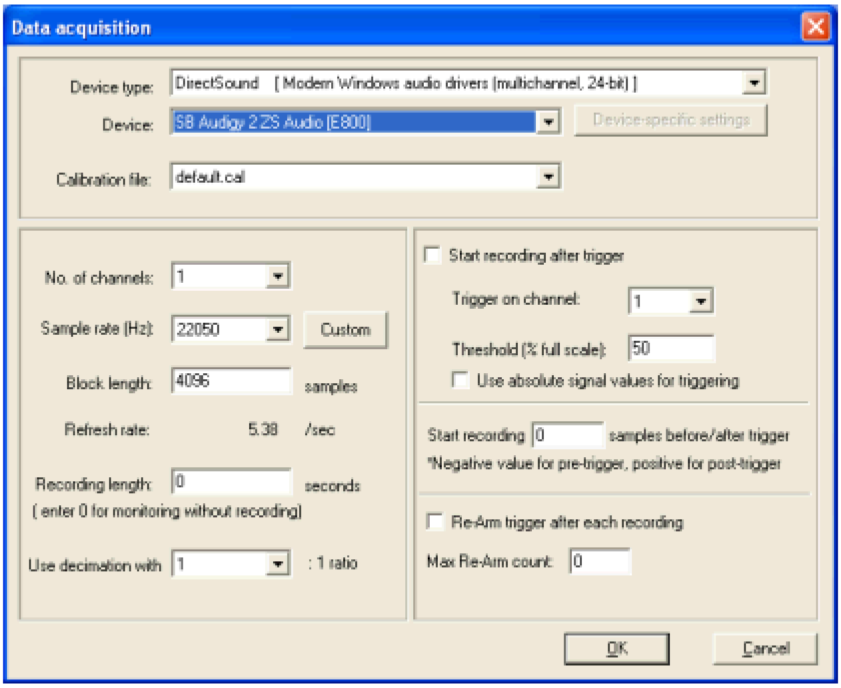
How-To

11. How-To

11.1. Use custom filter curve for better measurement results

Measurement and data acquisition devices like microphones or sound cards have their own frequency characteristics which is added to your signal during recording, i.e. corrupts your signal. By using Custom Filter curve SIGVIEW feature, you can measure these effects and remove them automatically from recorded signals.

1. Let us take sound card device as an example and try to measure its frequency response first. Before you start, you should disconnect all sound input devices like microphones - we want to measure only internal sound card frequency characteristics in this example. Open "Data acquisition" dialog ("Data Acquisition/Open data acquisition" menu item), choose DirectSound as device type and your sound card name as "Device". Leave all other default settings. Press OK and data acquisition signal window will open.



Data acquisition

Device type: DirectSound [Modern Windows audio drivers (multichannel, 24-bit)]

Device: SB Audigy 2 ZS Audio [E800]

Calibration file: default.cal

No. of channels: 1

Sample rate (Hz): 22050

Block length: 4096 samples

Refresh rate: 5.38 /sec

Recording length: 0 seconds
(enter 0 for monitoring without recording)

Use decimation with 1 : 1 ratio

□ Start recording after trigger

Trigger on channel: 1

Threshold (% full scale): 50

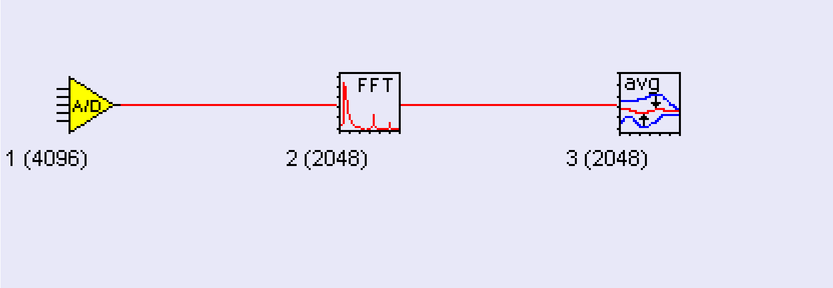
□ Use absolute signal values for triggering

Start recording 0 samples before/after trigger
*Negative value for pre-trigger, positive for post-trigger

□ Re-Arm trigger after each recording

Max Re-Arm count: 0

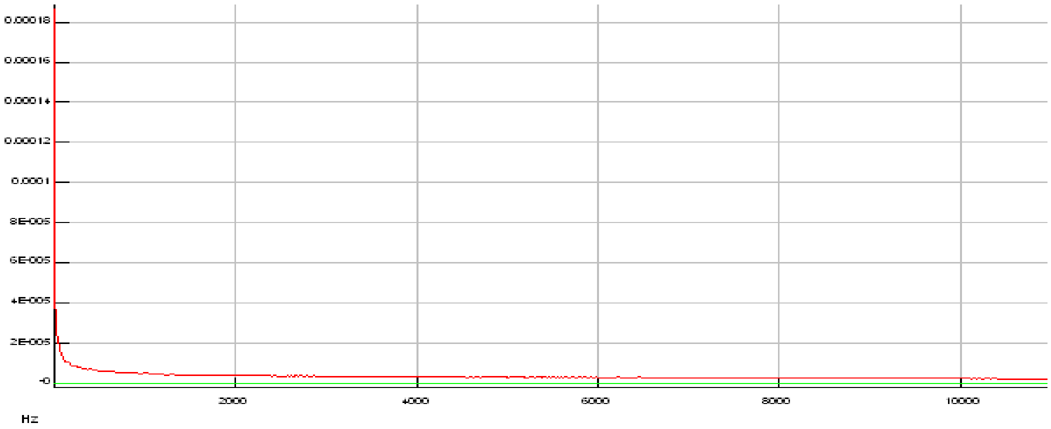
2. While on data acquisition signal window, press FFT button in toolbar to calculate spectrum of a signal and then choose "Signal tools/Averager" option on spectrum window. Open properties of the FFT window (Properties option in Context-Menu) and turn off "Apply Window" option. The result in Control Window should look like this:



3. Now go to data acquisition signal window and choose "Data acquisition/Start" menu option (or REC) button in toolbar. Recording from the sound card will start, signal and spectrum will

