[#] Aurora 4

Aurora 4 is a complete system for measuring electro-acoustical impulse responses and performing auralization through headphones and/or loudspeakers.

The Aurora package is based on a series of additional modules ("plugins") which are attached to a standard multichannel sound editor, namely Adobe Audition (any version since 1.0 and above).
Aurora 4.3 currently contains the following modules:

TIM filter
Linear Convolution with Clipboard
Warped Convolution with Clipboard
Generate Multi MLS Signal
Deconvolve Multi MLS Signal
Generate IRS Signal
Deconvolve IRS Signal
Generate SWEEP Signal
Synchronous Average
IR Selection
Acoustical Parameters
Inverse Filter
Flatten Spectrum
Kirkeby's Inverse Filter
Cross Functions
ITU P56 Active Speech Level
Speech Transmission index STI

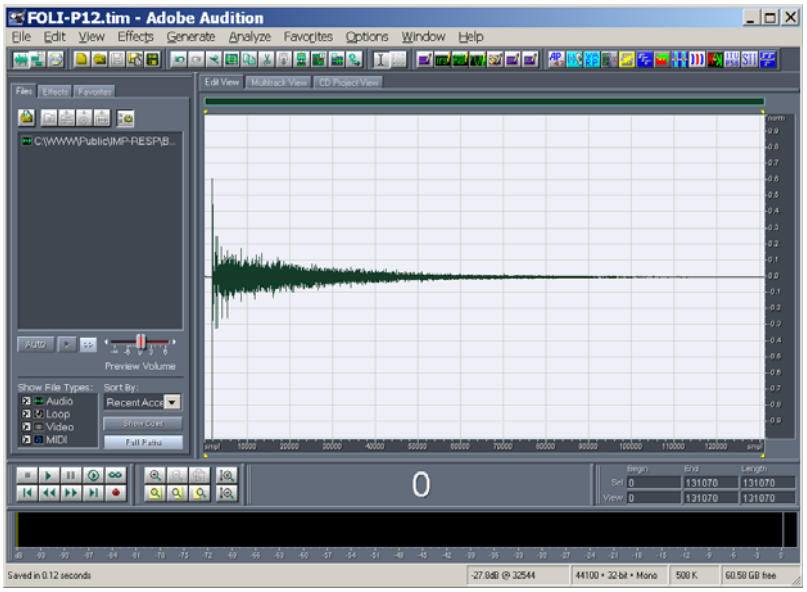
The following obsolete modules are still available, albeit they are not anymore supported under release 4.3 and subsequent of Aurora:

RT Convolver
Generate MLS Signal
Deconvolve MLS Signal
Acoustical Parameters 32
Advanced Frequency Analysis

[#] FIRSTPAGE

$ K MLSSA (.TIM) Import/export filter

The MLSSA (pronounced "Melissa") system is a dedicated hardware and software tool, originally developed by Douglas Rife as a measurement tool for loudspeaker evaluation and room acoustics. It makes use of a special full-sized ISA board (named A2D160), which includes a hardware MLS-signal generator. The board has only one analog input, but it can generate the MLS signal simultaneously with the acquisition. This way the Impulse Response between the excitation signal and the microphone connected with the input is measured, and this includes both the loudspeaker Transfer Function and the reverberation of the room.
This board is accompanied by a very powerful and versatile software (MLSSA.EXE). Although this is an old DOS-based application, it is still nowadays an invaluable tool for making computations and extracting information from Impulse Responses.
The MLSSA .TIM file format is quite strange: the import/export filter for Audition makes it possible to manage .TIM files exactly as if they were .WAV files: this can be useful for importing in Aurora Impulse Responses measured with the MLSSA system, or, on the other hand, for exporting to MLSSA Impulse Responses measured with Aurora.
For example let we start with an Aurora-measured Impulse Response. If it was stereo, we have first to separate it in two mono files, because MLSSA is a single-channel system:

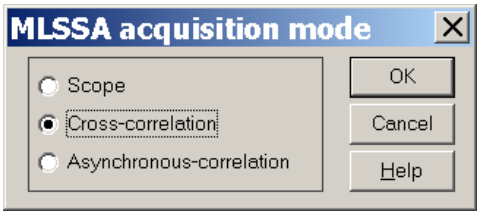


This file was already saved in MLSSA .TIM format, as it shown from its name.
The only point to be noted here is that, when saving the file, an "Options" button appears in the Common Dialog Box. Pressing it, the following Form is displayed:

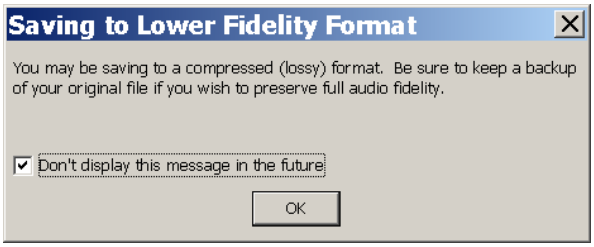
FILTRO_TIM

$ .TIM file read/write filter

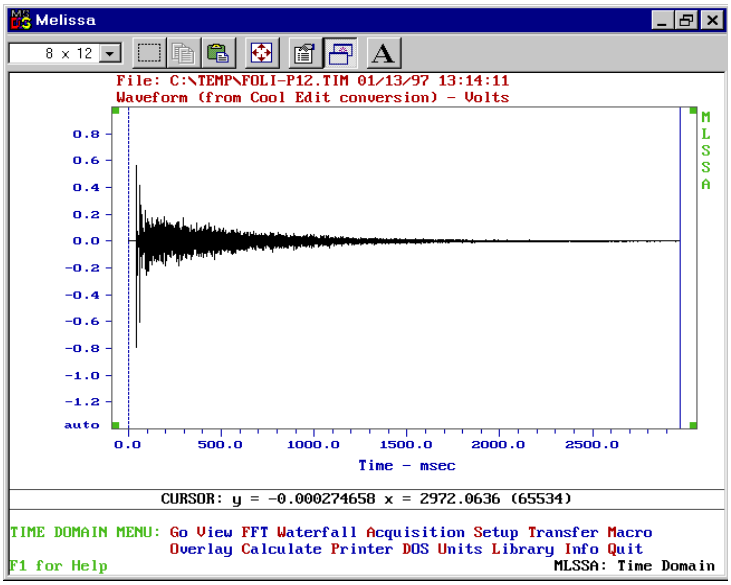
K TIM Filter



This makes it possible to select the MLSSA acquisition mode which is flagged inside the .TIM file: this way the user can "fool" MLSSA, changing the acquisition mode after a measurement is made, thus making it possible to perform computations that MLSSA does not allow for certain kinds of data (for example computing the Acoustical Parameters from Scope data).
When saving, the following warning message is displayed by Audition:



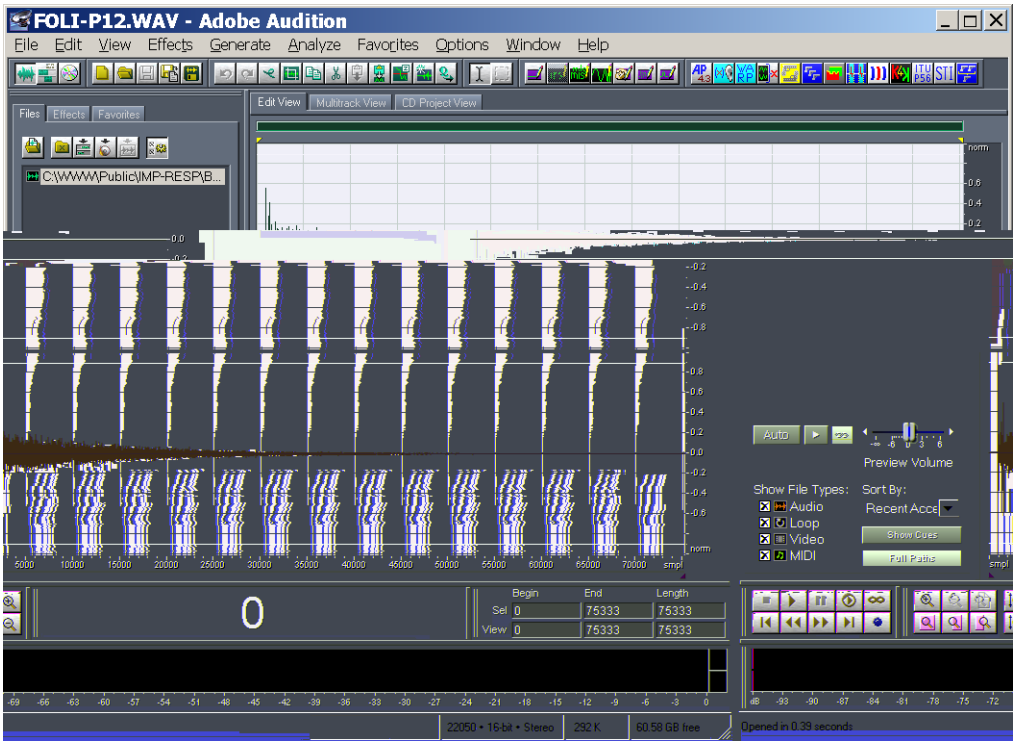
This warning is not true: the TIM format is not lossy, nor compressed: it preserves the complete information contained in the waveform, including true 32-bits (float) depth. The only case in which some data is lost is when the Impulse Response to be saved is longer than 65536 samples, because this is the maximum length allowed by the MLSSA program.
Starting the MLSSA program and opening the .TIM file, it looks properly converted:



[#] [$] [K] Convolve with Clipboard

This module performs linear convolution of the currently selected waveform with the waveform actually present in the Windows Clipboard. This means that, before using this module, You have to load in the Windows Clipboard (not in the Audition Clipboards !!!) the Impulse Response that You want to use as a filter.
So, the first thing to do is to select the Windows Clipboard (Audition defaults to the usage of its internal clipboard n.1): Edit – Set Current Clipboard – Windows'

Now we can start opening the Impulse Response file:

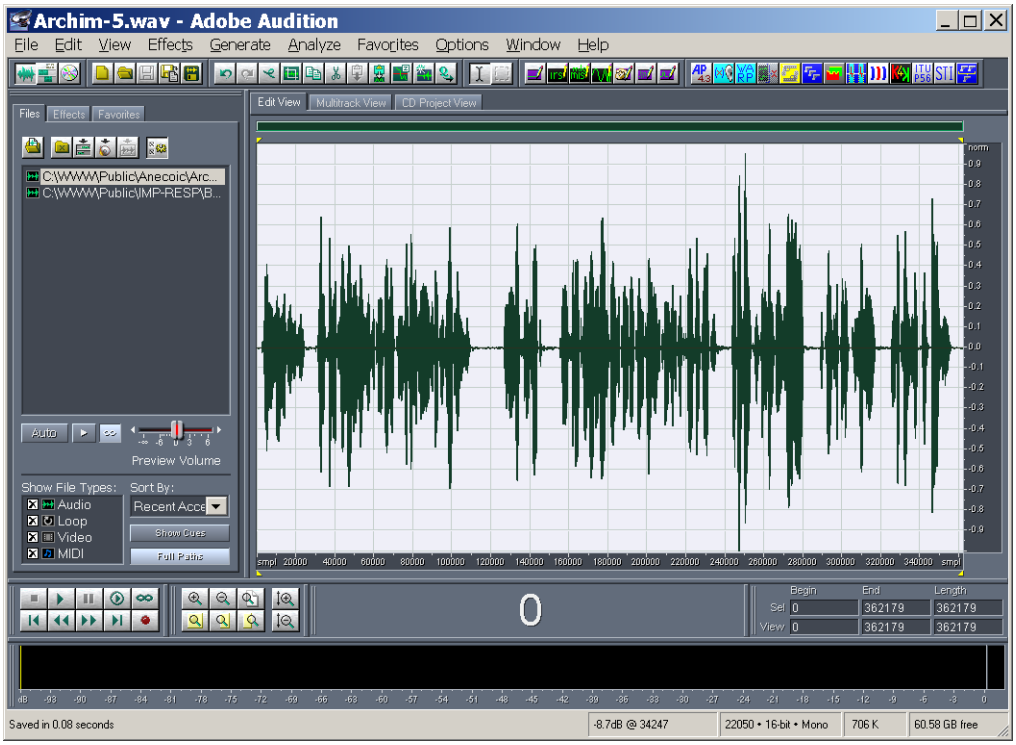


Now we copy this binaural stereo IR to the Windows Clipboard (Edit – Copy).
Then we can load an anechoic signal:

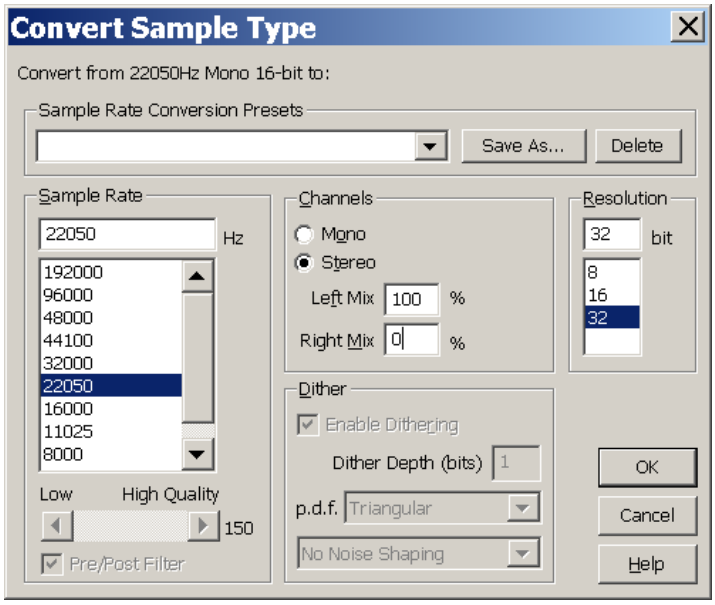
[#] CONVOLVE

[$] Convolve with Clipboard

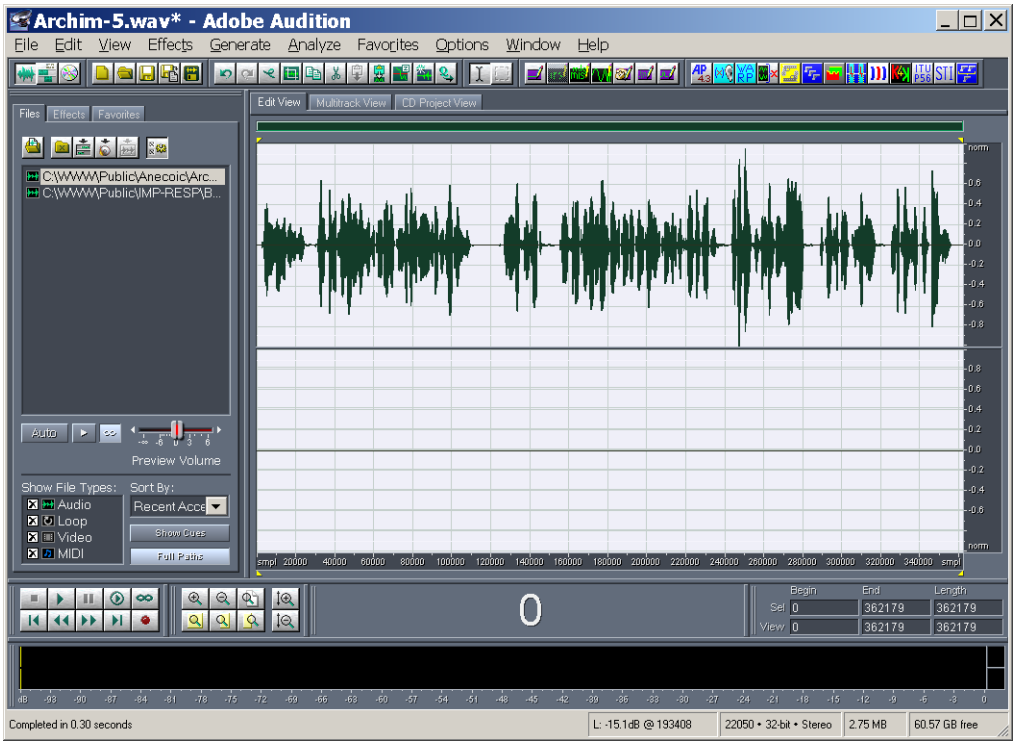
[K] Convolve with Clipboard



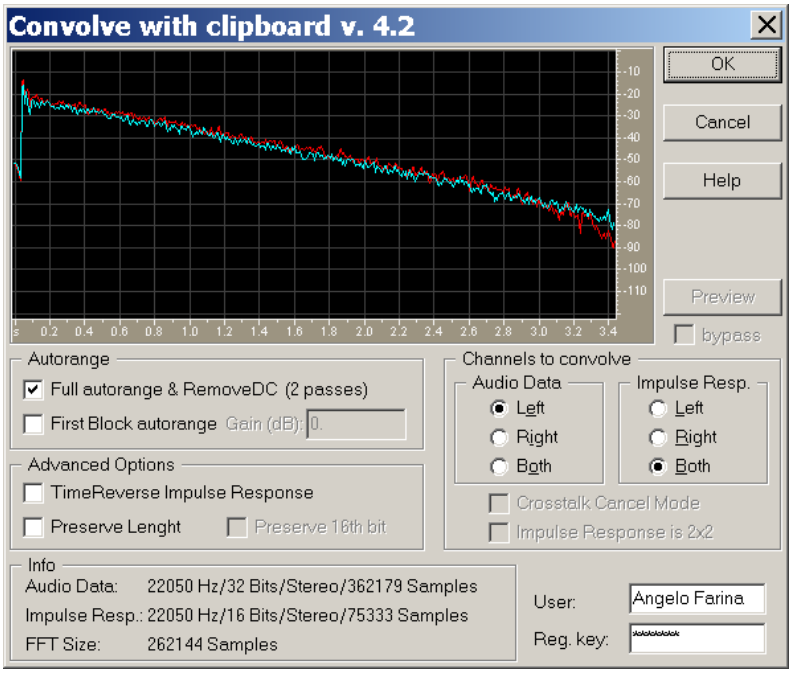
This is a mono file: if we want it to be convolved with the binaural stereo IR, producing a stereo result (auralisation), we need first to convert the waveform to stereo, making use of the standard Audition "Convert Sample Type" function.
It is also advisable to extend the bit depth from 16 to 32 bits, because the following convolution will expand significantly the dynamic range:



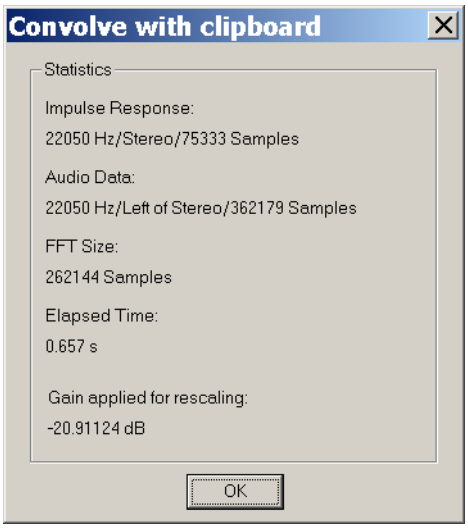
We have mixed the sound data to the left channel only, obtaining this result:



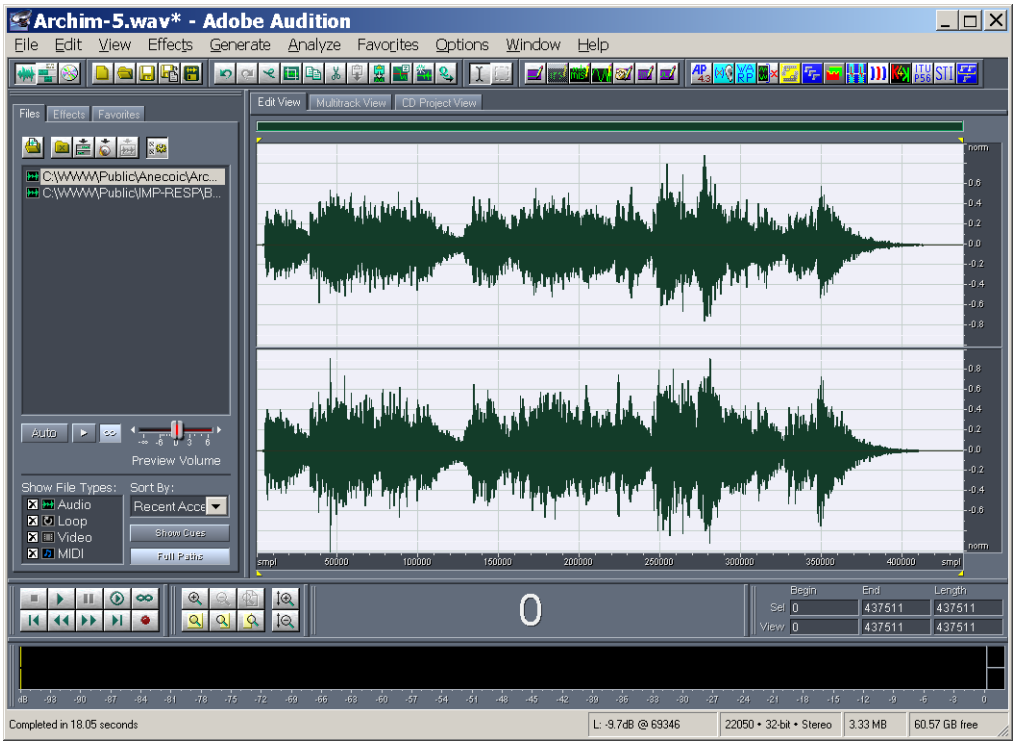
At this point, pressing the "Convolve with Clipboard" button, the following dialog Box appears:



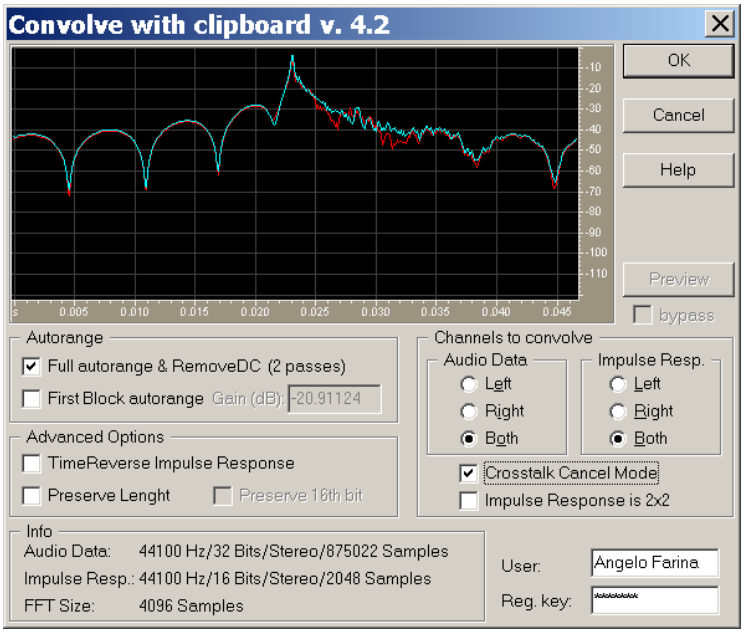
In this case, as it is shown, we want to convolve the Left channel of the current audio data with both the channels of the IR, and we want to perform a full autorange (selecting full autorange makes the “preview” function unavailable).
When OK is pressed, the standard Progress Meter of Audition appears, until the convolution process is finished. At this point the final dialog box is displayed, reporting statistics about the convolution process:



Pressing OK, the new convolved waveform is displayed.



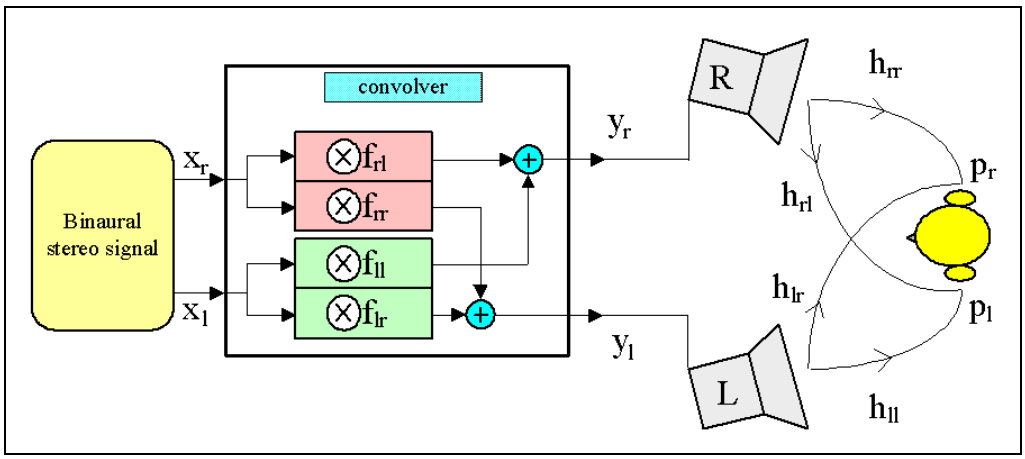
This stereo waveform, obtained by convolution of a mono signal with a binaural IR, is now suitable for headphone listening. If such a binaural signal needs to be reproduced onto a loudspeaker pair, it is necessary to apply a further convolution with a set of properly-designed cross-talk canceling filters. This can be done again with Convolve with Clipboard module.
There are several other possibilities embedded in the Convolve with Clipboard module. If both the inputs of a stereo waveform are passed to the module, the "cross-talk cancel mode" flag becomes available:



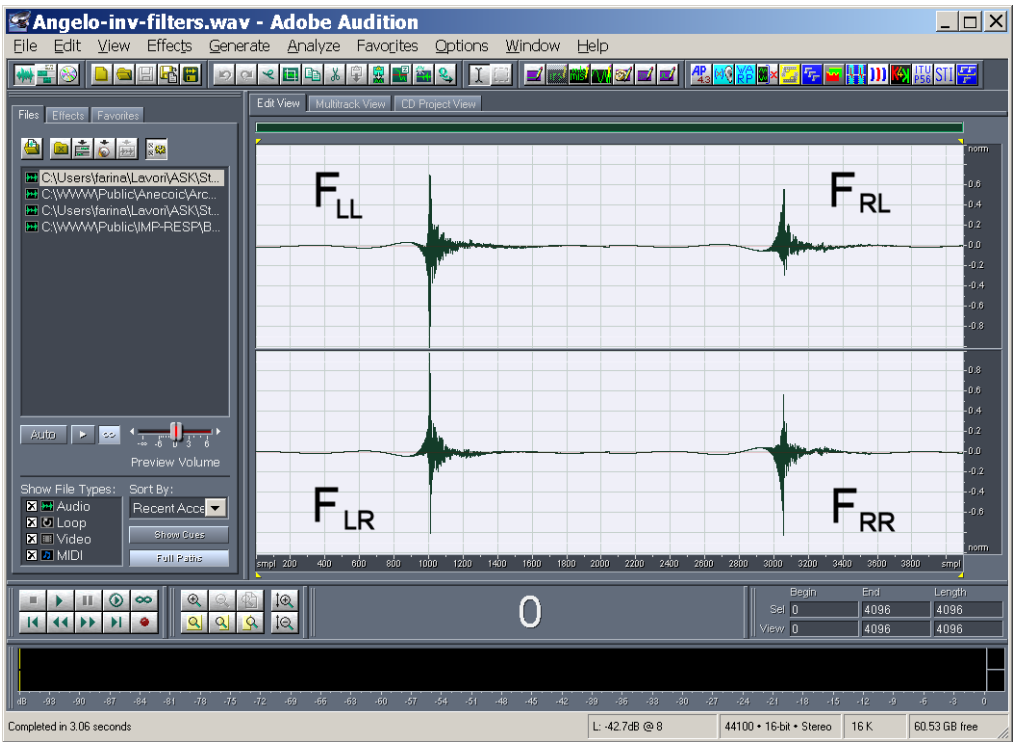
When the module is operated in this way, the Left Channel of the IR is treated as the ipselateral filter $\mathbf{F_I}$, and the right channel is treated as the cross-talk filter $\mathbf{F_C}$. The result of the convolution is:

$$\mathbf{Y_L = X_L \otimes F_I + X_R \otimes F_C}$$
$$\mathbf{Y_R = X_L \otimes F_C + X_R \otimes F_I}$$

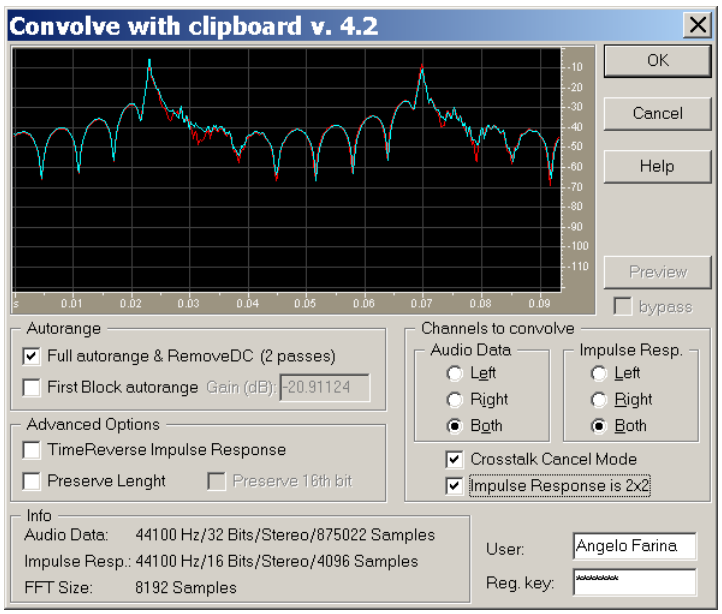
This mode of operation is useful for performing the reproduction of a binaural recording **X** with a set of cross-talk canceling filters **F** designed for a perfectly symmetric loudspeaker pair – these inverse filters are computed by the Invert Kirkeby module.
When the Crosstalk Cancel Mode flag is selected, it becomes possible also to specify that the Impulse Response employed is in the 2x2 format. This is a special way of storing a matrix of 2x2 cross-talk canceling filters, which is useful for managing the set of 4 independent inverse filters which are required for optimal cross-talk cancellation in the case of a not-symmetric loudspeaker system.
This general case is described in the following picture:



In the general case, the system is not perfectly symmetric, and consequently nothing ensures that the ipselateral filters F_{LL} and F_{RR} are equal, or that the crosstalk filters F_{LR} and F_{RL} are equal either.
In practice, after the system's transfer functions H have been measured (employing the MLS or the Sweep method), the Invert Kirkeby module is employed, which computes the corresponding cross-talk canceling filters F, which are shown in the following picture:



These inverse filters F need to be loaded into the Windows clipboard before the Convolve with Clipboard module is invoked. Then a binaural recording (or auralization) signal X is opened, and is convolved with the clipboard, specifying both the Crosstalk Cancel mode Flag and the Impulse Response is 2x2 flag.

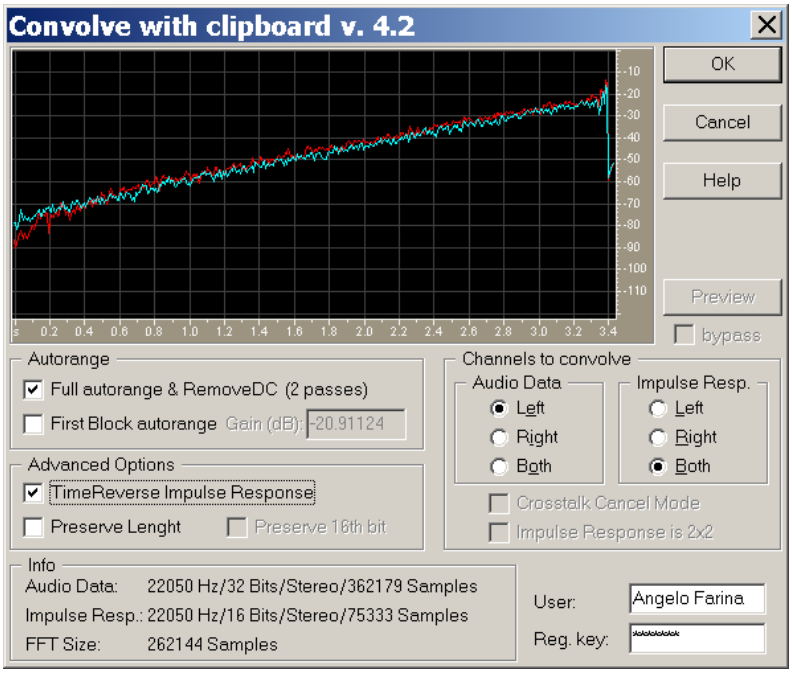


This way, the following computations are performed:

$$Y_L = X_L \otimes F_{LL} + X_R \otimes F_{RL}$$
$$Y_R = X_L \otimes F_{LR} + X_R \otimes F_{RR}$$

The result is a new stereo waveform, which is now suitable for reproduction over the loudspeaker pair, with perfect cross-talk cancellation. In practice, to the left ear only the original left channel X_L is arriving, and the cross-talk signal is being cancelled.
This method works better if the position of the loudspeakers is almost frontal (Stereo Dipole configuration), typically at +/- 10°, instead of the “standard” +/- 30° employed with “normal” stereo signals.

ADVANCED OPTIONS
There are three possible advanced options:
TimeReverse Impulse Response: the meaning should be clear, and the graphical display actually shows that this time-reversal is being applied, as shown here:



Preserve Length: normally, convolving a signal having a length of L samples with an impulse response having a length of N samples causes the result to be L+N-1 points long. Specifying this Advanced option, instead, the result is forced to be the same length L as the input signal. For minimizing the risk that the final decay is cutted, the software applies a suitable time-shift to the convolved waveform to the left, so that the attach point remains substantially the same (latency correction for the time-of-flight). However, if convolving a very long impulse response, the final decay will always be cutted…

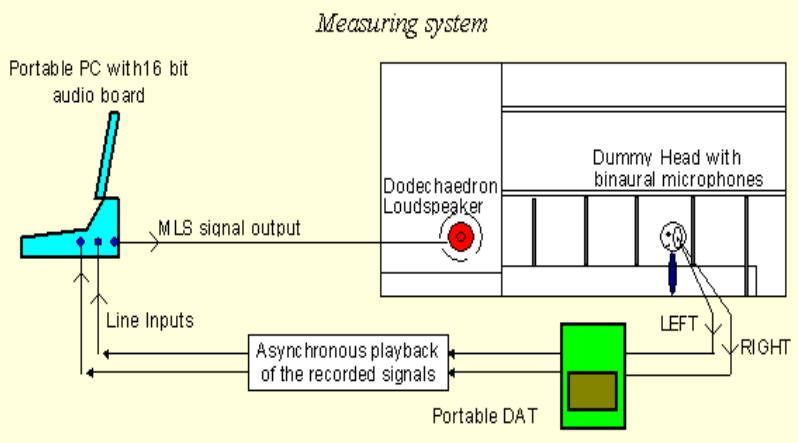
Preserve 16th bit: this option is only available when processing 16-bits waveforms. Selecting this option also forces the Preserve Length option to be selected.
Preserving the LSB of a 16-bit waveform can be useful if this bit was not actually containing audio information, but is instead employed for storing some other information: an example is the usage of 16th bit in old, low quality binaural systems, such as Head Acoustics and Artemis software – these obsolete systems did use 16-bits converters running at 48 kHz, but they did throw away the 16th bit (reducing further the already bad dynamic range), and storing RPM information and other scaling data in the 16th bit.
So, if the 16-bit file must be processed with Audition, and then reprocessed again with one of these obsolete systems, it is advisable to leave untouched the contents of the 16th bit.
This makes it possible to use Audition and Aurora for applying very effective digital filters to the recorded binaural waveforms, much better than the analogue filters available for headphone equalization in these systems.

$ K Generate MLS Signal

$ K Generate Multiple MLS Signal

$ K Generate IRS Signal

These three modules perform very similar tasks: they create the test signals, which are required for exciting the system when doing impulse response measurements with the following scheme:



MLS stands for Maximum Lenght Sequence, IRS stands for Inverse Repeated Sequence. Both these excitation signals are defined as a pseudo-random sequence of binary digits (0 and 1), built with a complex number-theory technique.
The important properties of these sequences are the following:

- Periodic: the signal can be repeated continuously, through the "Loop Mode" of playback in Audition. Averaging the system's response over many repeated sequences, the Signal-To-Noise ratio can be largely improved

---

GENMLS

$ Generate MLS Signal

K Generate MLS Signal

GENMULTI

$ Generate Multiple MLS Signal

K Generate Multiple MLS Signal

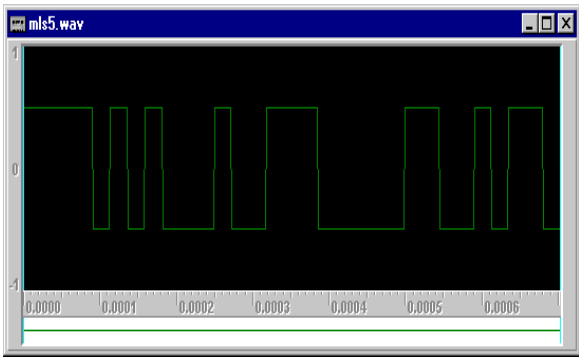
GENIRS

$ Generate IRS Signal

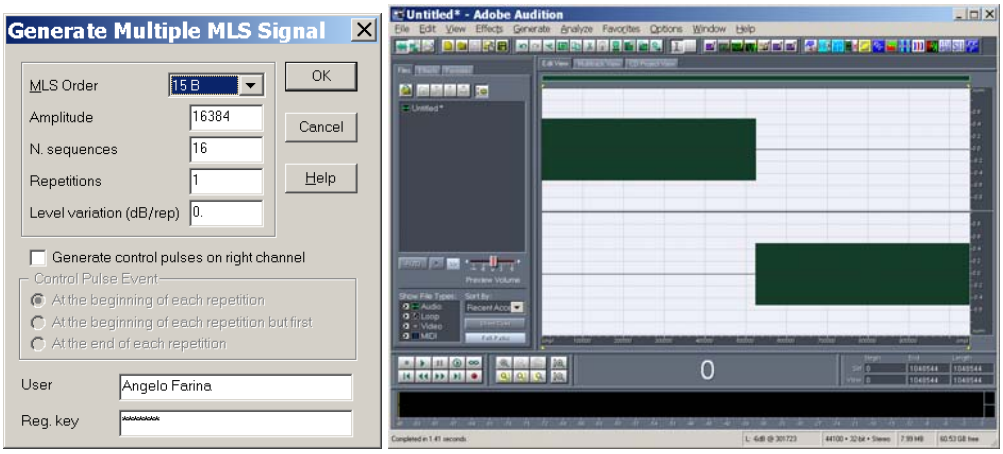
K Generate IRS Signal

- Binary: as the signal switches between only two extreme values, the crest factor is 0 dB, and this gives maximum power within a constrained amplitude.
- Flat spectrum: the frequency content of these signal is almost perfectly flat, resembling an ideal "white noise".
- Autocorrelation is a Dirac's Delta function, and this enables the deconvolution of the system's impulse response by correlating the measured response with the original sequence (correlating means convolving with the time-reversal of the test signal).
- The MLS sequence has the great advantage of making it possible the correlation process through a very fast algorithm, the Fast Hadamard Transform, which yields the result through a limited number of sums and subtractions, directly in the time domain. The disadvantage is that, if the system exhibits some kind of non-linear distortion, this can cause the appearance of some spurious "phantom echoes" at some well-defined tap position.
- The IRS sequence is made of two MLS sequences, the second being an inverted copy of the first one. This means that a 2*N sequence is required for measuring an N-point impulse response, doubling the measurement time. Furthermore, the fast Hadamard transform cannot be used for deconvolving the result, and the usual processing in the frequency domain (by use of very large FFTs and IFFTs) is required during the correlation. Although with this sequence the measurement is slower, it exhibits a larger immunity to distortion, and particularly to odd-order distortion.