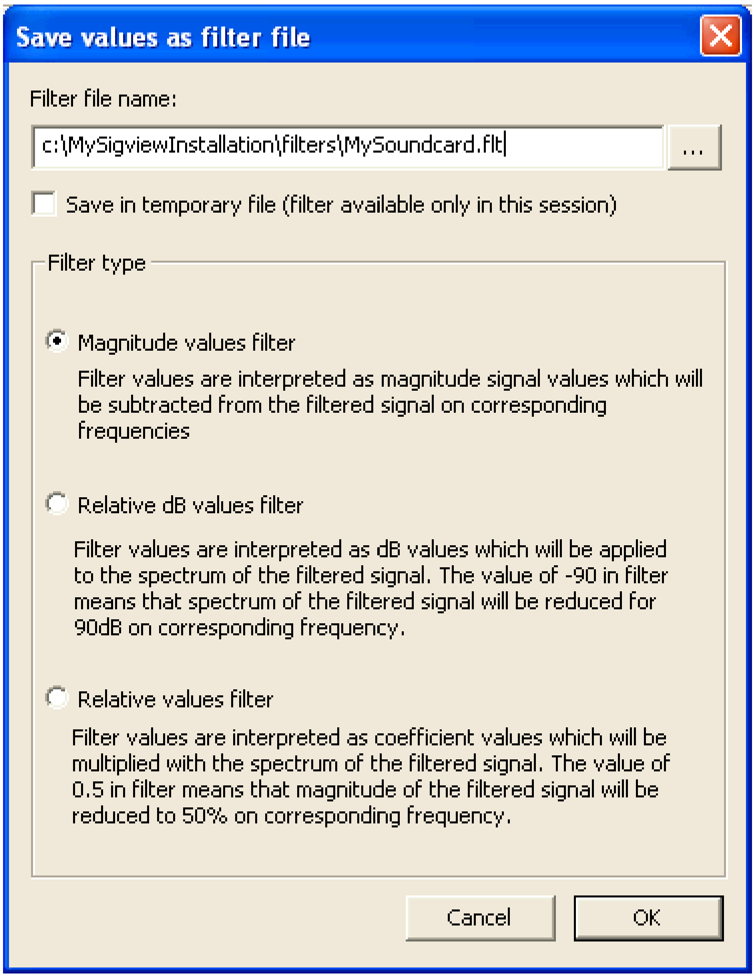
change rapidly and Averager window will calculate average of all spectrums during this session.

4. Wait for ~30s until Averager window shows stable curve and stop the data acquisition. Since we do not use logarithmic (dB) spectrum amplitude values and there is no real signal on the input, do not expect to see too much of the content. It may look like this:



5. This curve is characteristic frequency response of your sound card. Normally, this curve is shown on log scale, but we need it like this so we can convert it to a Custom Filter curve. As a next step, go to Averager window and choose "File/Save as filter curve" option from a menu.

6. In Custom filter curve dialog, enter a name of your filter (for example MySoundcard.flt) and save it to default directory. Choose "Magnitude Values filter" as value type and press OK to save your filter.



Save values as filter file

Filter file name:

c:\MySigviewInstallation\filters\MySoundcard.flt

□ Save in temporary file (filter available only in this session)

Filter type

● Magnitude values filter

Filter values are interpreted as magnitude signal values which will be subtracted from the filtered signal on corresponding frequencies

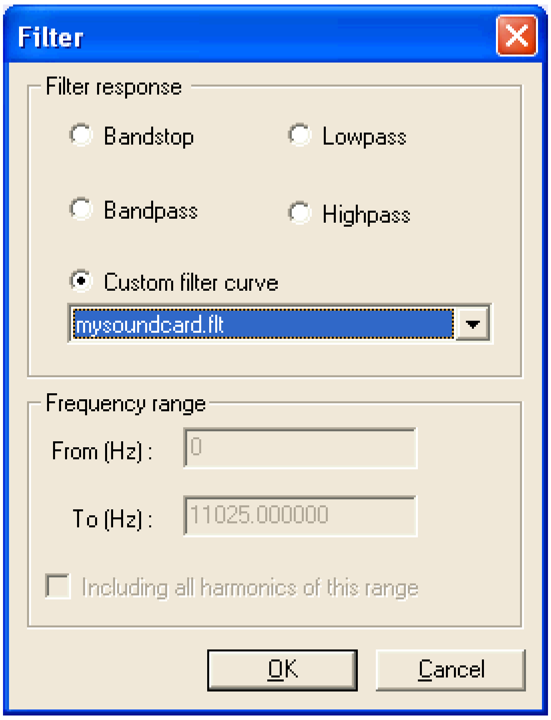
○ Relative dB values filter

Filter values are interpreted as dB values which will be applied to the spectrum of the filtered signal. The value of -90 in filter means that spectrum of the filtered signal will be reduced for 90dB on corresponding frequency.

○ Relative values filter

Filter values are interpreted as coefficient values which will be multiplied with the spectrum of the filtered signal. The value of 0.5 in filter means that magnitude of the filtered signal will be reduced to 50% on corresponding frequency.

7. You can use this saved filter for any data analysis using sound card signal to correct your measurement and remove artifacts caused by the sound card imperfection. To do so, go to your data acquisition signal from step 1 and apply "Signal tools/Filter..." menu option on it. In the dialog, choose your saved MySoundcard.flt filter. It means that we remove from the signal exactly the frequency components introduced by the sound card:



Filter

Filter response

○ Bandstop ○ Lowpass

○ Bandpass ○ Highpass

● Custom filter curve

mysoundcard.flt

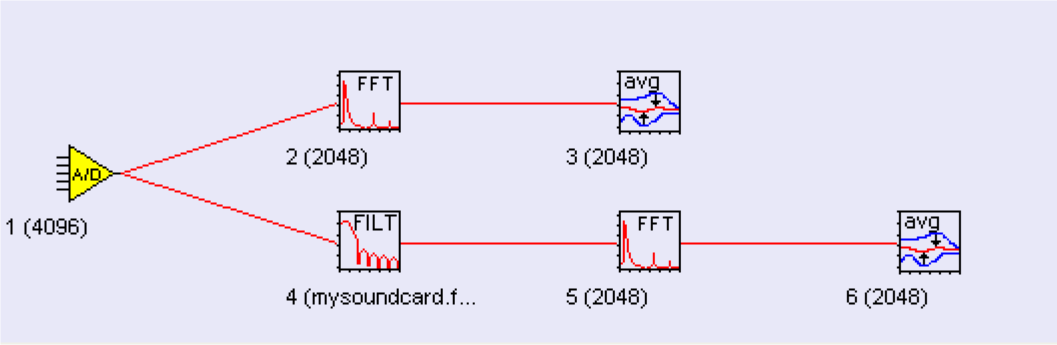
Frequency range

From (Hz) : 0

To (Hz) : 11025.000000

□ Including all harmonics of this range

8. To compare results with the original, apply FFT and Averager on the filtered signal as described above for data acquisition window. The result should look like this in Control Window.



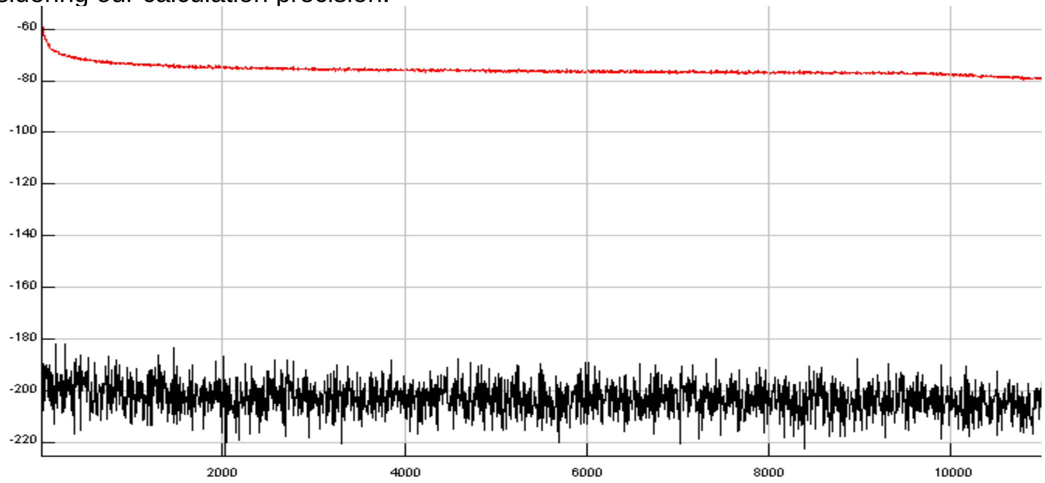
Additionally, open Properties dialog on both FFT windows (Properties option in Context-Menu) and select "Logarithmic Y-Axis (dB units)" option so you can see results better and turn *Apply Window* option off as in the first FFT. Go to both Averager windows and press F3 to reset their content.

9. Start data acquisition again and stop it after ~30s. The second Averager window (number 6 in example) will contain pure spectrum of the input signal with removed sound card characteristics. Since there is no real input signal in this case, we expect this spectrum to be near zero or at least with much lower amplitude then the original.

10. You can use Overlay option to show both resulting averaged spectrums in one window (select both in Control window and choose "Show selected as overlay" from Context-menu).

11. How-To

The result will be a window showing both average spectrums. The red one is original and a black one is filtered. Obviously we achieved to reduce sound card noise for ~130 dB by using custom filter curve. The result is almost perfectly flat curve as close to zero as it gets considering our calculation precision.



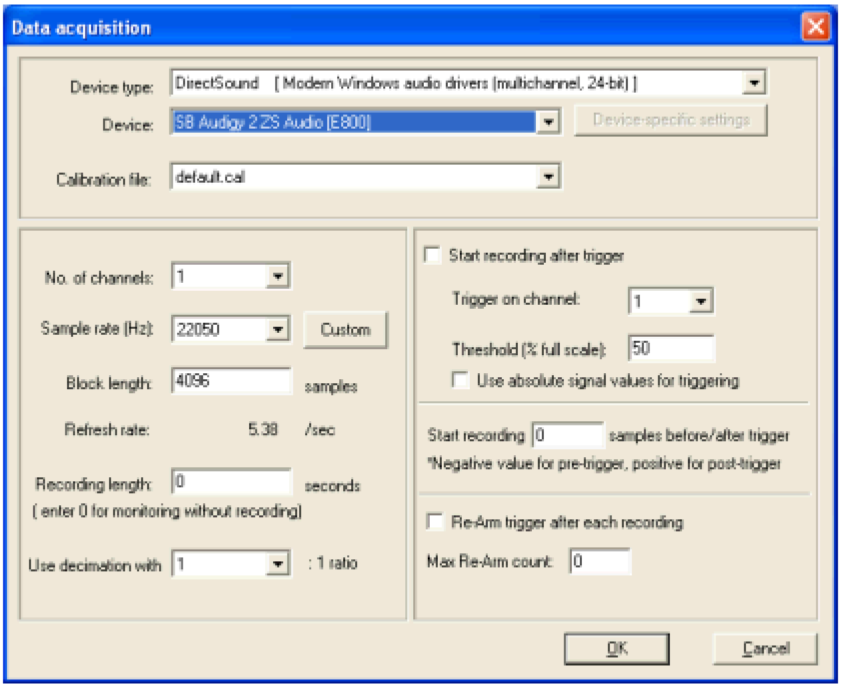
11. You can use this method to calculate frequency characteristics of any other more complex data acquisition system including microphones, sensors etc.

12. Another method to use Custom filter curve is to apply it to a spectrum instead of using it as signal filter. In that case, you see how the spectrum would look like if you would apply custom filter curve as a filter on a signal.

11.2. Track changes of signal parameters through time

We will show you here how to track changes of some signal parameters during data acquisition or monitoring.

1. Let us take sound card device as an example. You can connect some signal source to it before you start (microphone). Open "Data acquisition" dialog ("Data Acquisition/Open data acquisition" menu item), choose DirectSound as device type and your sound card name as "Device". Leave all other default settings. Press OK and data acquisition signal window will open.



Data acquisition

Device type: DirectSound [Modern Windows audio drivers (multichannel, 24-bit)]

Device: SB Audigy 2 ZS Audio [E800]

Calibration file: default.cal

No. of channels: 1

Sample rate (Hz): 22050

Block length: 4096 samples

Refresh rate: 5.38 /sec

Recording length: 0 seconds
(enter 0 for monitoring without recording)

Use decimation with 1 : 1 ratio

□ Start recording after trigger

Trigger on channel: 1

Threshold (% full scale): 50

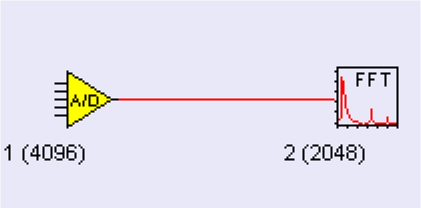
□ Use absolute signal values for triggering

Start recording 0 samples before/after trigger
*Negative value for pre-trigger, positive for post-trigger