Aurora makes it possible to generate MLS and IRS sequences up to order 21: this means that the maximum length of the measured IRs will be 2^21-1, that is 2,097,151 samples, corresponding to more than 47 s at a sampling frequency of 44.1 kHz !!!
For the most commonly used orders (14, 15, 16 and 17), more than one sequence is available (labelled as 16A, 16B and 16C, for example). Each of them is produced according to a different permutation of the numerical basis, and thus the results are uncorrelated. In case of distortion, each of these sequences produces spurious echoes at different positions, so that they can be easily identified and removed.
The following picture shows an MLS sequence of order 5 (period of 31 samples):

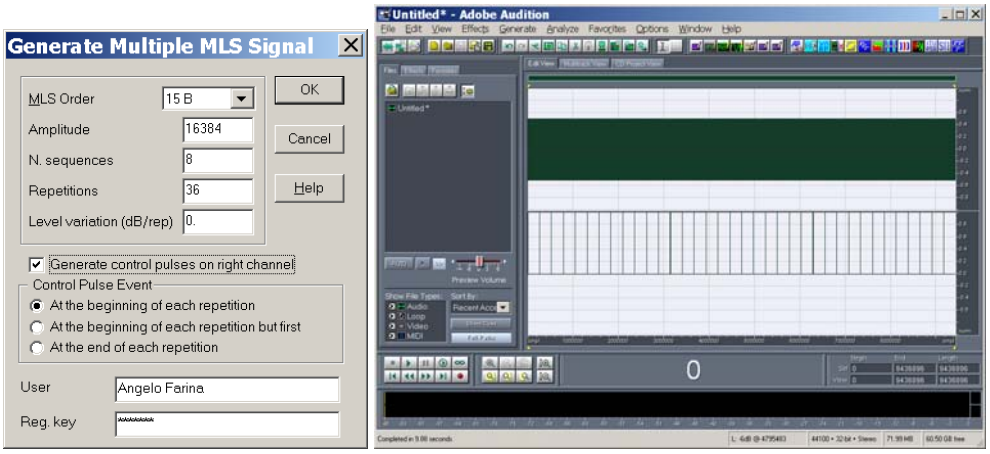


The most versatile of these three generation modules is the Generate Multi MLS. This module was optimized for performing measurements of multi-channel systems (typically 2x2 systems, in which two loudspeaker and 2 microphones are involved, such as the case of a binaural microphone placed in front of a loudspeaker pair).
When generating a stereo signal, this module creates first a series of MLS sequence on the left channel, followed by the same signal repeated later on the right channel:

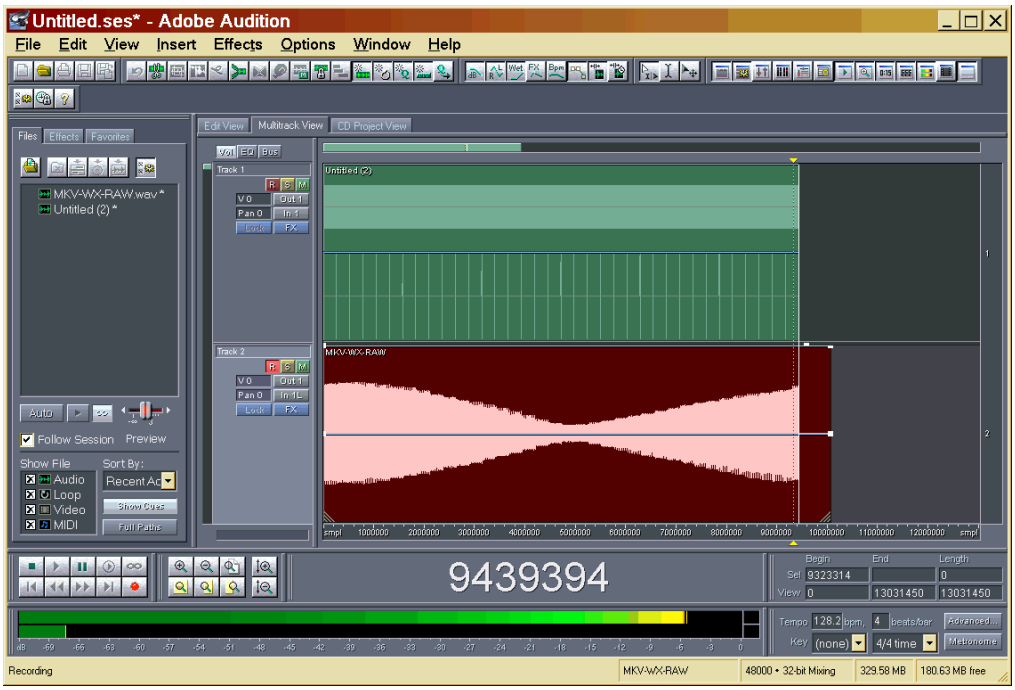


This way, the measurement can be performed playing the stereo signal, and recording the system's response simultaneously; first the left loudspeaker will be excited, followed by the right loudspeaker.
After deconvolution performed with the Deconvolve Multi MLS Signal module, the impulse responses will appear already in the 2x2 format (a stereo signal containing first the binaural IR of the left loudspeaker, followed by the binaural IR of the right loudspeaker).

The second advanced feature of this module is the capability to employ the right channel for controlling the rotation of a rotating table. This is very useful for measuring the polar response of acoustic transducers (loudspeakers and microphones).
Suppose that we want to measure the polar response of a loudspeaker, making use of a low-cost rotating table (for example, the Outline ET-2 turntable). We connect the left channel output of the soundcard with the power amplifier driving the loudspeaker, and the right channel output is connected to the digital input of the control unit of the turntable. Usually the turntable is designed for advancing of an angular step when a TTL pulse is applied to its digital input. Most soundcards have a voltage swing suitable for triggering such a digital input, so it is possible to wire directly the Right channel output to the digital input. Some notebooks, however, provide a signal which is too weak for triggering the rotating table, so a buffer is required between the output of the soundcard and the input of the rotating table.
We suppose here to want to measure the polar pattern with an angular resolution of 10°, and to have already set the angular step of the rotating table to this angle.
Now we generate a very long test signal, which is made by 36 repetitions of a cluster of 8 MLS sequences; we also specify to generate, on the right channel, the bursts of noise which will trigger the rotating table, and that we want one of these bursts at the beginning of each cluster, except for the first one:



Then the measurement is performed employing the multitrack mode of Audition:



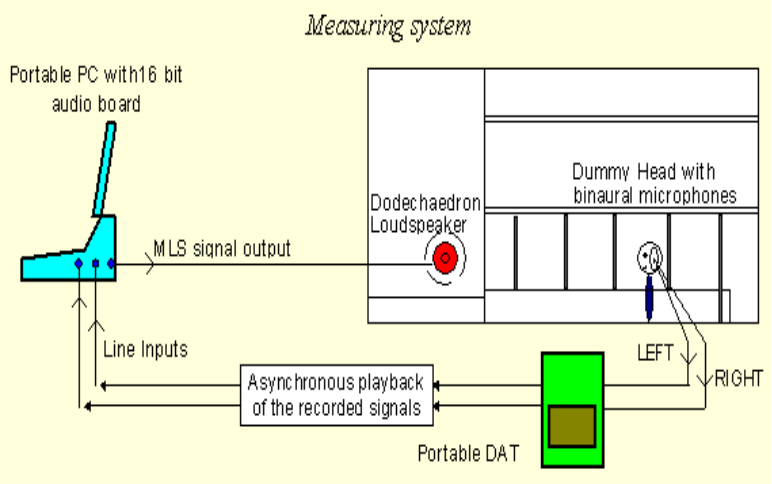
The recorded signal will then be deconvolved employing the Deconvolve Multi MLS Signal module, and the result will be a sequence of 36 impulse responses, one for each angular position.

$ K Deconvolve MLS Signal

$ K Deconvolve Multiple MLS Signal

$ K Deconvolve IRS Signal

These three modules perform very similar tasks: they recover the impulse response during measurements made following this scheme:



The computer is used for generating the MLS (or IRS) sequence, and for feeding the omnidirectional loudspeaker (or the sound reinforcement system, if this must be included in the measurement). The recording of the response is not made simultanously to the playback of the signal, for three reasons:
1. A full-duplex audio board is required for simultaneous playback and recording, and most notebooks are not equipped with such boards.

DECOMLS

$ Deconvolve MLS Signal

K Deconvolve MLS Signal

DECOMULTI

$ Deconvolve Multiple MLS Signal

K Deconvolve Multiple MLS Signal

DECOIRS

$ Deconvolve IRS Signal

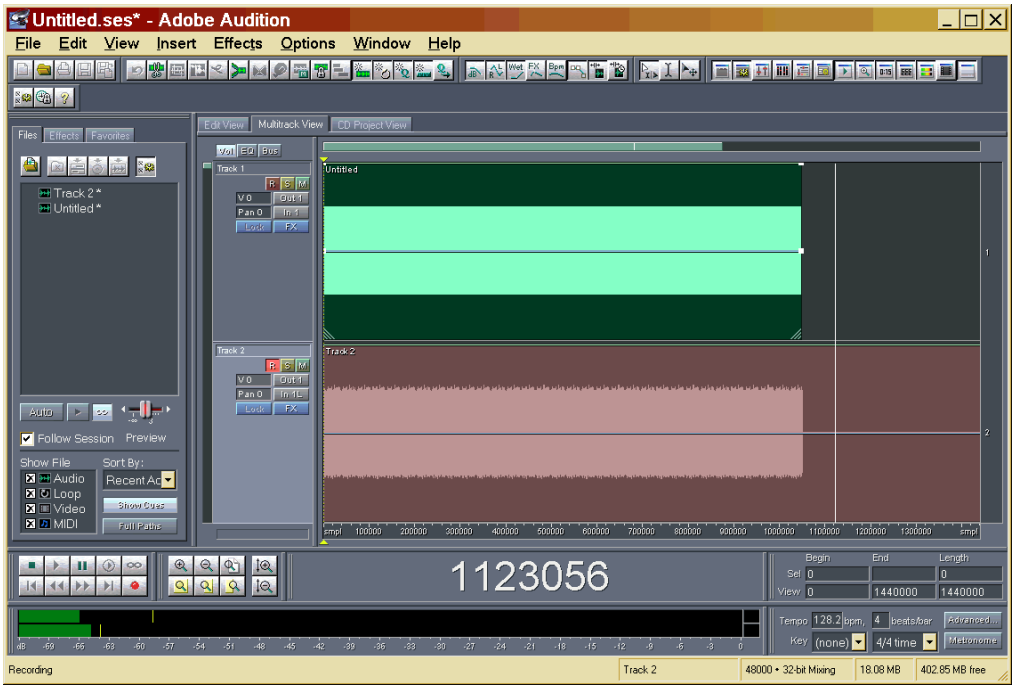
K Deconvolve IRS Signal

2. Most audio boards don't have a stereo MIC input (preamplifiers should be required for entering the LINE IN stereo input).
3. You need long cables or a stereo wireless link for connecting the microphones with the computer

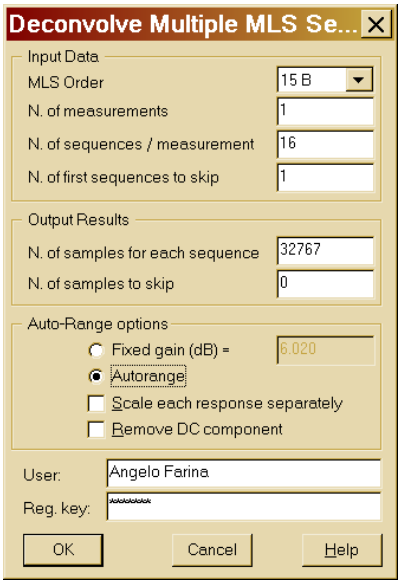
A small, portable DAT machine solves all these problems, enabling very fast measurement inside the room. Then, at home, You can playback the sampled signals from the DAT cassette to the audio board, recovering the binaural impulse response in each point of the room where You have recorded the signals.

Obviously, if You make laboratory measurement with a full-size computer, You probably have a full-duplex audio board, You don't have problems using high quality preamplifiers and cables, and You prefer to have real-time measurements for looking quickly at the results.

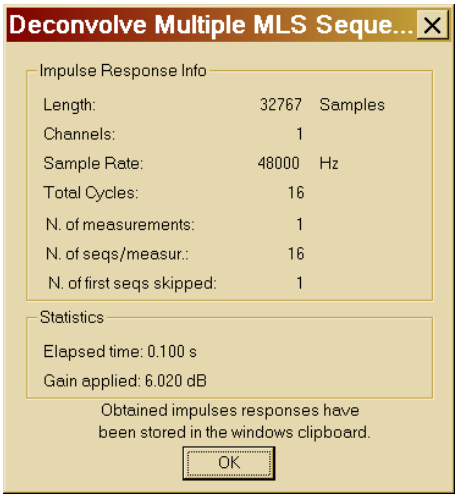
The following picture shows how the signal appears in Audition (Multitrack View) after recording a period corresponding to 32 repetitions of a MLS 15B signal:



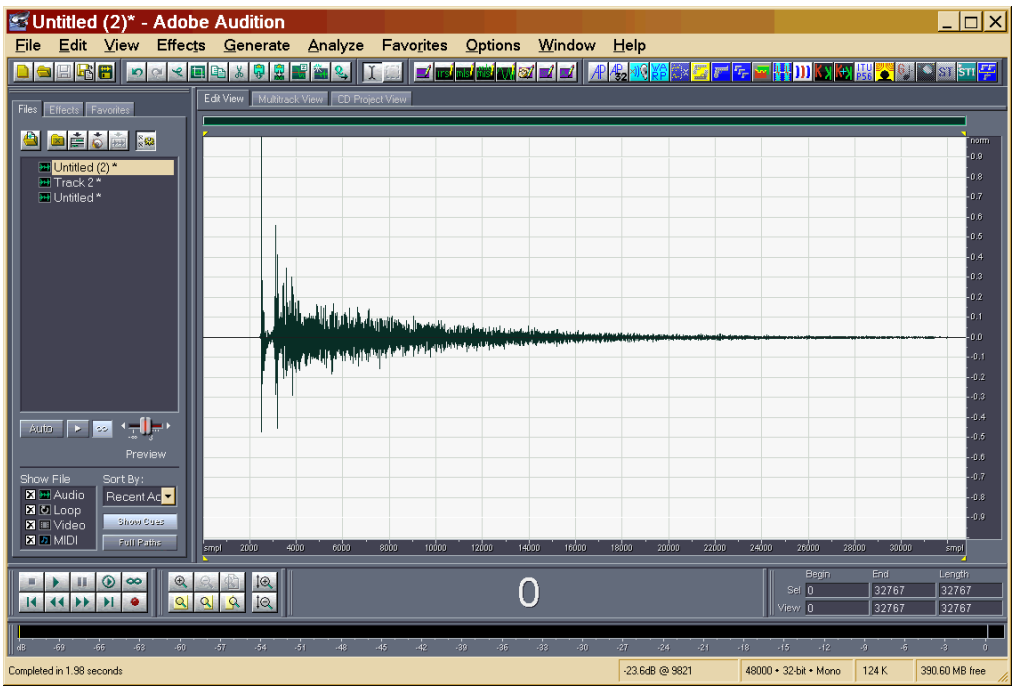
Now Track2 is opened in the Edit View mode (just double-click over it), and the Deconvolve Multiple MLS module is invoked (or the DecoMLS or the DecoRS, if appropriate). The following mask does appear:



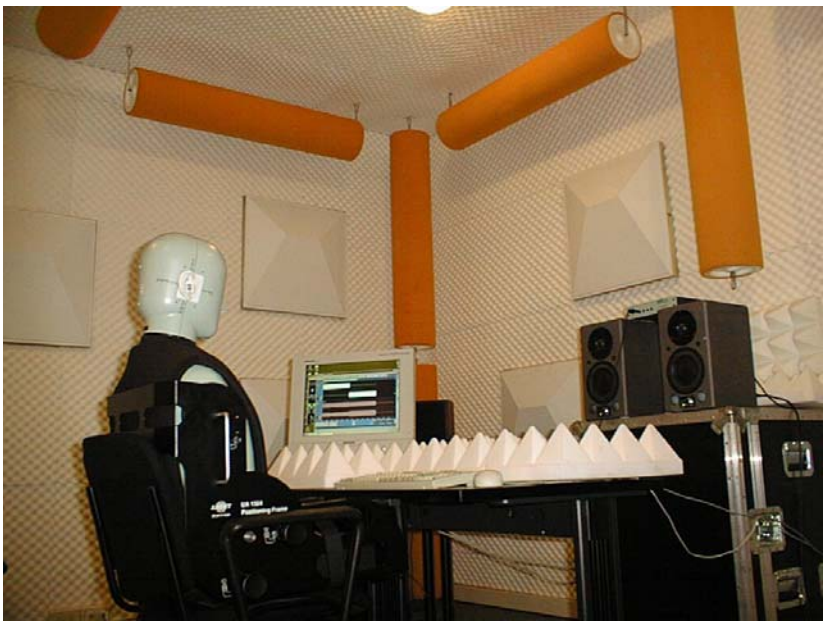
It is mandatory, for obtaining a proper deconvolution, to employ the proper module (MLS, Multi MLS or IRS), and to specify exactly the same parameters as those employed during the generation of the test signal.
In this case, it was an MLS 15B signal, it is a single (mono) measurement, the MLS sequence was repeated 16 times, and the first of these 16 repetitions is being discarded, because the system did not reach yet the steady state during the first one.
The whole available length of the IR is being retained (32767 samples), and no samples are skipped at the beginning.
After applying the deconvolution process, as shown above in the dialog box, You get this result:



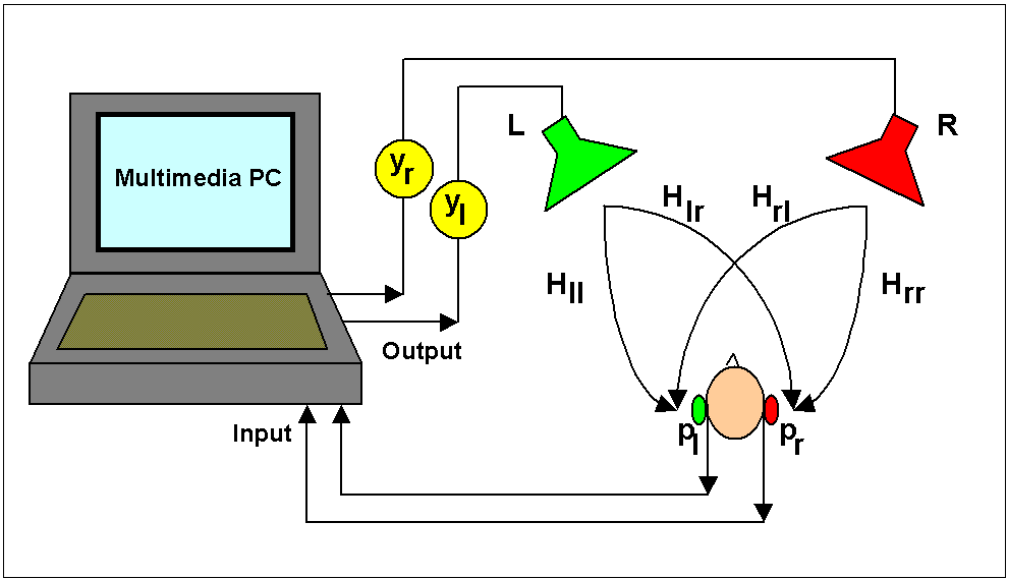
At this point, the deconvolved impulse response has been loaded by the module onto the Windows clipboard. For recovering it, without destroying the original recorded waveform, You can operate an Edit – Paste to New command, which will create a new waveform window, containing the Clipboard data: this is the waveform obtained in this way:



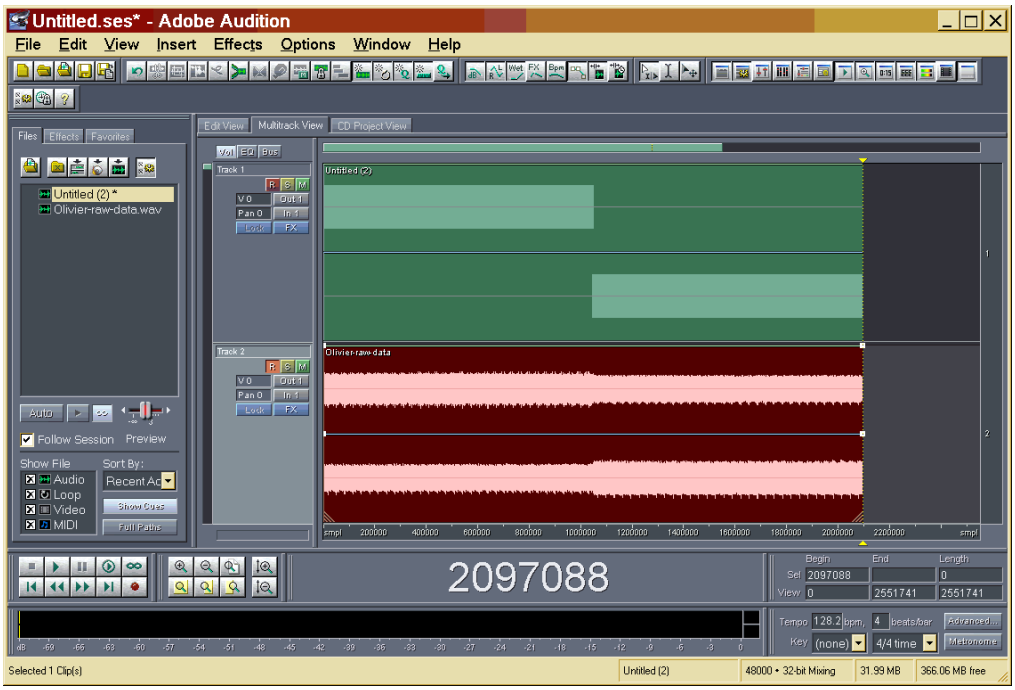
Now let's have a look at a typical 2x2 measurement: a binaural dummy head microphone placed in front of a pair of loudspeakers:



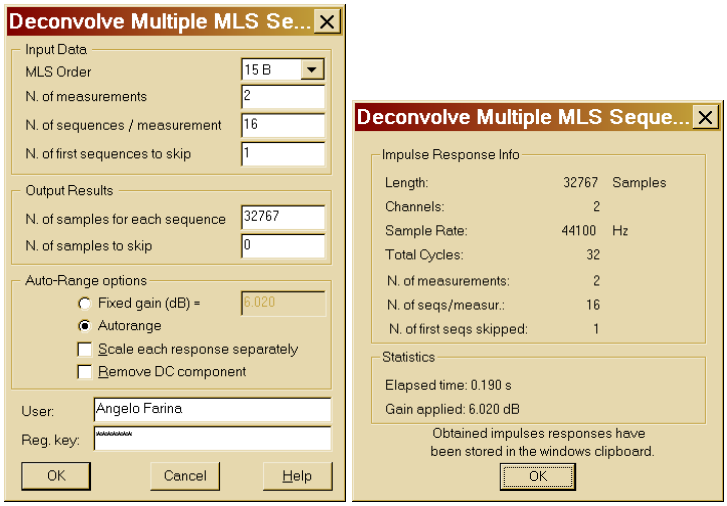
In this case, 4 impulse responses need to be measured, following this scheme:



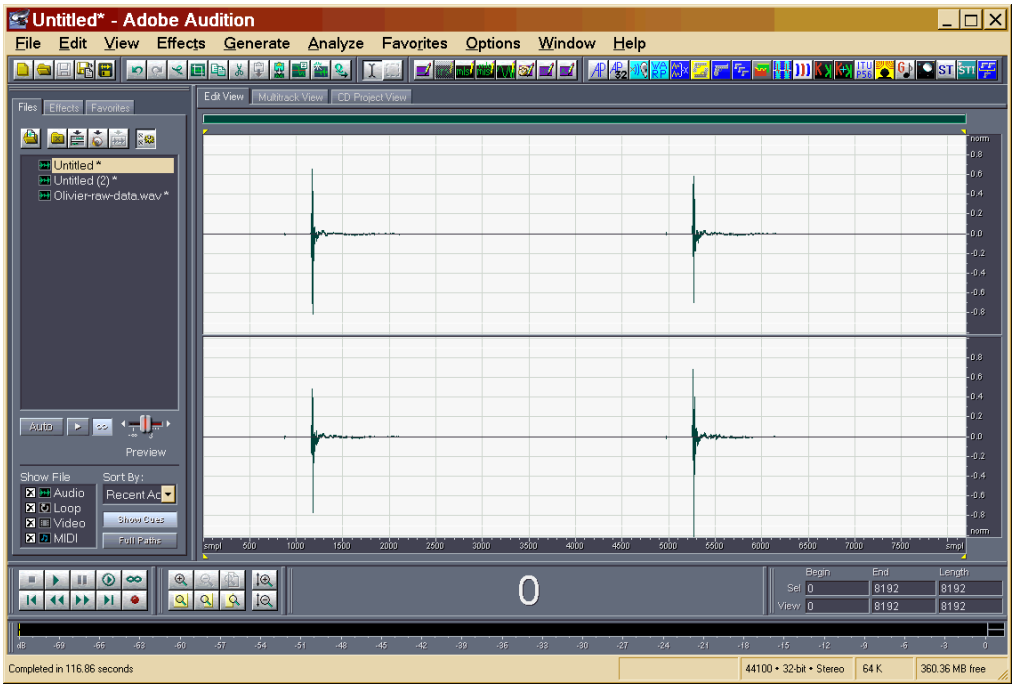
In this case, first a stereo multiple MLS signal is generated, employing the Generate Multi MLS Signal module.
After the synchronous measurement has been performed, the situation is as follows:



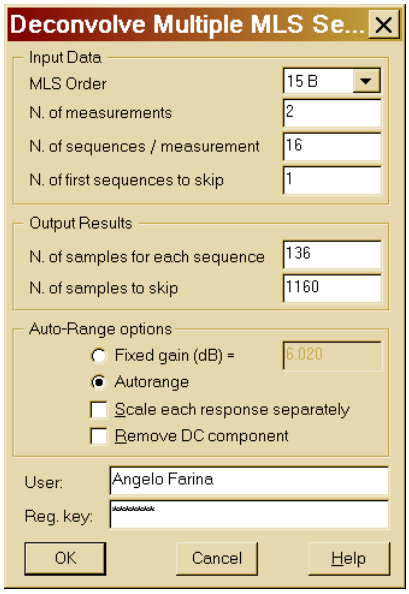
At this point, the Deconvolve Multiple MLS Sequence module is invoked, specifying the proper generation parameters (32 repetitions of the MLS 15B sequence), and that these are 2 measurements (first the left loudspeaker, second the right loudspeaker):



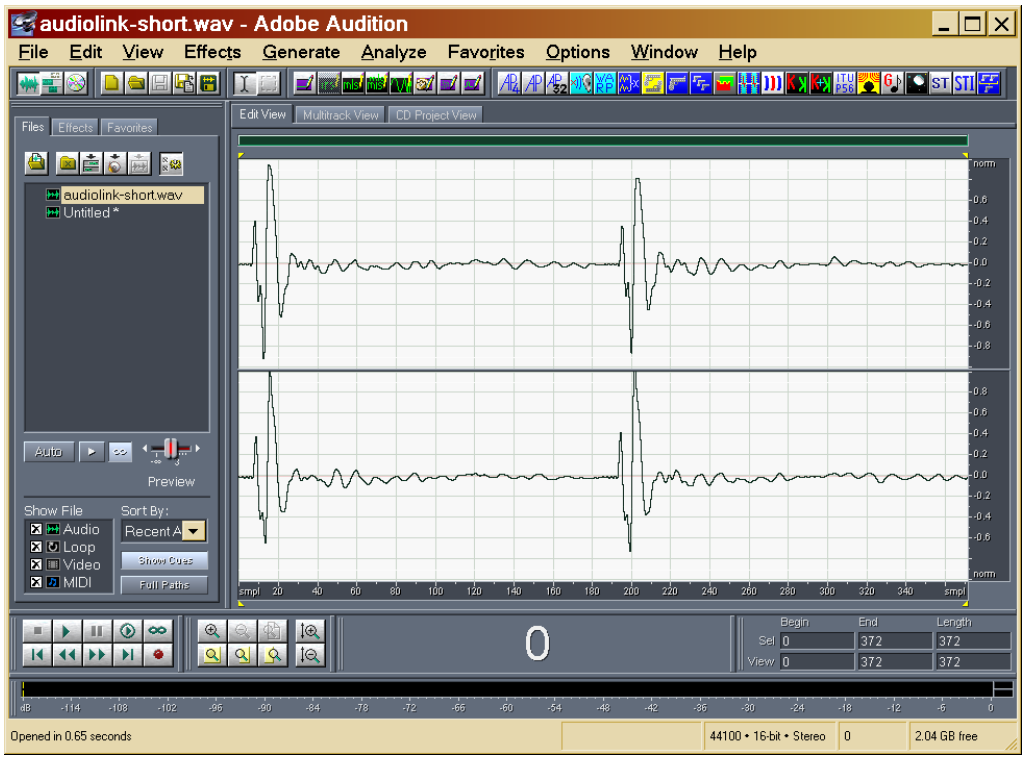
After pasting the results from the Windows clipboard to a new waveform, the following is obtained:



These results are already in the standard 2x2 format, which is managed properly both by the Convolve with Clipboard module and by the Invert Kirkeby module.
However, it could be advisable to retain only a very short time window which contains only the direct sound. This can be obtained repeating the deconvolution, after having inspected carefully the time lags of the above results. It can be seen that the direct sound is long only 136 samples before the first reflection on the floor occurs, and that it is advisable to skip the first 1160 samples, which are the long piece of silence before the arrival of the direct sound. So the deconvolution is performed again on the original raw waveform with the following parameters:



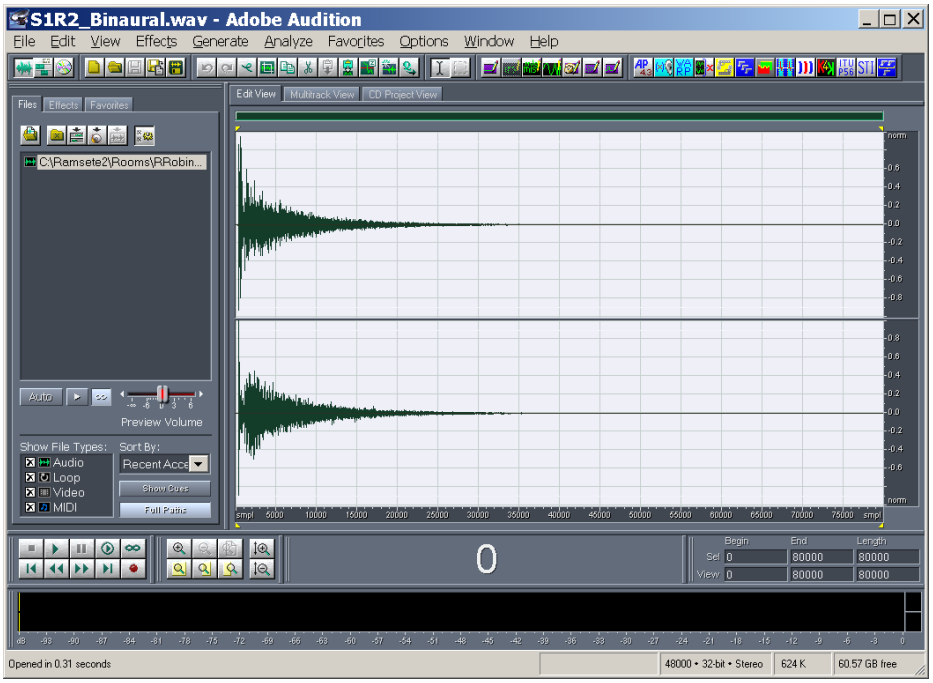
And the following result is obtained:



This “short” version of the 2x2 IR is much better for creating cross-talk canceling filters employing the Invert Kirkeby module.

$ K Acoustical Parameters

After measuring an Impulse Response, this module makes it possible to calculate the Acoustical Parameters defined in the ISO standard 3382, in 10 octave bands (from 31.5 Hz to 16 kHz). These include various kinds of Reverberation Time (EDT, T20, T30, Tuser), Clarity C50 and C80, Definition, Center Time.
Furthermore, for binaural measurements various quantities related to Ando's IACC theory are computed – instead, for pressure-velocity measurements, it is possible to display LE, LF and LFC. Finaly, it is possible to compute a pressure-velocity response from two pressure responses measured with a pair of matched capsules located at a very short distance one from the other (a Sound Intensity probe, for example).
The calculations are made accordingly to the ISO 3382 standard: the impulse response is octave-band filtered by means of IIR IEC-compliant filters, then it is squared (for obtaining the acoustic energy) and processed. For the reverberation time estimation, the squared impulse response is backward-integrated, with an optional noise-removal algorithm. Then a linear regression is performed over the proper dB range.
For example, let's start from the following binaural (stereo) impulse response:

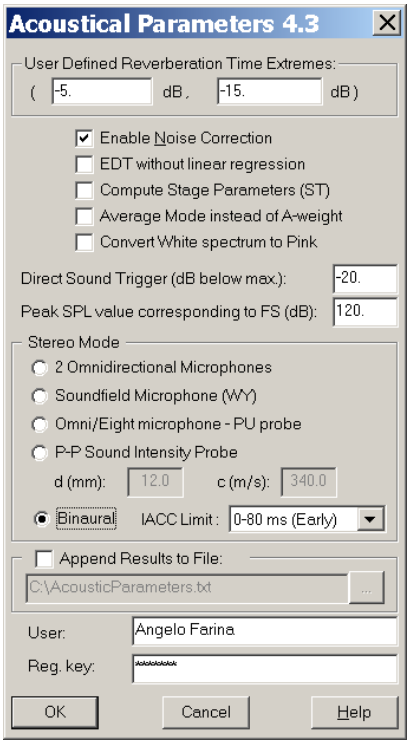


Now we invoke the Acoustical parameters plugin. The following set-up mask appears:

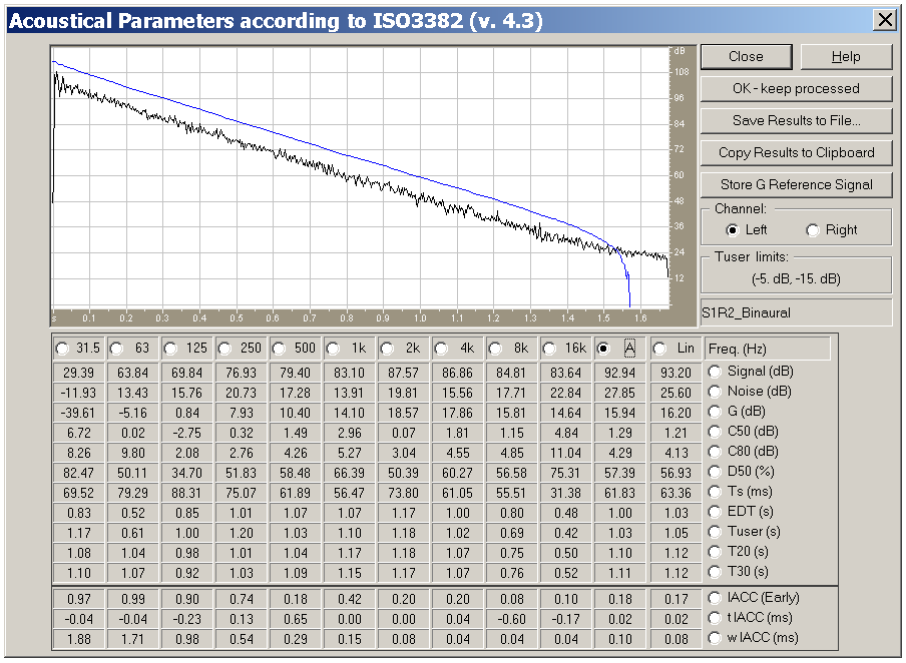
ACUPAR

$ Acoustical Parameters

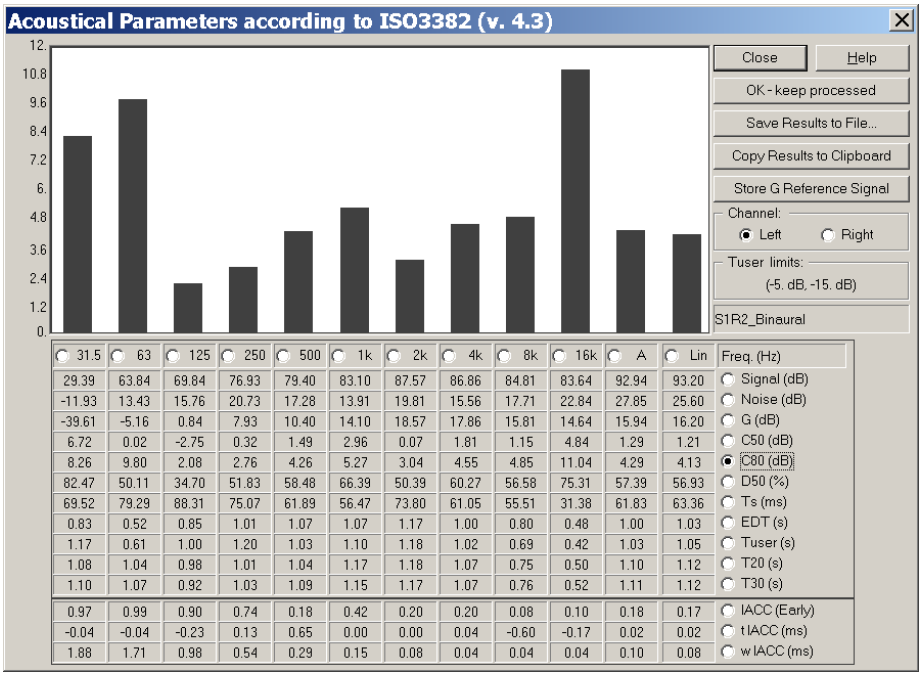
K Acoustical Parameters



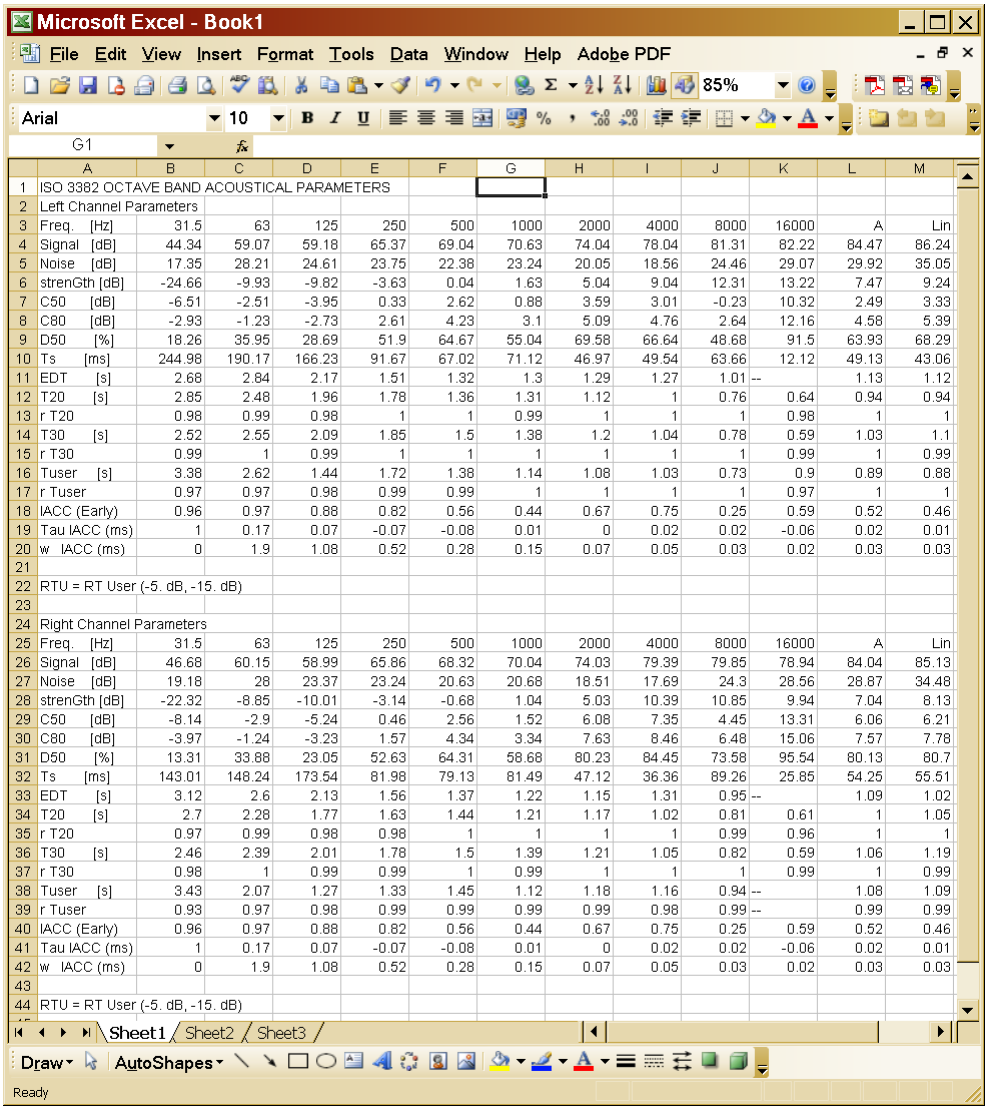
As this was a binaural impulse response, the “Binaural Dummy Head” microphone was selected.
Furthermore it was chosen to compute an “early” IACC, in the range 0-80 ms (it is also possible here to specify a “late” IACC (80ms->infinite), or All (0->infinite)
Pressing the OK button these acoustical parameters are obtained:



The user can choose to display a plot of the impulse response and of the Schroeder-integrated decay (in blue) at any of the frequency bands, just selecting it. Similarly, the user can switch to the right or left ear display at will.
Furthermore, also an octave-band spectrum of each parameter can be shown:



These data can easily be exported to other programs via the Windows Clipboard. After copying them to the clipboard, and reimporting them in Excel, they appear in this way:



Microsoft Excel - Book1

File Edit View Insert Format Tools Data Window Help Adobe PDF

Arial 10 G1

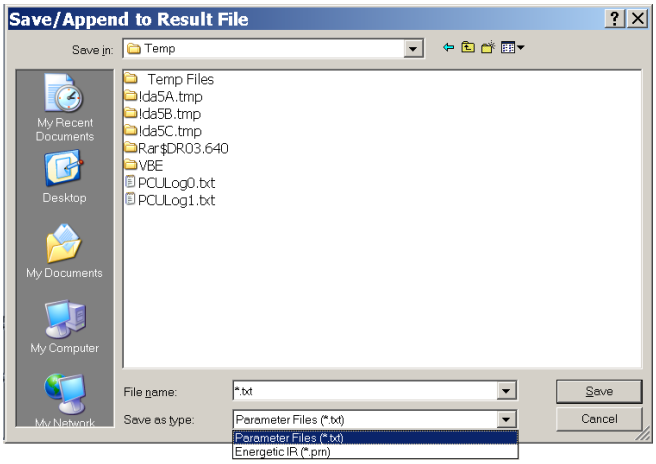
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ISO 3382 OCTAVE BAND ACOUSTICAL PARAMETERS												
2	Left Channel Parameters												
3	Freq. [Hz]	31.5	63	125	250	500	1000	2000	4000	8000	16000	A	Lin
4	Signal [dB]	44.34	59.07	59.18	65.37	69.04	70.63	74.04	78.04	81.31	82.22	84.47	86.24
5	Noise [dB]	17.35	28.21	24.61	23.75	22.38	23.24	20.05	18.56	24.46	29.07	29.92	35.05
6	strenGth [dB]	-24.66	-9.93	-9.82	-3.63	0.04	1.63	5.04	9.04	12.31	13.22	7.47	9.24
7	C50 [dB]	-6.51	-2.51	-3.95	0.33	2.62	0.88	3.59	3.01	-0.23	10.32	2.49	3.33
8	C80 [dB]	-2.93	-1.23	-2.73	2.61	4.23	3.1	5.09	4.76	2.64	12.16	4.58	5.39
9	D50 [%]	18.26	35.95	28.69	51.9	64.67	55.04	69.58	66.64	48.68	91.5	63.93	68.29
10	Ts [ms]	244.98	190.17	166.23	91.67	67.02	71.12	46.97	49.54	63.66	12.12	49.13	43.06
11	EDT [s]	2.68	2.84	2.17	1.51	1.32	1.3	1.29	1.27	1.01	--	1.13	1.12
12	T20 [s]	2.85	2.48	1.96	1.78	1.36	1.31	1.12	1	0.76	0.64	0.94	0.94
13	r T20	0.98	0.99	0.98	1	1	0.99	1	1	1	0.98	1	1
14	T30 [s]	2.52	2.55	2.09	1.85	1.5	1.38	1.2	1.04	0.78	0.59	1.03	1.1
15	r T30	0.99	1	0.99	1	1	1	1	1	1	0.99	1	0.99
16	Tuser [s]	3.38	2.62	1.44	1.72	1.38	1.14	1.08	1.03	0.73	0.9	0.89	0.88
17	r Tuser	0.97	0.97	0.98	0.99	0.99	1	1	1	1	0.97	1	1
18	IACC (Early)	0.96	0.97	0.88	0.82	0.56	0.44	0.67	0.75	0.25	0.59	0.52	0.46
19	Tau IACC (ms)	1	0.17	0.07	-0.07	-0.08	0.01	0	0.02	0.02	-0.06	0.02	0.01
20	w IACC (ms)	0	1.9	1.08	0.52	0.28	0.15	0.07	0.05	0.03	0.02	0.03	0.03
21													
22	RTU = RT User (-5. dB, -15. dB)												
23													
24	Right Channel Parameters												
25	Freq. [Hz]	31.5	63	125	250	500	1000	2000	4000	8000	16000	A	Lin
26	Signal [dB]	46.68	60.15	58.99	65.86	68.32	70.04	74.03	79.39	79.85	78.94	84.04	85.13
27	Noise [dB]	19.18	28	23.37	23.24	20.63	20.68	18.51	17.69	24.3	28.56	28.87	34.48
28	strenGth [dB]	-22.32	-8.85	-10.01	-3.14	-0.68	1.04	5.03	10.39	10.85	9.94	7.04	8.13
29	C50 [dB]	-8.14	-2.9	-5.24	0.46	2.56	1.52	6.08	7.35	4.45	13.31	6.06	6.21
30	C80 [dB]	-3.97	-1.24	-3.23	1.57	4.34	3.34	7.63	8.46	6.48	15.06	7.57	7.78
31	D50 [%]	13.31	33.88	23.05	52.63	64.31	58.68	80.23	84.45	73.58	95.54	80.13	80.7
32	Ts [ms]	143.01	148.24	173.54	81.98	79.13	81.49	47.12	36.36	89.26	25.85	54.25	55.51
33	EDT [s]	3.12	2.6	2.13	1.56	1.37	1.22	1.15	1.31	0.95	--	1.09	1.02
34	T20 [s]	2.7	2.28	1.77	1.63	1.44	1.21	1.17	1.02	0.81	0.61	1	1.05
35	r T20	0.97	0.99	0.98	0.98	1	1	1	1	0.99	0.96	1	1
36	T30 [s]	2.46	2.39	2.01	1.78	1.5	1.39	1.21	1.05	0.82	0.59	1.06	1.19
37	r T30	0.98	1	0.99	0.99	1	0.99	1	1	1	0.99	1	0.99
38	Tuser [s]	3.43	2.07	1.27	1.33	1.45	1.12	1.18	1.16	0.94	--	1.08	1.09
39	r Tuser	0.93	0.97	0.98	0.99	0.99	0.99	0.99	0.98	0.99	--	0.99	0.99
40	IACC (Early)	0.96	0.97	0.88	0.82	0.56	0.44	0.67	0.75	0.25	0.59	0.52	0.46
41	Tau IACC (ms)	1	0.17	0.07	-0.07	-0.08	0.01	0	0.02	0.02	-0.06	0.02	0.01
42	w IACC (ms)	0	1.9	1.08	0.52	0.28	0.15	0.07	0.05	0.03	0.02	0.03	0.03
43													
44	RTU = RT User (-5. dB, -15. dB)												

Sheet1 Sheet2 Sheet3

Draw AutoShapes

Ready

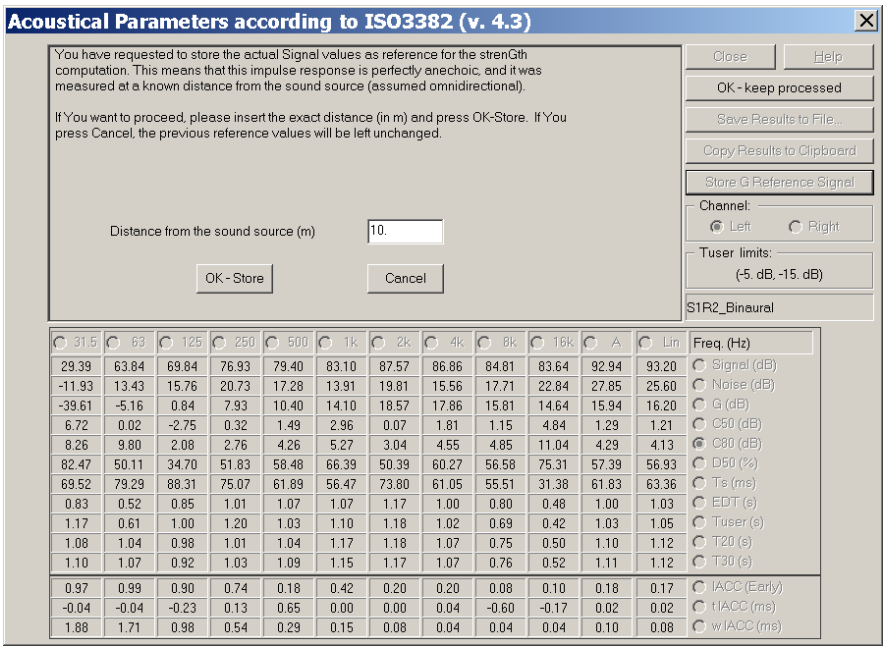
The user can also save both the parameters and the processed multispectrum in an ASCII text file. Pressing the button labeled Save Results to file… - the following window appears:



Chosing Parameter Files exactly the same table as stored to the clipboard is saved to a TXT file. Choosing Energetic IR, instead, a PRN file is created, containing a table of energetic values in octave bands, sampled with an 1ms time resolution.
Both files can be easily imported in Excel, Matlab or other programs for further processing.

Another advanced option is the “OK – keep processed” button. If this button is pressed whilst any of the “filtered” bands is shown (octave bands or A-weighted band), then the plugin is terminated, but the currently selected waveform is replaced by the filtered version of it.

A final remark is about the “Store G reference signal” button. Pressing this button, the graph disappears, and the following controls are displayed:



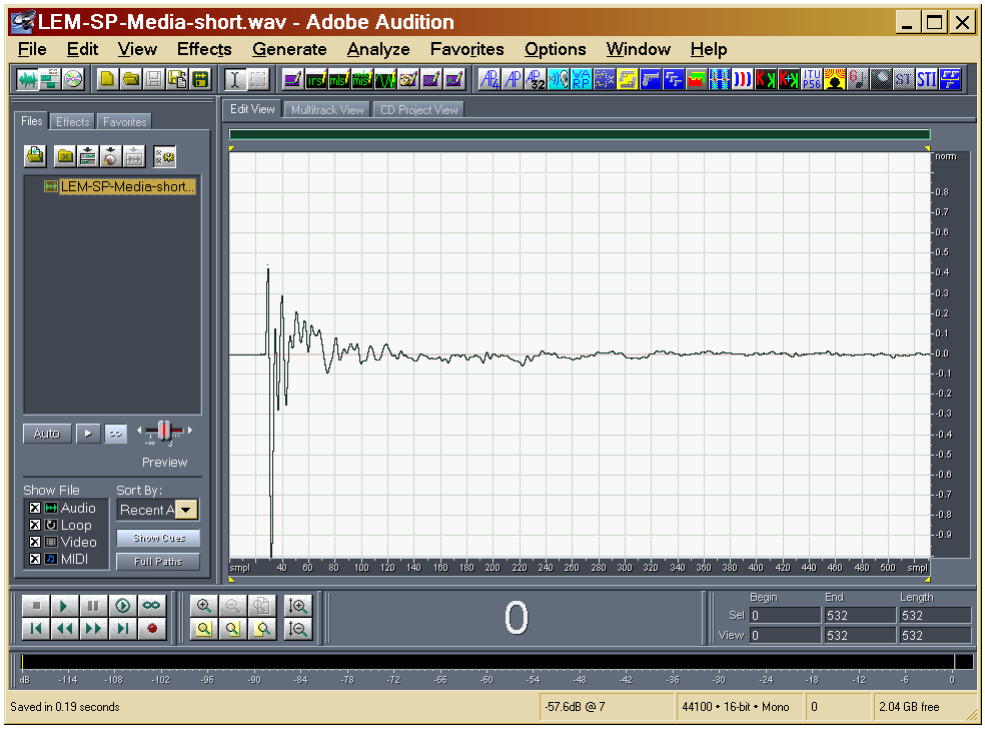
The user can enter the distance between the omnidirectional source and the microphone (the standard value is 10m, but in most practical cases it is advisable to employ a shorter distance, for example 2.5m). Then, pressing OK-Store, the current value of the Signal parameter as stored as the reference anechoic emission values of the sound source.
This means that it makes sense to store these values only if the processed impulse response was an anechoic impulse response measurement. In the case an anechoic room was not available, the anechoic segment of a “normal” IR can be used. But it is not recommended simply to select the anechoic portion of an IR, which is usually very short. It is instead recommended to keep a length of the IR of at least 1s, and to silence the signal just after the end of the direct sound, but still selecting a very long time segment before invoking the Acoustical parameters plugins. This way, also at low frequency the time-spreading caused by the octave filtering will not push the energy outside the time window, and the correct value of Signal level will be computed. You can check if the time window is long enough displaying the 31.5 Hz response – it should decay to zero well before the end of the window…
The user must however be warned that the measurement of G (strength) is quite delicate, and usually it is very difficult to get accurate estimate of G through impulse response measurement. The only reliable way to measure G is to employ a calibrated-power source, radiating steady pink noise. The calibration of the effectively radiated sound power level Lw can be obtained in a reverberant room according to ISO 3743, or in free field using a Sound Intensity apparatus and the ISO 9614/2 standard.
Then the SPL values inside the concert hall are measured by means of a portable real-time, octave-band analyzer, and finally the value of G, at each frequency, is obtained by:

$$G = SPL - Lw + 31\ dB$$

This is the recommended way of measuring G, and the only one which produces truly reproducible results. The values of G computed by the Acoustical Parameters plugin tend to the true ones, but the error is acceptable only if the sound source was truly point-like and perfectly omnidirectional – this is particularly problematic during the Reference anechoic measurement, where it is often observed that, rotating the “omnidirectional” dodecahedron by a few dB, the free-field Signal changes of several dB, particularly at frequencies above 1 kHz.
Finally, during the IR measurement process it is very easy to alter the absolute SPL calibration, applying uncontrolled gain or automatic rescaling – of course the measurement must be perfectly calibrated, if the value of G are required, and any change in the recording level or in the rescaling factor during the IR deconvolution must be absolutely avoided.
Again, the usage of a calibrated source and of a portable SLM are recommended, as this avoids any calibration problem.

[#] [$] [K] Flatten Spectrum

This module creates an inverse filter, starting from a generic Impulse Response, and inverting only the Minimum Phase component of it. This means that the Magnitude of the Transfer Function is inverted, and the inverse filter, applied to a signal containing the effect of the original impulse response, perfectly equalizes it in terms of spectral flatness.
It must be nothed, anyway, that the non-minimum phase component is left not-inverted: as this component carries the reverberation, the equalised signal has a proper spectral frequency response, but it still suffers of echoes or reverberation present in the original signal. Thus this module is ideal for creating an equalising filter taking into account only the direct wave coming from a loudspeaker with poor frequency response.
For example, suppose that we have measured the anechoic IR of a loudspeaker:



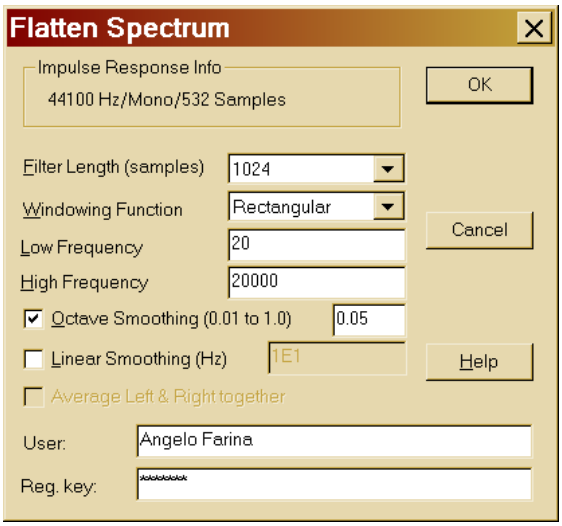
through Audition we can look at its Frequency Response:

[#] SPECTRUM

[$] Flatten Spectrum

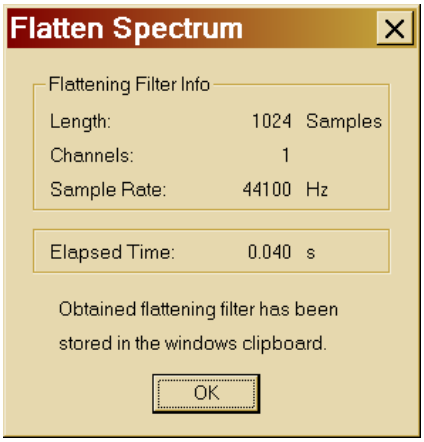
[K] Flatten Spectrum

Now we can press the Flatten Spectrum button, and the following Dialog Box appears:

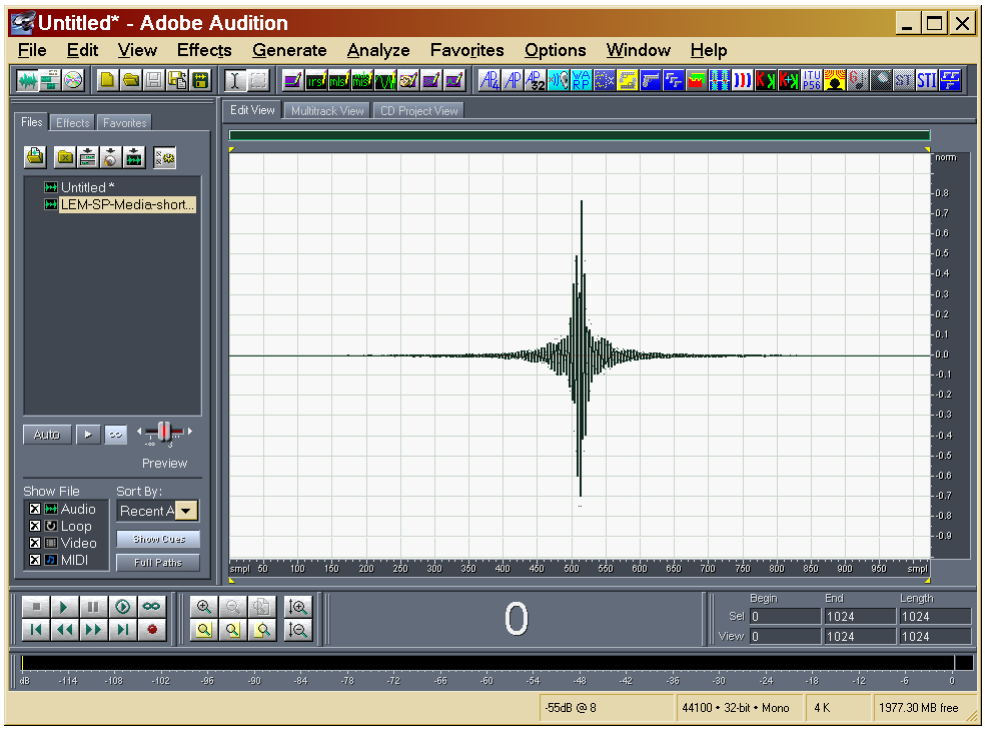


With this dialog box it is possible to select some parameters, which affect the computation of the Inverse Filter.
First of all there is the Filter Length: the larger this value, the most accurate is the frequency equalisation.
Then we have the Windowing function to be used before the FFT is applied to the original data.
The user can also limit the inversion within a useful frequency range, in this example 20 Hz to 20 kHz.
The "minimum phase" inverse filter requires a certain amount of "frequency smoothing": so we have to choose between Octave Smoothing or Linear Smoothing, specifying in either case the proper smoothing bandwidth.