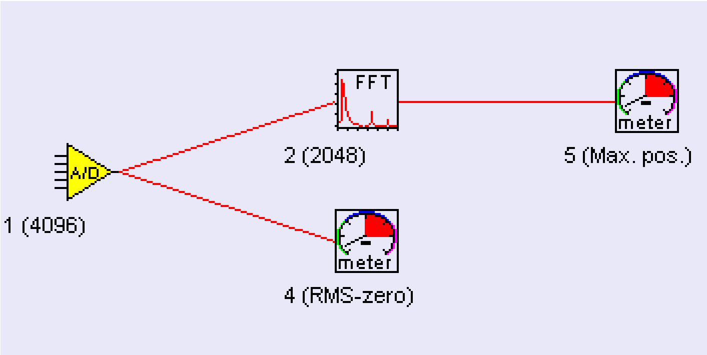
□ Re-Arm trigger after each recording

Max Re-Arm count: 0

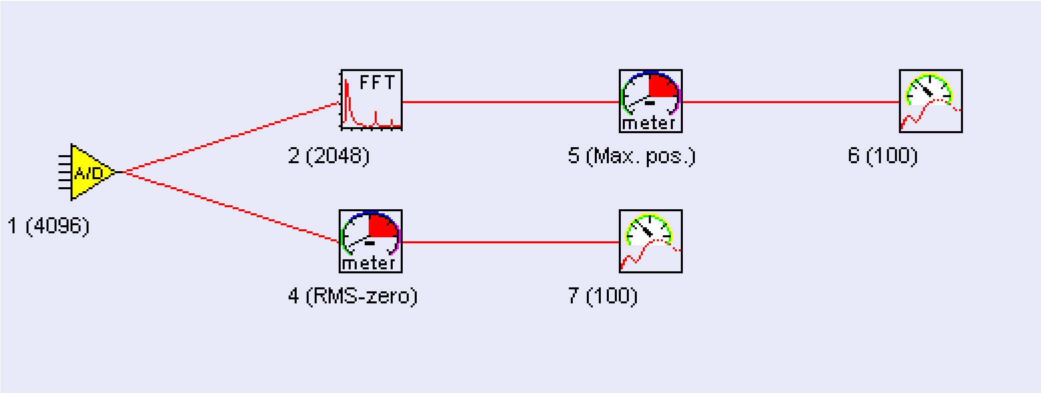
2. While on data acquisition signal window, press FFT button in toolbar to calculate spectrum of a signal. Spectrum window appears and Control Window should look like this:



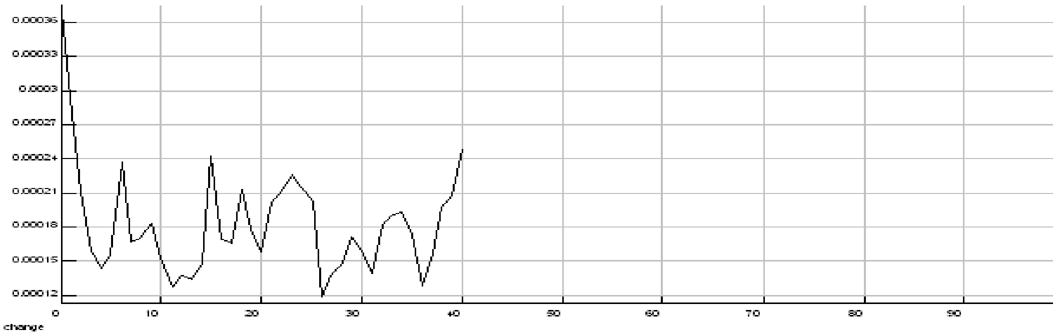
3. As an example, we will track changes in signal RMS energy and the position of the peak spectrum frequency. Click on signal acquisition window and select "Instruments and markers/RMS" main menu option. Then click on FFT window and select "Instruments and markers/Maximum position (with marker)" main menu option. Two instrument windows appear showing current values from the signal and spectrum (all zeros because data acquisition is not running yet) The resulting Control Window should look like this:



4. Now we would like to track (log) instrument values as they change through time. To do this click on each of two instrument windows and select "Log instrument values" from their right-click menu. In the small dialog asking you about buffer length for a log, leave default value of 100. Control window should look like this:



5. Now go to data acquisition signal window and choose "Data acquisition/Start" menu option (or REC) button in toolbar. Recording from the sound card will start and instruments will change their values. On each instrument value change, the last value will be stored in a corresponding log window - log window signal will grow as new values come.



6. Note that X-axis units on log window is "change" - it is simply an index of a value change starting from 0.

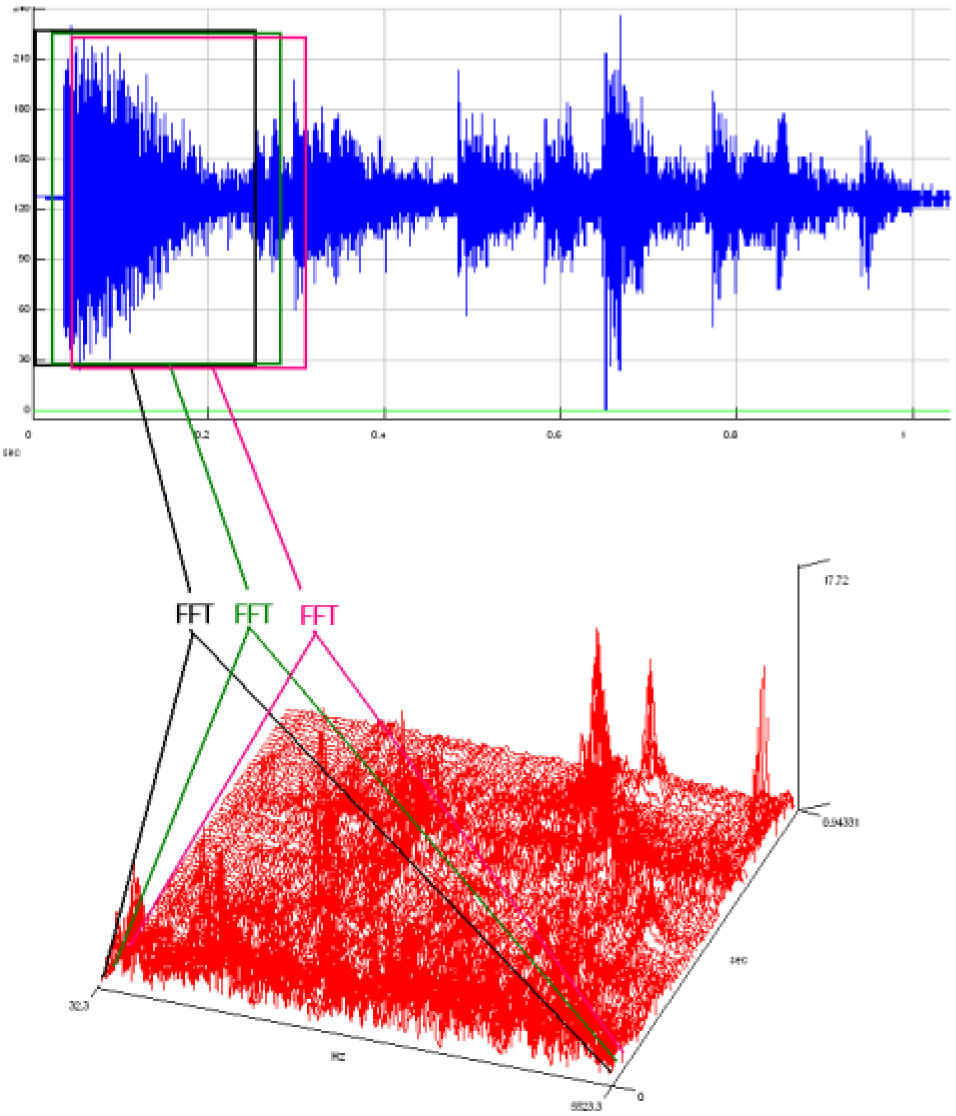
7. After 100 changes (remember your setting for buffer length), log windows will be full and