When the OK button is pressed, the computation is made (it is almost instantaneous), and the following dialog box is displayed:



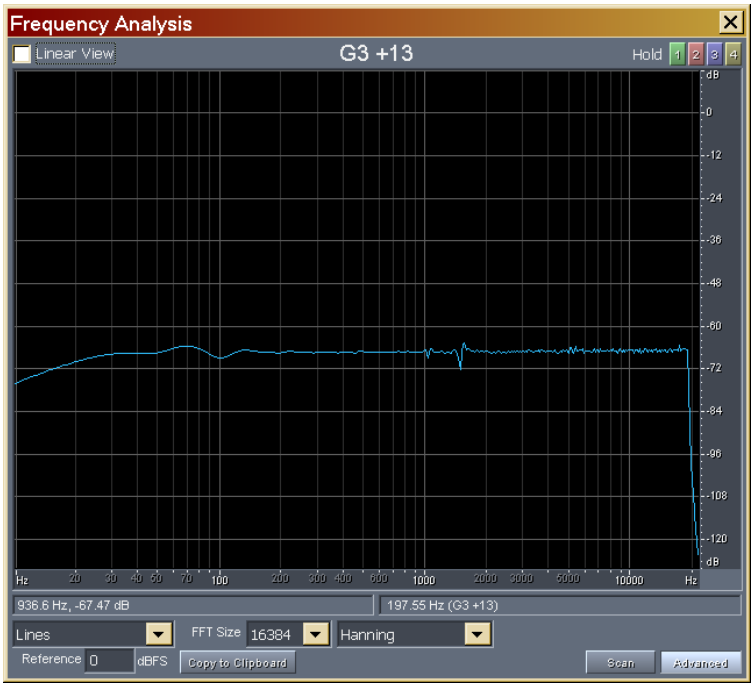
At this point the Inverse Filter has been stored in the Windows Clipboard. Making a Paste-New from the Windows Clipboard, the following signal is recovered:



And looking at its frequency response, we obtain:

This is roughly the specular image of the original Impulse Response, and thus the recovered signal is the required equalising filter.
For verification, if we apply it by convolution to the original signal we obtain the following:

And looking at the frequency response we obtain:



This is only almost flat, due to the limited number of taps employed for the creation of the Inverse Filter. And the frequency smoothing applied to the inverse filter.

$ K Inverse Filter

The creation of a causal, stable inverse filter for a given mixed-phase impulse response is generally an impossible task, and only approximate solution are available for this problem. This argument was deeply investigated by Mourjopoulos [12], particularly with the goal of obtaining a digital equalization of the room's response. Among the various techniques explored by him and others, the straightforward least-squares technique was chosen, as clearly explained in the Mourjopoulos paper. A least-square problem is set up, requiring that the unknown inverse filter, g(t), when convoluted with the original impulse response h(t), produces as result a delayed Dirac's delta function. After proper formulation, a standard linear equation system is formed, in which a square matrix [R] is multiplied by the unknown vector {g}, producing the known terms vector {k}.

$$[R] \cdot \{g\} = \{k\}$$

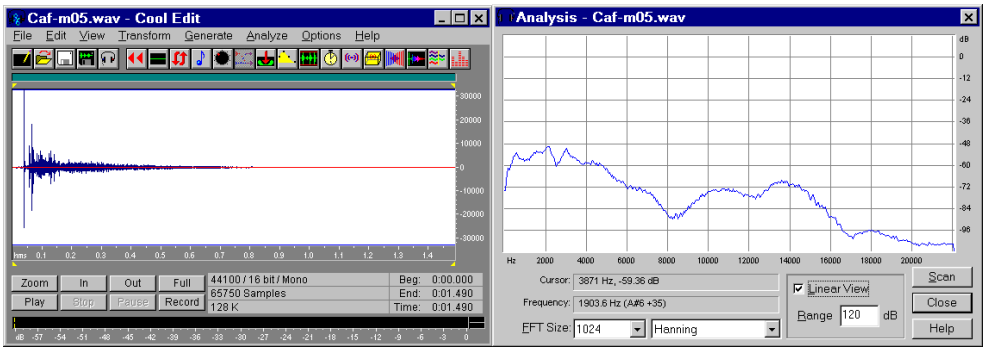
It turns out that the latter is simply a delayed, time reversed copy of the original impulse response h(t), while the [R] matrix is a Toepltiz matrix, in which the first row is simply the autocorrelation function of the original signal, and each of the subsequent rows is obtained by circular permutation (1 sample to the right) of the upper row. Fortunately, the solution of a Toeplitz system is possible with standard algorithms, included in any library of scientific functions. This way only the first row of the matrix has to been generated and stored, and the computation is thus possible also for very long IRs on standard computers. The successful creation of an inverse filter is anyway not always easy: if the frequency response of the original impulse response has very deep valleys, these turn out in strong peak of the inverse filters, which tends to become quite long. Furthermore, if the high frequency part of the spectrum of the original signal has low energy (this is common when an anti-aliasing filter is used), the inverse filter has maximum amplitude at these inaudible frequencies, and a very low gain at the lower frequencies, so that the filtered signal can fall below the numerical noise floor. For these reasons a certain pre-processing of the original impulse response is usually needed before attempting its inversion, but in this field only the experience and skillness of the user can be useful. Anyway, the inverse filter can always be quickly checked, applying it to the original impulse response or to other raw signals, by means of the convolution module. Often our ears are the better instrument for checking if the inverse filter is acceptable, or if it produces intolerable artifacts.

For example, the following picture reports an impulse response of a room excited by a loudspeaker and recorded by a microphone placed at a certain distance from it, so that a lot of reverberation and wall reflections are evident. This causes also an evident frequency response coloration, as shown in the same picture.

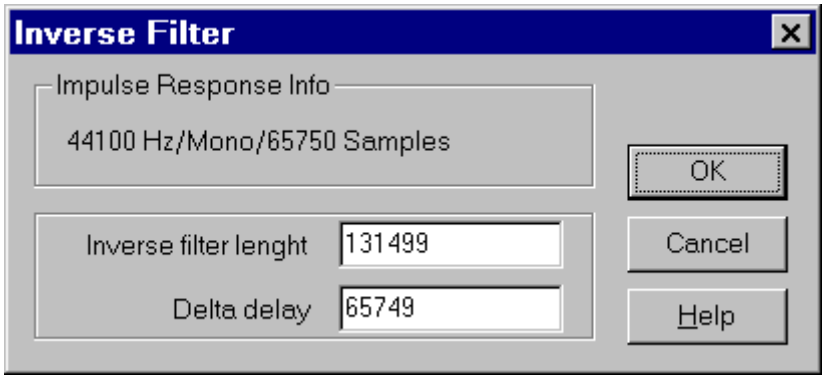
INVERT

$ Mourjopoulos’ Inverse Filter

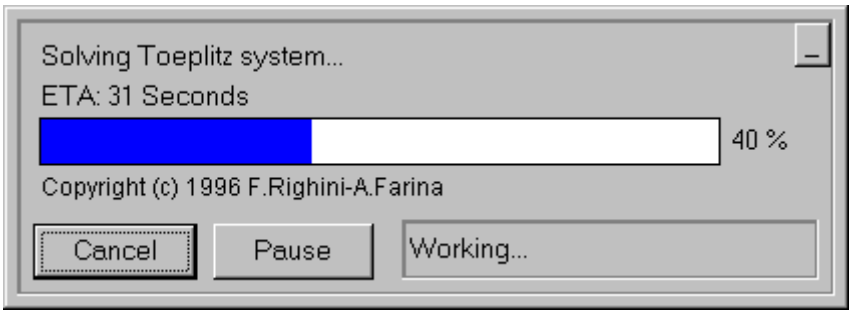
K Inverse Filter



Invoking the Transform / Inverse Filter module, a dialog box appears, asking for the length of the inverse filter (in samples) and for the wanted delay of the signal (this is necessary for ensuring that the inverse filter is causal, and is usually chosen around half the length of the filter). Mourjopoulos suggests a filter length approximately equal to the reverberation time of the room, although in some case shorter filters works reasonably well.

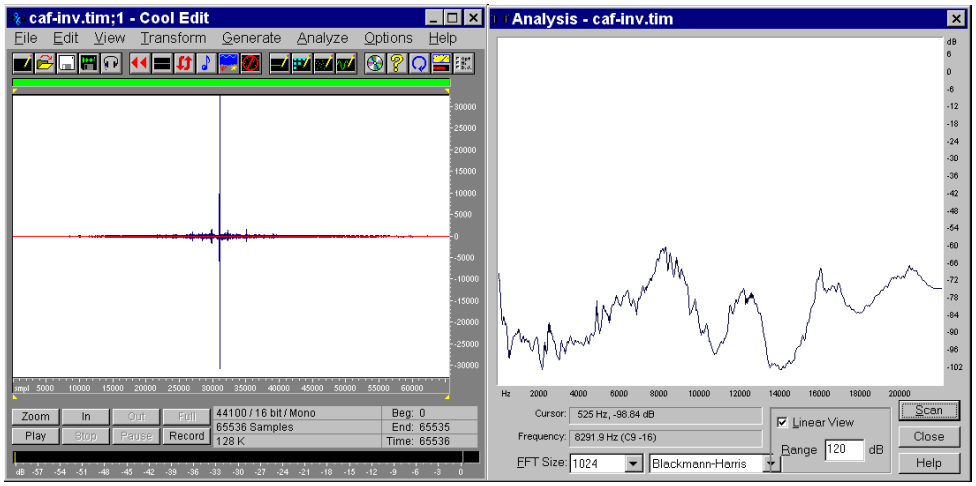


Then the solution of the Toeplitz problem is started: the computation time increases with the square of the filter length, because it is an iterative algorithm, which loops for a number of times equal to the filter length, and the number of operations inside each cycle is also proportional to the number of unknowns.

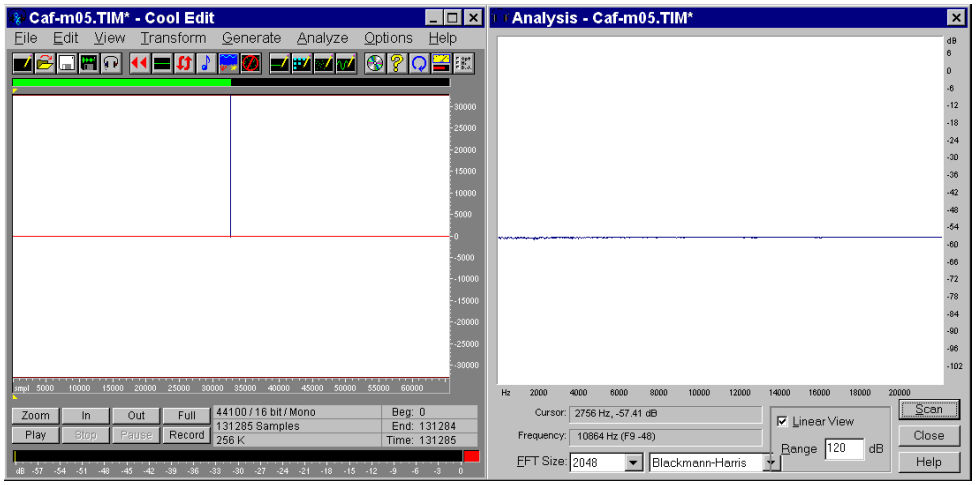


After a few seconds, anyway, a dialog box is displayed, informing the user that the inverse filter has been successfully stored in the Windows clipboard.

From there it can be recovered, opening a new instance of CoolEdit and pasting from the Windows clipboard. The following figure reports the inverse filter of the room response.



The verification of the proper working of the filter can be done by convolving the original impulse response with it. This is shown in the following pictures: an almost perfect Dirac's delta function has been obtained in the time domain, and in the frequency domain the response is flat.

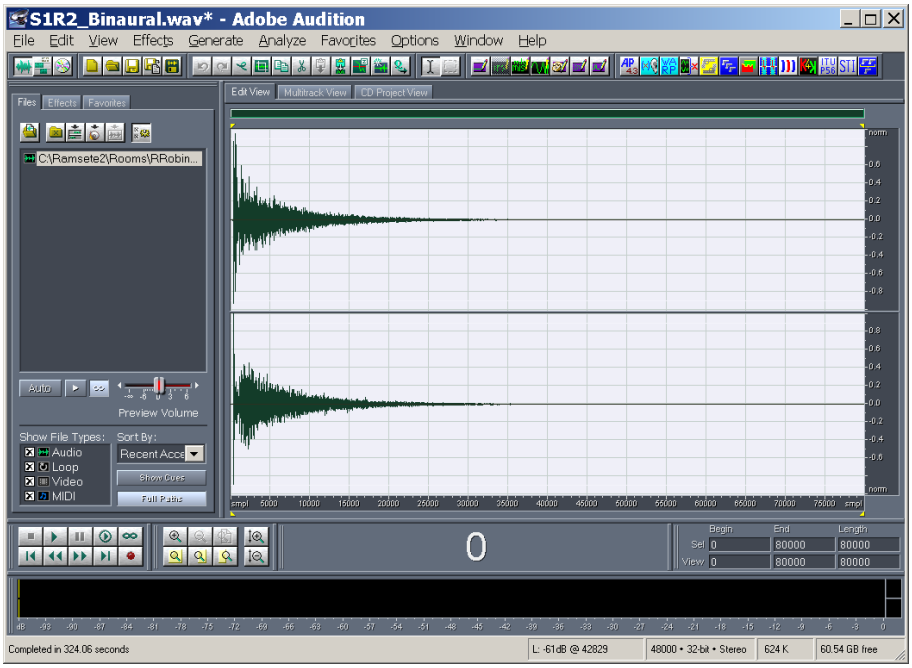


This way the effect of the transducers can be removed from a measured impulse response, or in general any recording done through a linear channel can be cleaned from its effects: in principle, a recording affected by too much reverberation can be de-reverberated, although this effect is quite difficult to obtain, because even a small difference in the position of the source and the receiver makes the inverse filter unmatched with the signal. Mourjopoulos already addressed this problem [12], and an enhanced dereverberation module is being developed now, following his principles.

[#] [$] [K] Warped Convolution with Clipboard

This module performs warped convolution of the currently selected waveform with the waveform actually present in the Windows Clipboard. This means that, before using this module, You have to load in the Windows Clipboard (not in the Audition's Clipboard !!!) the Impulse Response that You want to use as a warped filter.

For example, we can start opening the Impulse Response file:

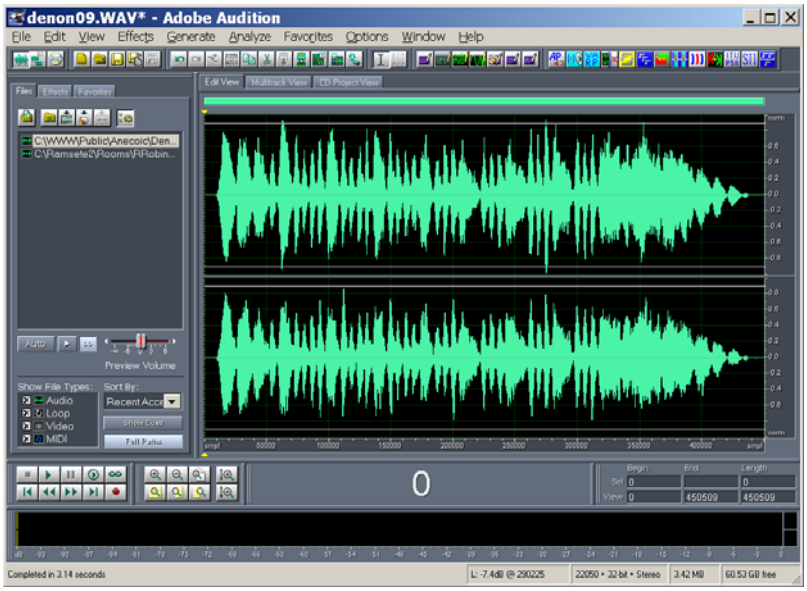


As the sampling rate of this Impulse Response is 48 kHz, we need to convert it to 22 kHz using the proper internal function of Audition. After copying the resampled IR to the windows Clipboard, we can load an anechoic sample:

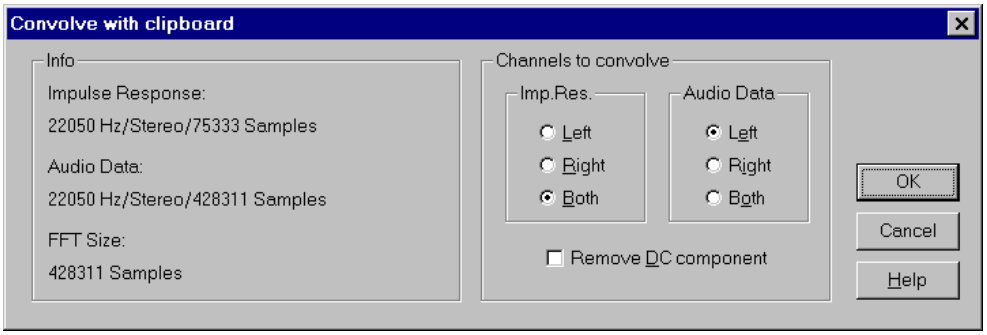
[#] CONVOWARP

[$] Warped Convolution with Clipboard

[K] Warped Convolution

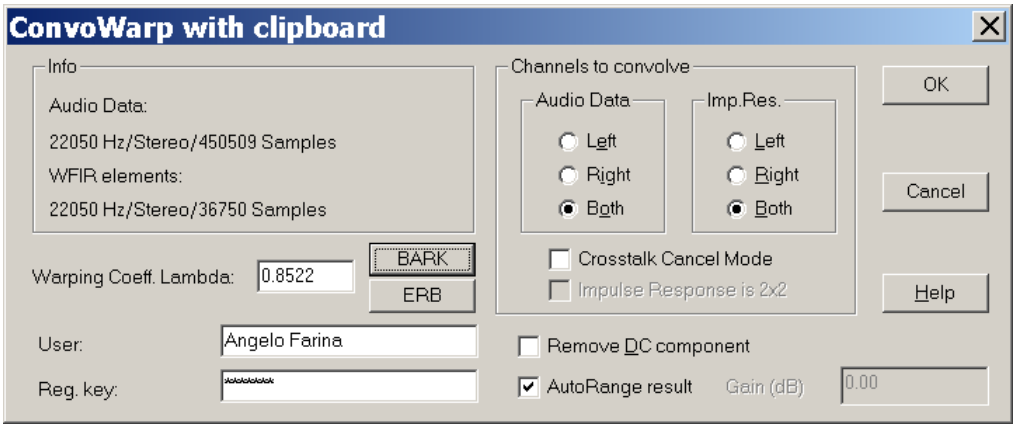


At this point, pressing the "Convolve with Clipboard"  button, the Following dialog Box appears:



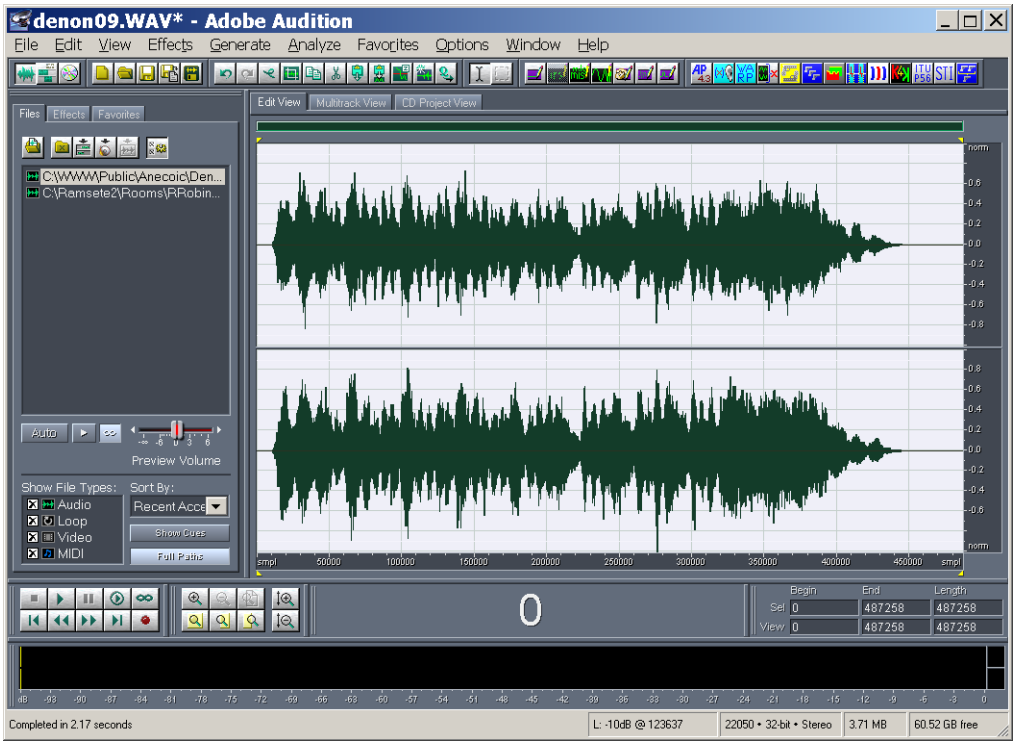
In this case, as it is shown, we want to convolve the Left channel of the current audio data with both the channels of the IR.