display will start scrolling. You can use context menu to reset data (i.e. start from the beginning) or to stop/start logging. Value log signal will try to determine values for its X-axis automatically by backtracking through the analysis chain. If those X values are equidistant (i.e. difference between two consecutive values is always the same), those will be display on X-axis. If not, X-axis will switch to unit called "change" where each new value will simply be labeled with its index, i.e. 0,1,2,3,4,...

8. Instrument log windows can be used as any other signal window, i.e. you can calculate its spectrum, show them in overlay with other signals, store values in ASCII file etc.

11.3. Track analysis results through time as 3D graphics

One of the most popular analysis methods for time/frequency analysis is a Time-FFT. You can perform this analysis in *SIGVIEW* automatically to see how signal spectrum changes through time. The resulting graphics is a 3D graphics with time and frequency as X/Y axes and signal amplitude on Z-axis. This function simply divides your signal into possibly overlapping segments, calculates FFT for each segment and puts all FFT results next to each other in a 3D graphics.



1. In this How-To, we will show you how to apply the same concept to other analysis functions and to see how, for example, probability distribution of a signal changes through time. Of course, you can use it to track any other function, for example autocorrelation, integral curve, ...

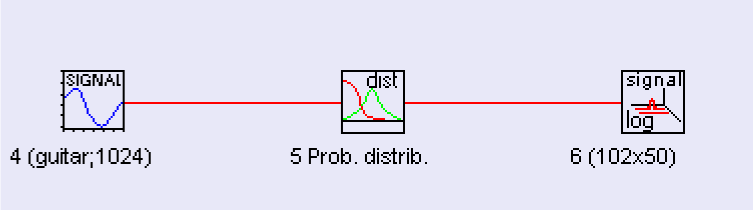
2. As a first step, you should load one signal in *SIGVIEW*, for example guitar.wav from Examples subdirectory of *SIGVIEW* installation directory.

11. How-To

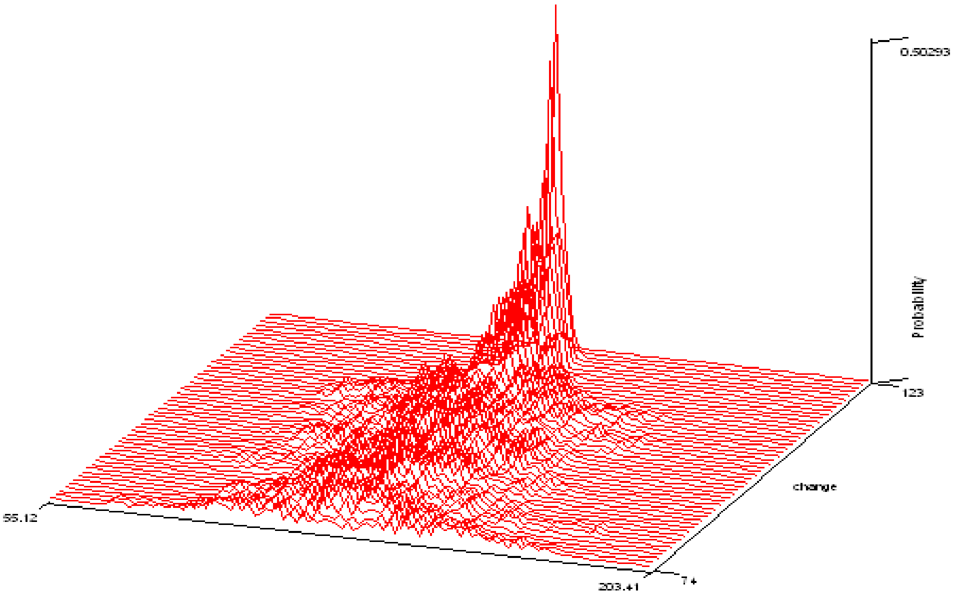
3. Zoom-in to a smaller part of the loaded signal by using "Edit/Zoom to X samples/values" menu option. Choose 1024 samples length.

4. Use "Signal tools/Probability distribution curve" menu option to calculate distribution curve for zoomed signal part.

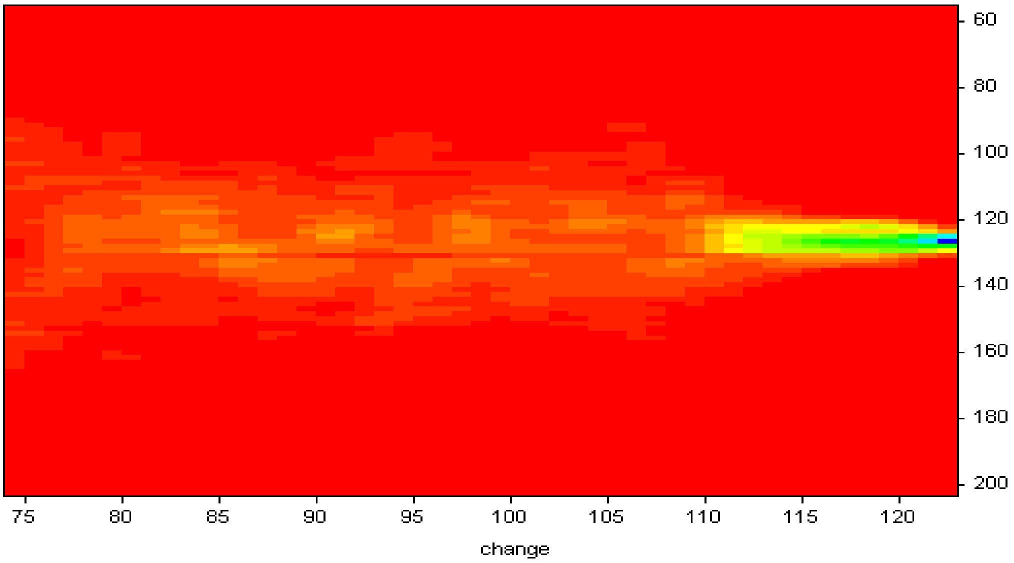
5. Click on new Distribution curve window and select "3D Tools/Track changes as 3D graphics..." from main menu. In the settings dialog leave default value of 50 for "Last changes to track option". That will be number of columns in your 3D graphics. The resulting Control Window should look like this:



6. Click on signal window and then click on "Play" button in toolbar (Play & Navigate/Play (no sound) menu option). Your signal part will move through the whole signals, distribution curve will change and add new column in 3D graphics on each change. The speed of moving is determined by the Step property of the window which can be changed by using Play & Navigate/Step change... option". At the end, last 50 changes of the curve will be stored in a 3D graphics. The result will look like this:



or like this if you switch your 3D window in "Spectrogram view" mode:



7. Note that X-axis units on 3D graphics is "change" - it is simply a column index starting from 0

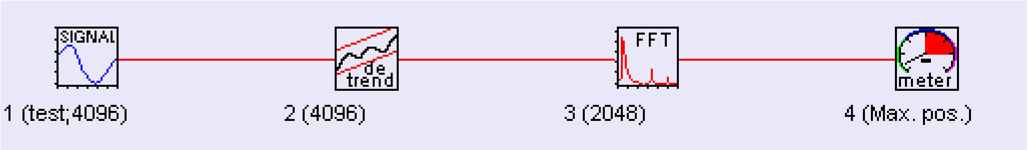
8. By using combination of 3D-ruler and Ruler extraction options, you can even use this graphics to restore each of columns as 2D graphics again.

11.4. Using custom tools and Control Window for batch operations on files

If you need to perform the same analysis very often and/or on multiple input files, you should learn more about custom tools and Control window. Both tools will enable you to perform such operations much faster.

1. Let us take an example of a signal where you would to remove linear trend (*Signal tools/Remove linear trend* menu option) from the signal, perform FFT on it with certain parameters and find frequency of maximal amplitude (*Instruments and markers/Maximum position* menu option) in the resulting spectrum.

2. As a first step you can load any signal and perform these operations manually on it. The resulting Control Window chain should look like this:



Please be sure that you set all options on the FFT you would like to use (context menu/Properties...), for example logarithmic Y-axis, smoothing,...

3. Now, go to first analysis window (Detrend) and choose *File/Save Window as custom tool/Window and its subtree* menu option. In a file save dialog, choose some name for your tool (for example MyTool.swt) and save it in a Tools directory. This tool will contain your analysis chain with Trend removal, FFT and Max. instrument windows.

4. Next time you would like to apply this tool to some other signal, simply go to signal window and choose *Signal tools/Use custom tool/MyTool* (or any other name you gave to it). The whole analysis chain will be applied to your signal and you should have a picture similar to above example in your Control Window.

5. So, this will speed up operations on signal signals. But if you need to perform the same analysis on dozens of signals at the same time, there is a way to perform one tool on all windows selected in the Control Window. Simply make your selection and while Control Window is active, choose *Signal tools/Use custom tool/MyTool* menu option just as if you are working with a single signal. Your tool will be applied to all selected windows.

6. One more hint: Applying above tool on 10 signals will produce 30 new windows and make your workspace quite complicated. If you just need to see last instrument while FFT and detrend are not interesting you can hide those before saving the tool. In the above Control Window chain, select windows 2 and 3 and choose "Hide selected windows" from context menu. Both windows will be hidden and Control Window will show "H"sign next to their icons. This property will be saved in a tool file and applied each time you apply your tool. As a result you would get only a 10 new instrument windows and Control Window will look like this:

11.5. Using Workspaces as analysis templates

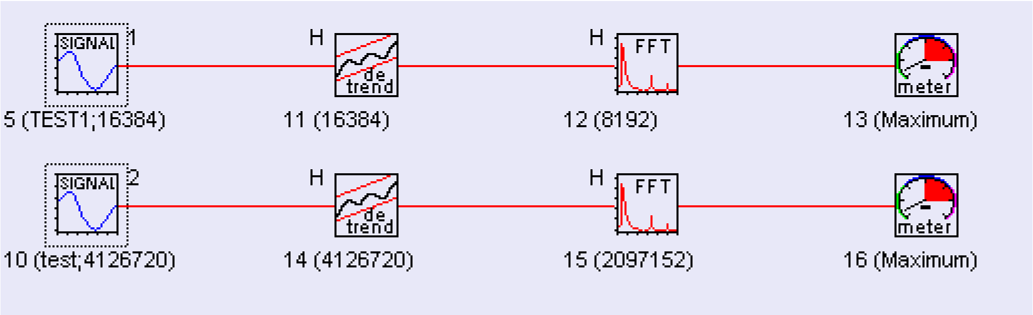
If you are saving a Workspace (SWS file) containing windows with file-based signals (for example, loaded WAV files), *SIGVIEW* will offer you two options:

1. Save full file names in a file: Each time you load the workspace, your files (if they still exist) will be automatically loaded. The result will be exactly the same as the workspace you saved.

2. Do not save file name information in a file: Saved workspace will be used as a template for operations on any files. Each time you try to load the workspace, *SIGVIEW* will offer you a file load dialog to choose a file which should be loaded for each window from the workspace which contained file-based signal.

The second option can be used to create templates which can be used on different files, similar to Custom tools. The main difference is that a Workspace can contain much more complex analysis system as well as information about the settings of the signal window (for example zoom-in length, position, etc.).

1. Let us take an example very similar to a previous How-To, Custom tools usage. The resulting Workspace with two analyzed files looked like this:



2. Save this Workspace to a file, for example MyWorkspace.sws (*File, Save Workspace...* menu option). In the message box about the file name saving, answer with No.

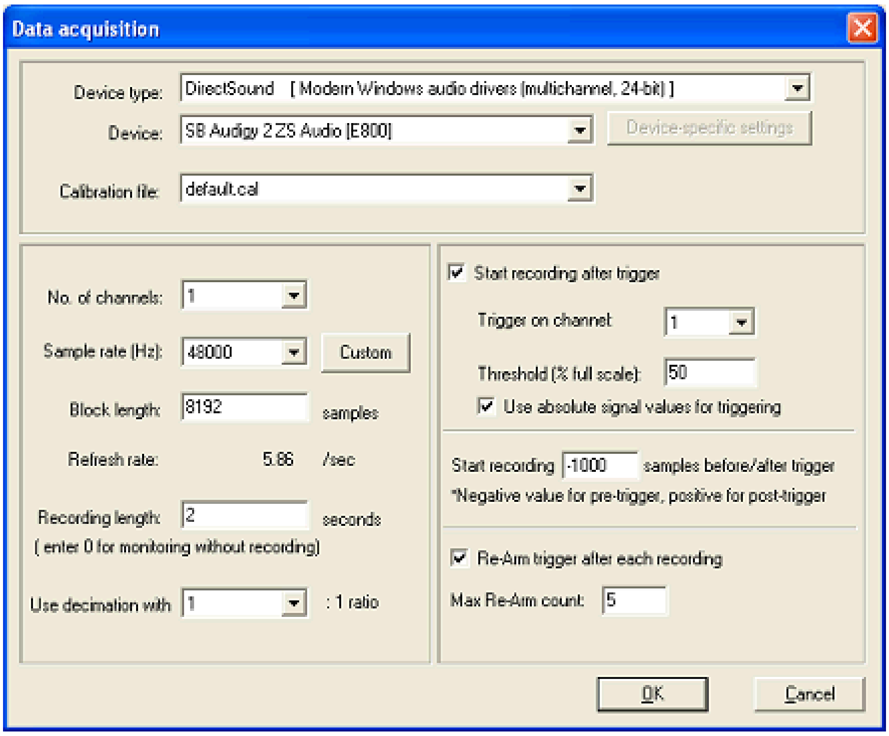
3. Now, close all windows and try to load your workspace again (*File/Load Workspace...* menu option). *SIGVIEW* will ask you twice for the file name because you had two file-based signal windows in your Wokspace. The title of file open dialog will show text "Choose file instead of: File_X" where X is ID of the window in the original Workspace (5 and 10 in our example). After you choose both files, analysis windows will be applied to them just as in your original Workspace.

4. If you would like to use above Workspace for analysis of only one file, simply press Cancel button in the Open file dialog for a second file.

11.6. Using data acquisition triggering options

SIGVIEW includes a comprehensive set of triggering options available for data acquisition from all supported devices. In this example, you will learn how to use these options to record multiple repeating events and use these recordings for further analysis. The events you are trying to record can be a hammer impact during vibration analysis or a sound from a musical instrument or anything else.

1. We will start by opening a Data acquisition dialog and performing some settings:



Data acquisition

Device type: DirectSound [Modern Windows audio drivers (multichannel, 24-bit)]

Device: SB Audigy 2 ZS Audio [E800]

Device-specific settings

Calibration file: default.cal

No. of channels: 1

Sample rate (Hz): 48000

Custom

Block length: 8192 samples

Refresh rate: 5.86 /sec

Recording length: 2 seconds

(enter 0 for monitoring without recording)

Use decimation with 1 : 1 ratio

☑ Start recording after trigger

Trigger on channel: 1

Threshold (% full scale): 50

☑ Use absolute signal values for triggering

Start recording -1000 samples before/after trigger

*Negative value for pre-trigger, positive for post-trigger

☑ Re-Arm trigger after each recording

Max Re-Arm count: 5

The device type, sampling rate, block length are not important - you can set those as needed for your application.

"Start recording after trigger" options is turned on. This means that *SIGVIEW* will start recording only after certain signal amplitude has been detected in the signal. The type of trigger is defined by "Trigger on channel", "Threshold" and "Use absolute..." options. In this example, *SIGVIEW* will start recording if absolute signal level on Channel 1 increases over 50% of the maximal signal amplitude (full scale).

You can also define offset for the recording start relative to the trigger position. In this example, we instruct *SIGVIEW* to start recording 1000 samples before the position where trigger was detected. You can define positive or negative offsets, i.e. start recording before or after the trigger. The maximal offset length is limited to the "Block length" number of samples.

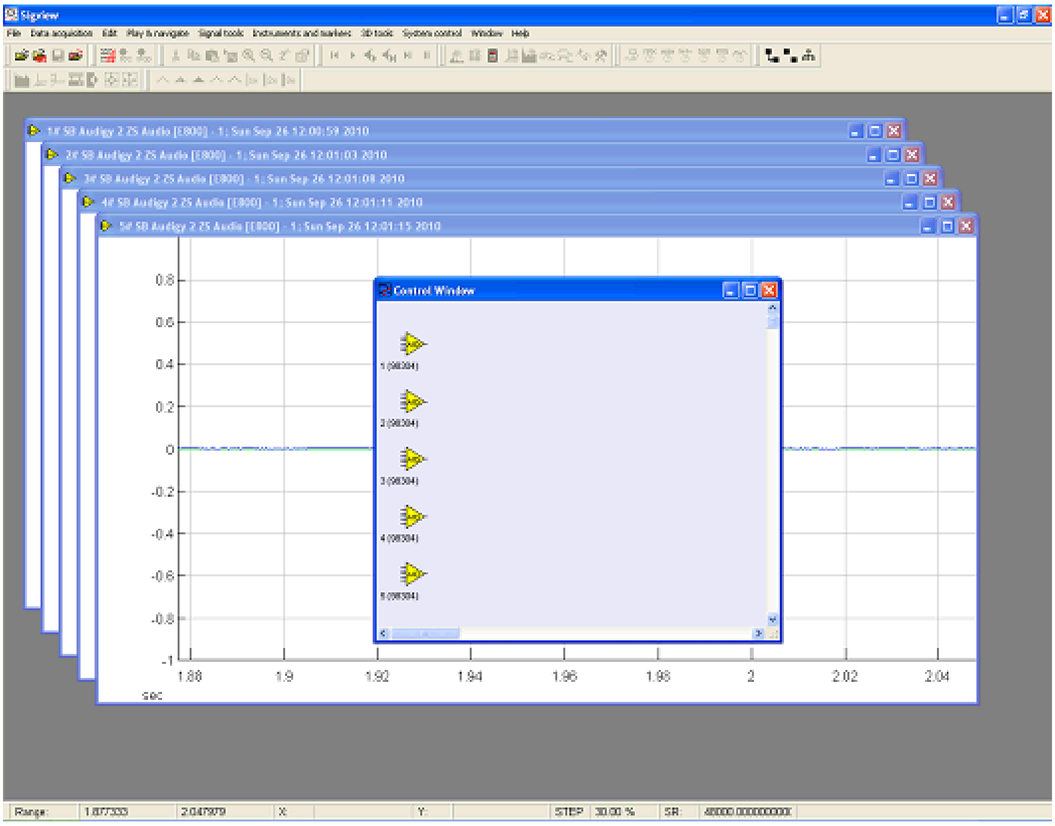
After a trigger is detected in the incoming signal, *SIGVIEW* will start recording for "Recording length" duration, in this example 2 seconds. You should set this parameter to be bigger then the maximal duration of your event.