When OK is pressed, the standard Progress Meter of Audition appears, until the convolution process is finished. At this point the final dialog box is displayed, reporting statistics about the convolution process:



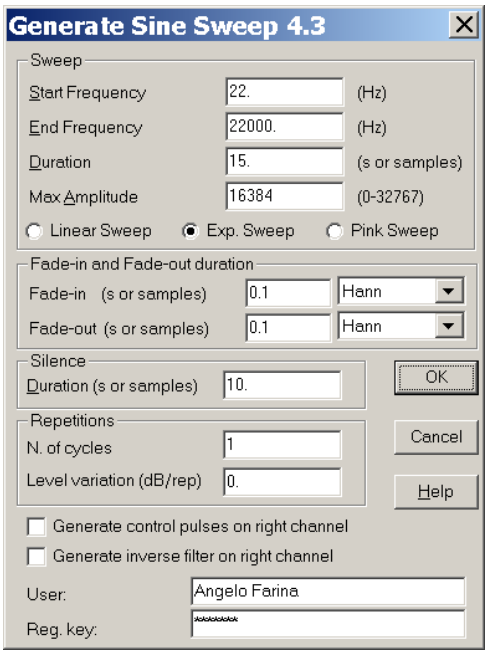
The Warping coefficient stretches the linear frequency scale of the filter (impulse response) to a not-linear scale. Two buttons allow for automatic selection of a warping coefficient resulting on a close approximation to the two most widely used notlinear frequency scales, the Bark scale and the ERB scale

Pressing OK, the new convolved waveform is displayed:



[#] [$] [K] Generate SWEEP Signal

This module generate multiple repetitions of the Sine Sweep signal (also known as Stretched Pulse, and not very dissimilar from TDS - Time Delay Spectrometry). In practice, it is a sinusoidal signal, which frequency is continously varied (not in steps) from a starting frequency up to an end frequency, covering the frequency range of interest. It is possible to produce multiple repetitions of the test signal, with an optional silence gap between them:



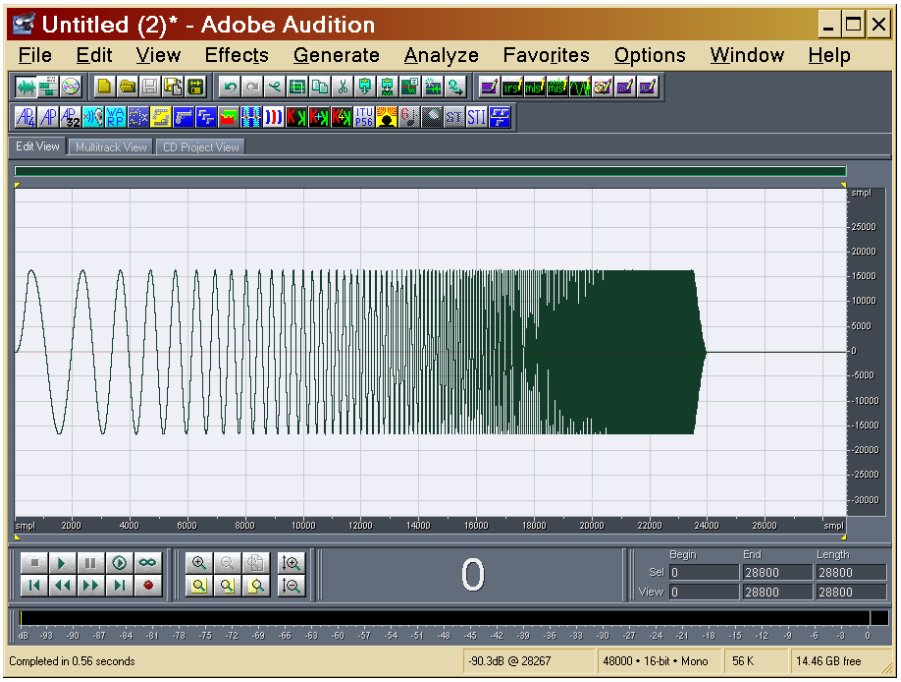
If a stereo waveform is generated, the selected number of repetitions is placed first on the left channel, thereafter on the right channel.

It is also possible to generate control pulses on the right channel, for triggering the advancement of a rotating table, as already explained with reference to the Generate Multiple MLS Signal module. In this case, anyway, a single repetition (cycle) of mono signal is produced, with exponential frequency variation ("pink", or log, spectrum):

[#] GENSWEEP

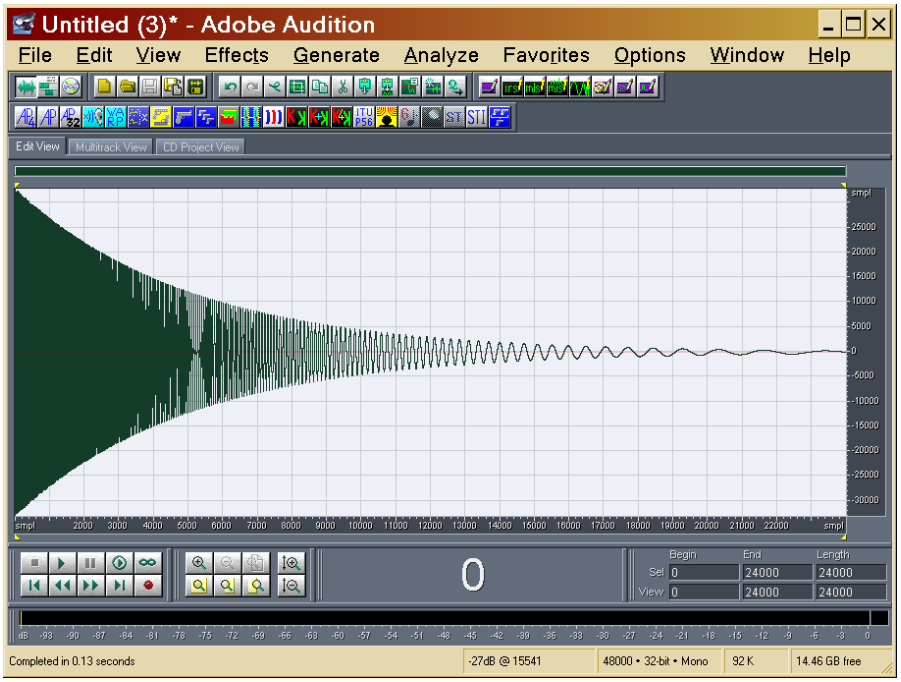
[$] Generate Sine Sweep test signal

[K] Generate Sine Sweep



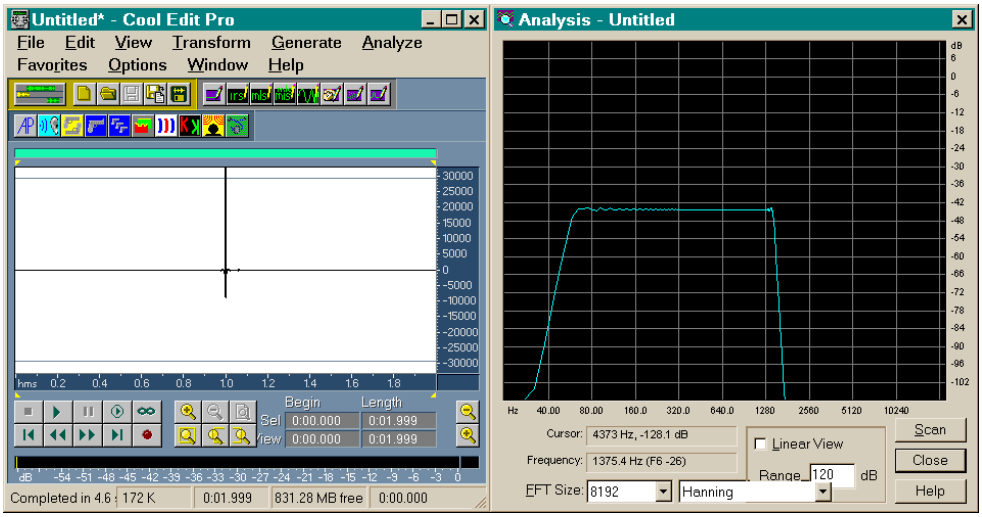
As it is shown, a proper fade-in and fade-out at the extremes of the signal are automatically performed.

During the generation of the excitation signal, a proper inverse filter for the deconvolution of the impulse response is preloaded onto the Windows clipboard:



For recovering the impulse response, the recording made after the excitation of the system with the sweep signal must simply be convolved with the inverse filter. This is

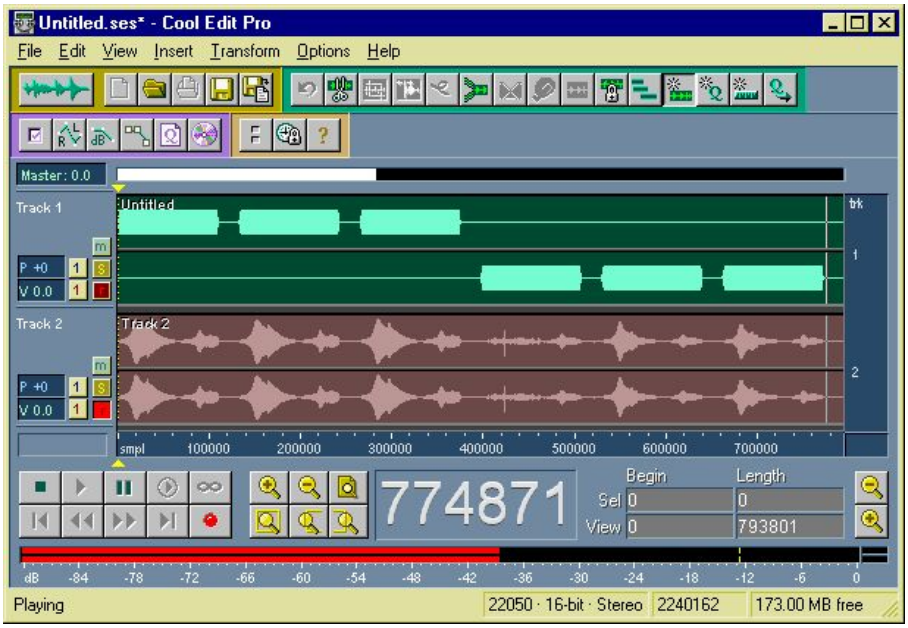
demonstrated by the fact that, convolving the above signal with its inverse filter, an almost perfect band-limited Dirac's delta function is obtained:



[#] [$] [K] Synchronous Average

This module retrieves the system's impulse response(s) from a recording made after excitation with the signal generated through the **Generate SWEEP Signal** module. For example, here is presented the case of a the measurement of a two-loudspeaker system in front of a stereo microphone, in which a repetition of 3 sweeps is played first on the left channel, then on the right channel, and the two microphonic channels are simultaneously sampled.

This processing is suitable only for measurement made with synchronous playback and recording, typically made under CoolEditPro in Multitrack mode, playing the excitation signal on track 1 and simultaneously recording the binaural response on track 2, as shown here:

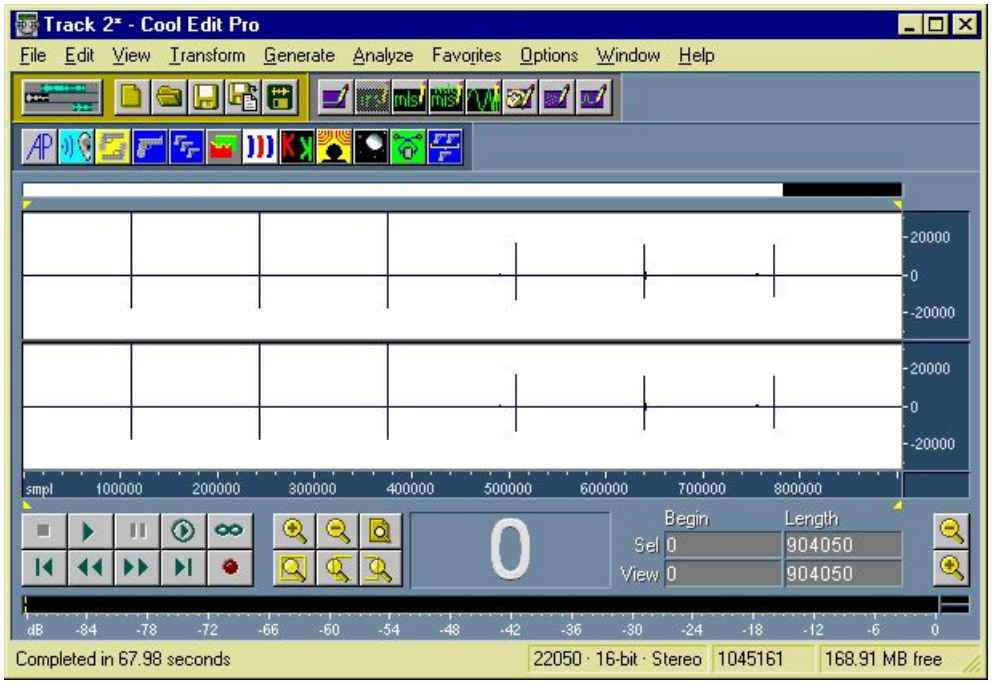


The deconvolution is made, in this case, simply convolving the recorded response (track 2) with the clipboard, where the proper inverse filter was already stored from the **Generate SWEEP Signal** module. This way, the following sequence of raw impulse responses is obtained:

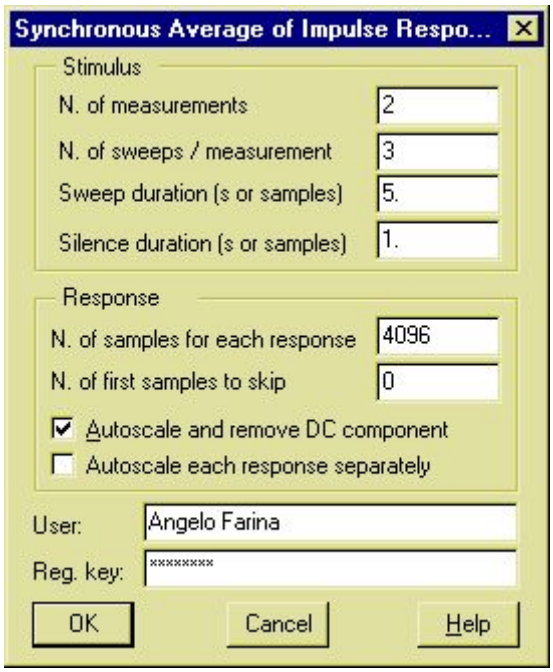
[#] SYNCAVG

[$] Synchronous Average of impulse responses

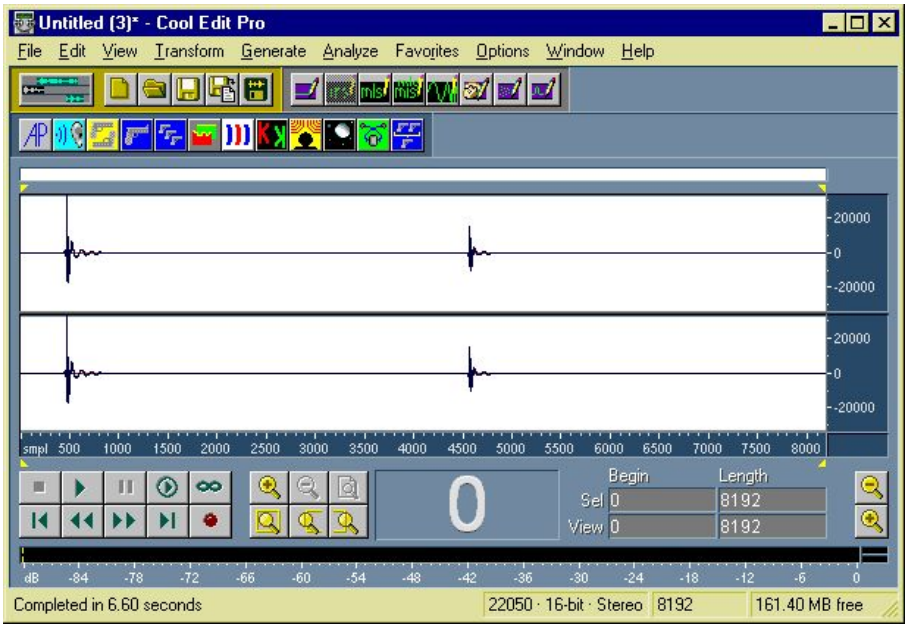
[K] Synchronous Average



At this point, we want to average together the responses to the three repetitions of each signal, while maintaining separated the response of the left and right loudspeaker. Furthermore, we want to keep only the first 4096 points of each response. All this is done with the Synchronous Average module, setting properly its parameters as shown here:



After the averaging, a set of 4 impulse response is obtained. First are the binaural IR of the Left loudspeaker, followed by the binaural IR of the Right loudspeaker:



The impulse response files, in this format called 2x2, are already suitable for the calculation of cross-talk canceling filters through the Invert Kirkeby module.

$ K IR Selection

This module retrieves the system's impulse response(s) from a recording made after excitation with the signal generated through the Generate SWEEP Signal module. It differs from the Synchronous Average module for three facts:

- No average is done, and consequently this module can only be used if a single sweep is done for each measurement.

- The results are overwritten over the originally selected waveform, without employing the Windows clipboard

- the IR Select module also allows for extraction of the “harmonic distortion” impulse responses, which precede the linear ones.

This module is particularly suitable for processing measurement sets coming from a rotating table. Typically 36 or 72 measurements are made, forcing the advancement of the rotating table by generating a suitable pulse on the right channel of the test signal.

The processing of such large data sets with this module is much faster than with Synchronous Average.