"Re-Arm..." trigger option instructs *SIGVIEW* to start waiting for a new trigger as soon as one recording has been made. After *SIGVIEW* detects a trigger for the first time and records 2 seconds of a signal it will automatically open new data acquisition window with the same settings and will wait for another trigger in it. That way, each 2 seconds recording will remain in a separate window. You can define maximal number of such windows with the "Max Re-Arm count".

2. Press OK in the data acquisition dialog and a signal window will open. Press "Start button" in a toolbar to start data acquisition, i.e. waiting for a trigger. Since there is still no signal, nothing will happen immediately. In the background, *SIGVIEW* is acquiring your signal and testing if the trigger has occurred.

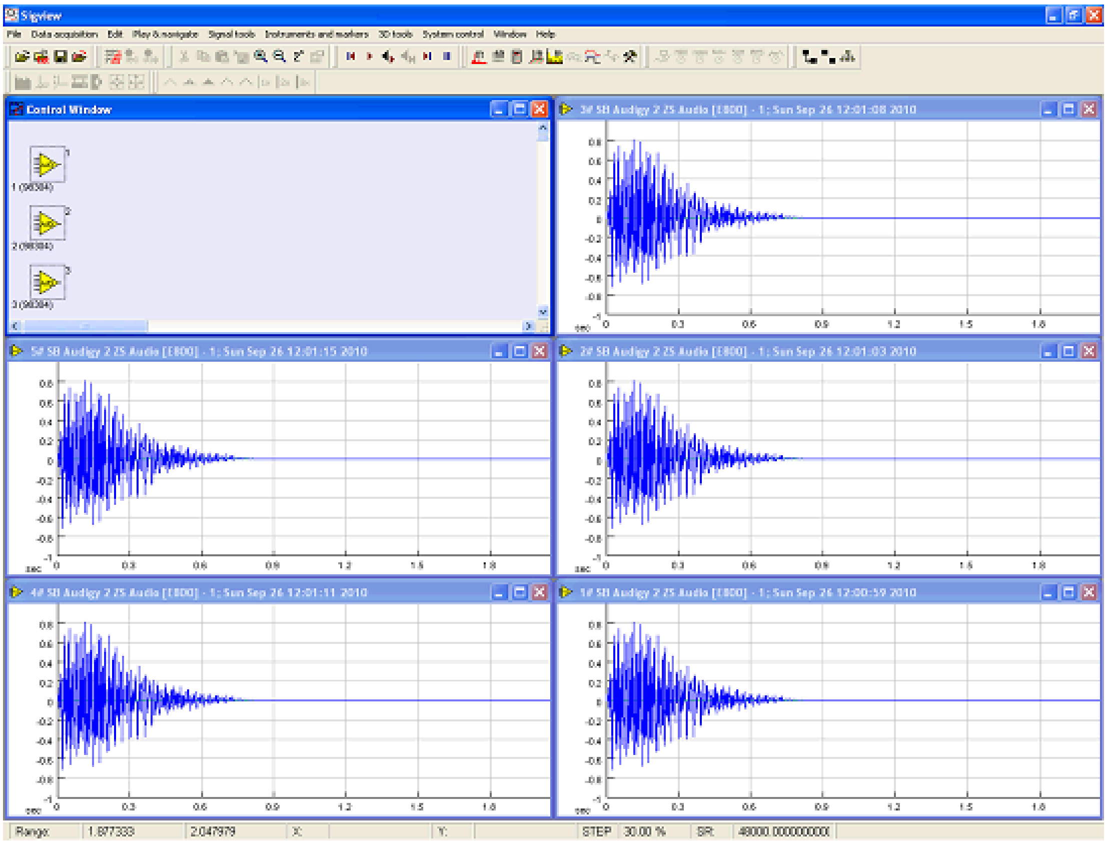
3. Now, trigger the signal event by playing some sound, impacting with a hammer or anything else causing the energy to be detected in the signal. *SIGVIEW* will start recording 1000 samples before the event and the recording will last 2 seconds.

4. As soon as the recording has finished, new window will open waiting for a next trigger. Please repeat this 5 times to get 5 windows with recordings. Your *SIGVIEW* window should look like this:


Sigview
File
Data acquisition
Edit
Play & navigate
Signal tools
Instruments and markers
3D tools
System control
Window
Help
1# SB Audigy 2 ZS Audio [E800] - 1: Sun Sep 26 12:00:59 2010
2# SB Audigy 2 ZS Audio [E800] - 1: Sun Sep 26 12:01:03 2010
3# SB Audigy 2 ZS Audio [E800] - 1: Sun Sep 26 12:01:08 2010
4# SB Audigy 2 ZS Audio [E800] - 1: Sun Sep 26 12:01:11 2010
5# SB Audigy 2 ZS Audio [E800] - 1: Sun Sep 26 12:01:15 2010
Control Window
1 (98304)
2 (98304)
3 (98304)
4 (98304)
5 (98304)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
1.88
1.9
1.92
1.94
1.96
1.98
2
2.02
2.04
sec
Range:
1.877333
2.047979
X:
Y:
STEP
30.00 %
SR:
48000.000000000

11. How-To

5. Now, use Window/Tile option to arrange all window. Select all signal windows in Control Window (Context menu/Select all window) and choose Zoom-Out function in a toolbar to see the complete content of all signal windows. The *SIGVIEW w*indow will look like this:


Sigview
File
Data acquisition
Edit
Play & navigate
Signal tools
Instruments and markers
3D tools
System control
Window
Help
Control Window
1 (98304)
2 (98304)
3 (98304)
3# SB Audigy 2 ZS Audio [E800] - 1; Sun Sep 26 12:01:08 2010
5# SB Audigy 2 ZS Audio [E800] - 1; Sun Sep 26 12:01:15 2010
2# SB Audigy 2 ZS Audio [E800] - 1; Sun Sep 26 12:01:03 2010
4# SB Audigy 2 ZS Audio [E800] - 1; Sun Sep 26 12:01:11 2010
1# SB Audigy 2 ZS Audio [E800] - 1; Sun Sep 26 12:00:59 2010
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
sec
0
0.3
0.6
0.9
1.2
1.5
1.8
Range:
1.877333
2.047979
X:
Y:
STEP
30.00 %
SR:
48000.000000000

6. While still in Control Window, you can apply some analysis to all 5 recordings at the same time, for example FFT. Also, you can show all recordings in an overlay window by using corresponding options from a Control window's context menu.

KEYWORD INDEX

End