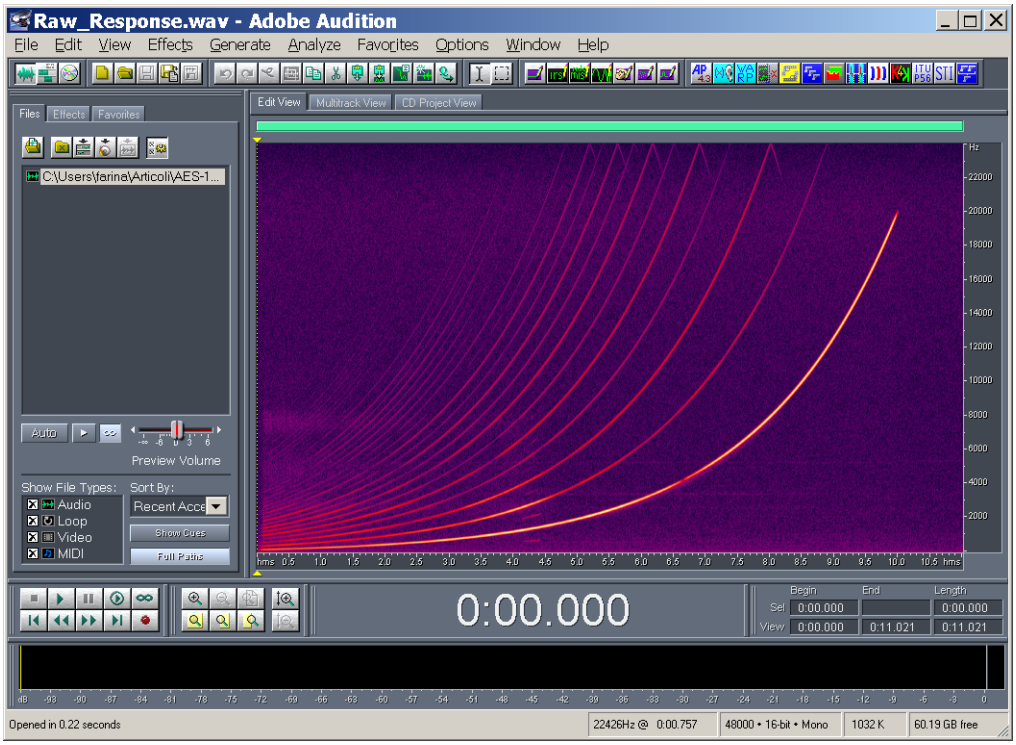
The example presented here refers however to a single measurement performed with a single sweep on a strongly notlinear system (a small loudspeaker pushed well abov its limits).

After generating a sine sweep test signal, playing it through the loudspeaker and recording the sound emitted from it, the sonogram of rerecorded sound show significant harmonic distortions:

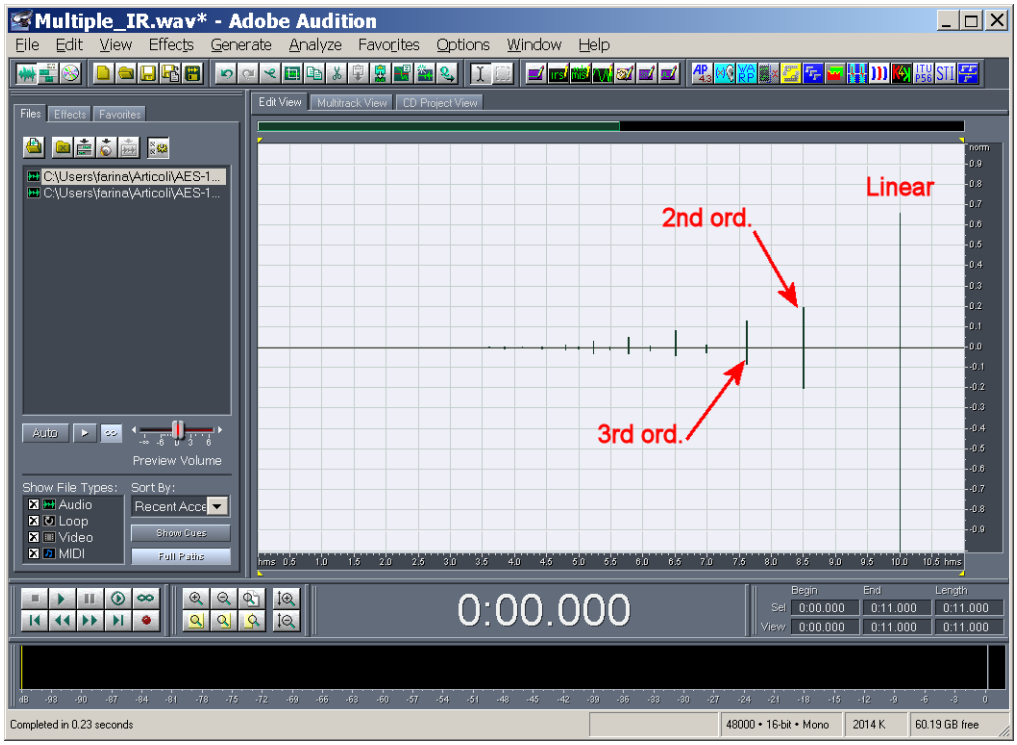
[#] IRSELECT

[$] Impulse Response selection

[K] Impulse Response selection

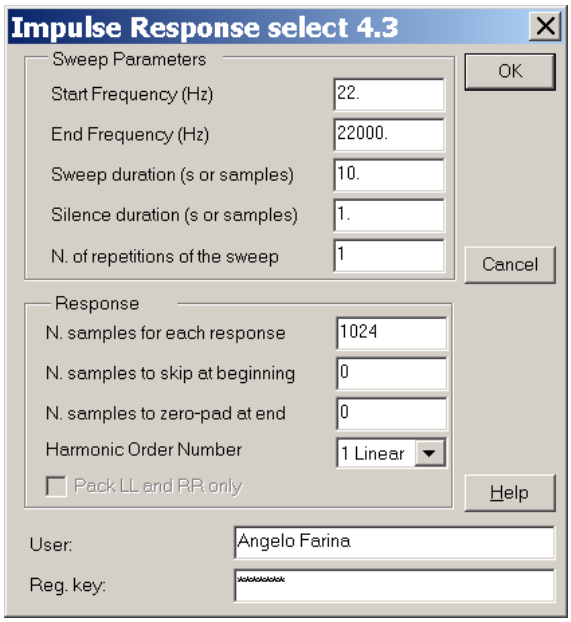


If now we convolve with the inverse sweep (which was automatically generated by the Gensweep module, and preloaded into the windows clipboard), we get the system's impulse response:

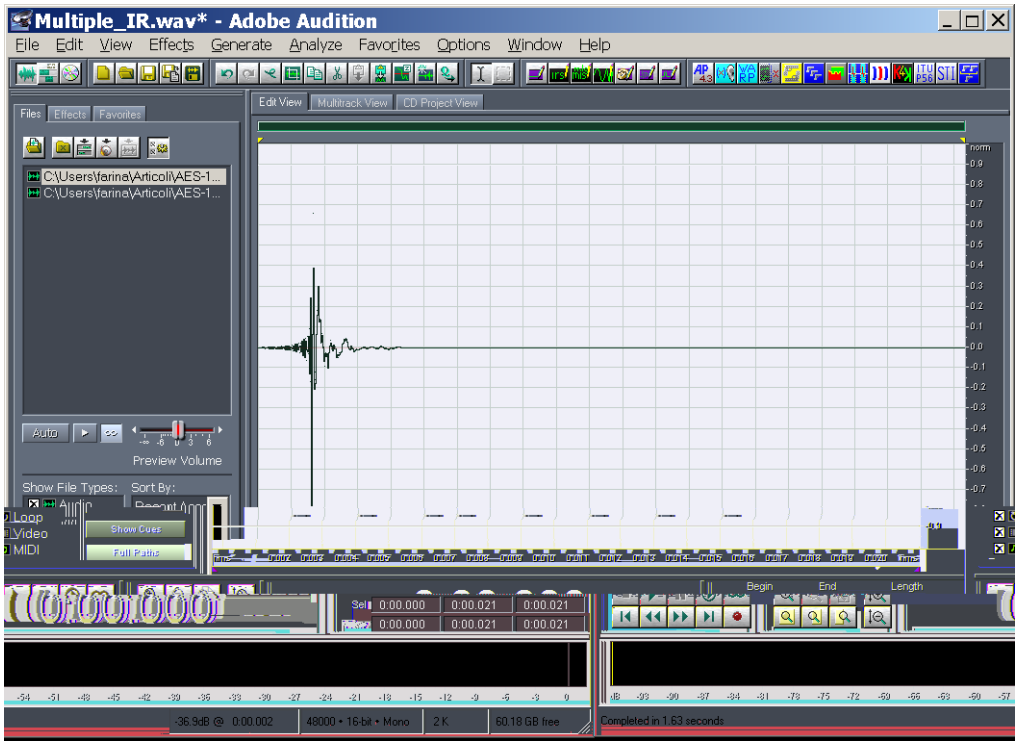


The last peak is the linear impulse response, the preceding ones are the harmonic distortion IRs of increasing orders…

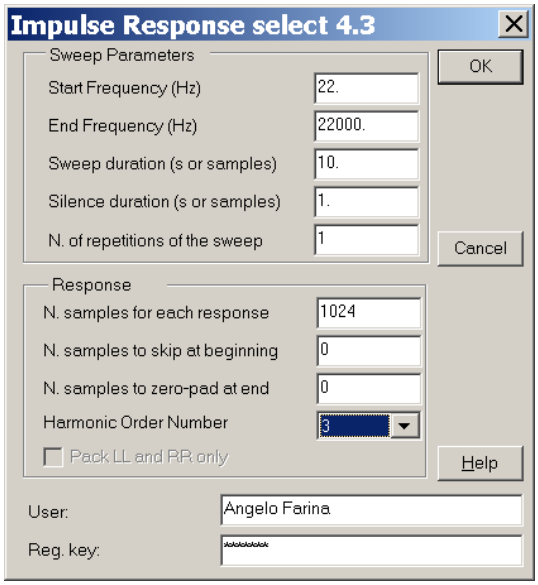
We can now invoke IR select, for extracting just the linear IR:



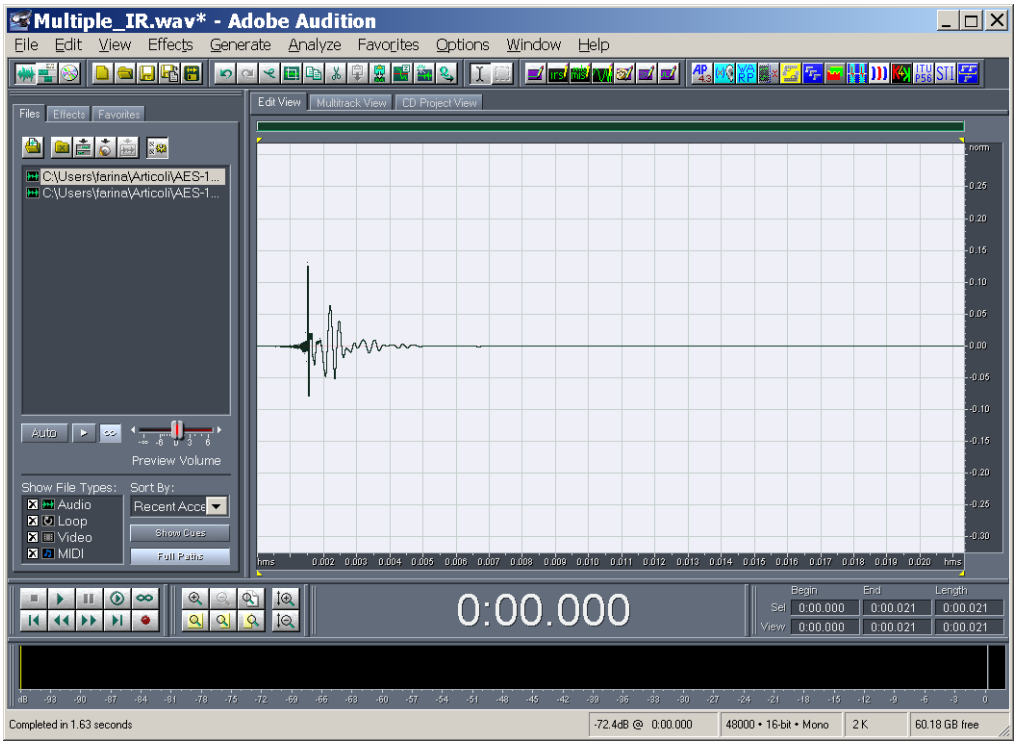
And we get this result:



We can now save it, then Undo the IR selection. Starting again from the multiple-IR signal, we can now IR-select, for example, the 3rd-order IR:



And we get this:



One by one, we can this get a complete extraction of all the harmonic orders, By analyzing separately each of them, the complete harmonic distortion spectrum can be obtained for each order.

[#] [$] [K] Kirkeby's Inverse Filter

This module computes an inverse filter according to the Nelson/Kirkeby frequency-domain regularization. If applied to a stereo or 2x2 impulse response, it also automatically computes the cross-talk cancelling filters for binaural reproduction over a pair of loudspeakers, optionally placed in the Stereo Dipole configuration.

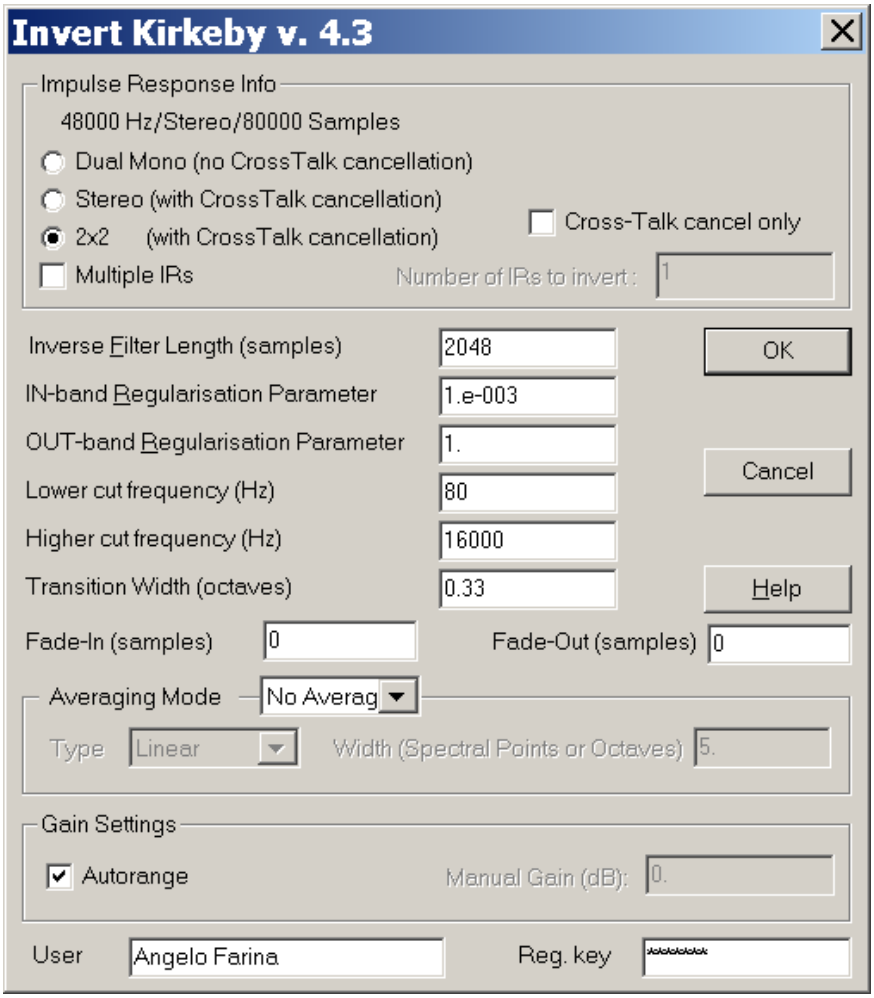
Here a quick examples is shown. First of all, we load a set of 4 impulse responses measured with binaural microphones in front of a loudspeaker pair, and stored in the 2x2 format:

Now we invoke the Kirkeby4 module, which has the following graphical interface:

[#] KIRKEBY

[$] Kirkeby's Inverse Filter

[K] Kirkeby's Inverse Filter



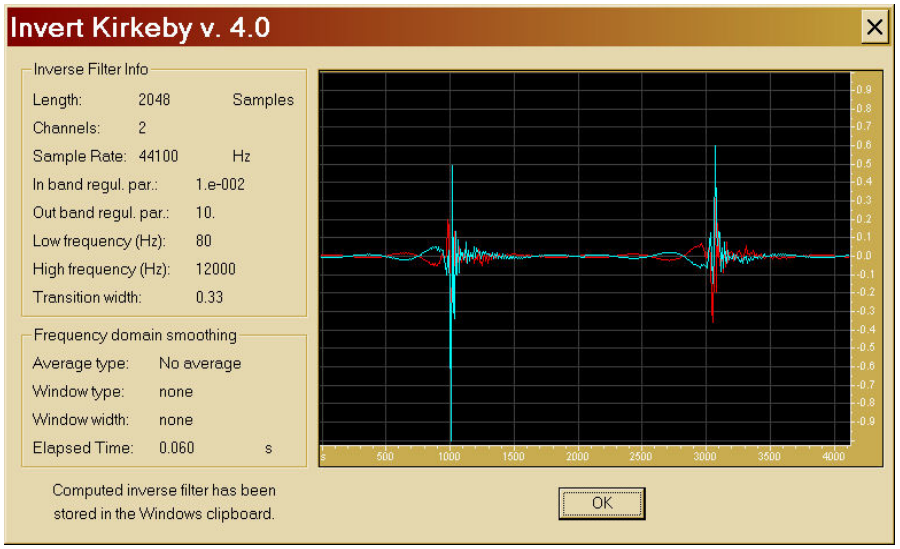
The choice of the parameters can be tricky: in this case, we did choose a filter length of 2048 samples, because it is more than twice the length of the original anechoic segment of the head-related transfer function which we want to invert.

The values of in-band and out-band regularization parameters has to be done by trial-and-error: the larger these values, the shorter will be the inverse filters computed.

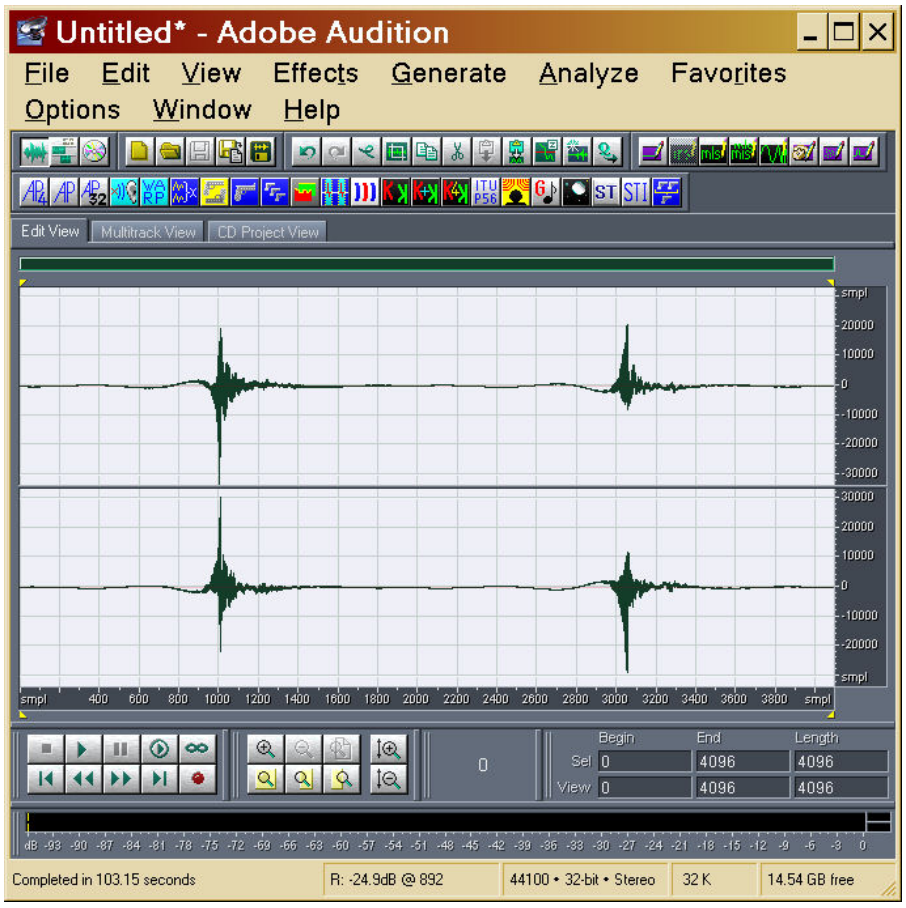
The frequency range inside which "accurate" inversion is done has to be chosen according to the frequency limits of the tramsducers employed (usually the loudspeakers are the limiting components of the whole system, particularly at low frequency).

Finally, for applications where very accurate inversion is not required, it is possible to apply some frequency response smoothing prior of the inversion: this generally produces shorter filters with better dynamic range, at the expense of some residual "ripple" in the frequency response.

The computation of the inverse filters is instantaneous, and this is the result:

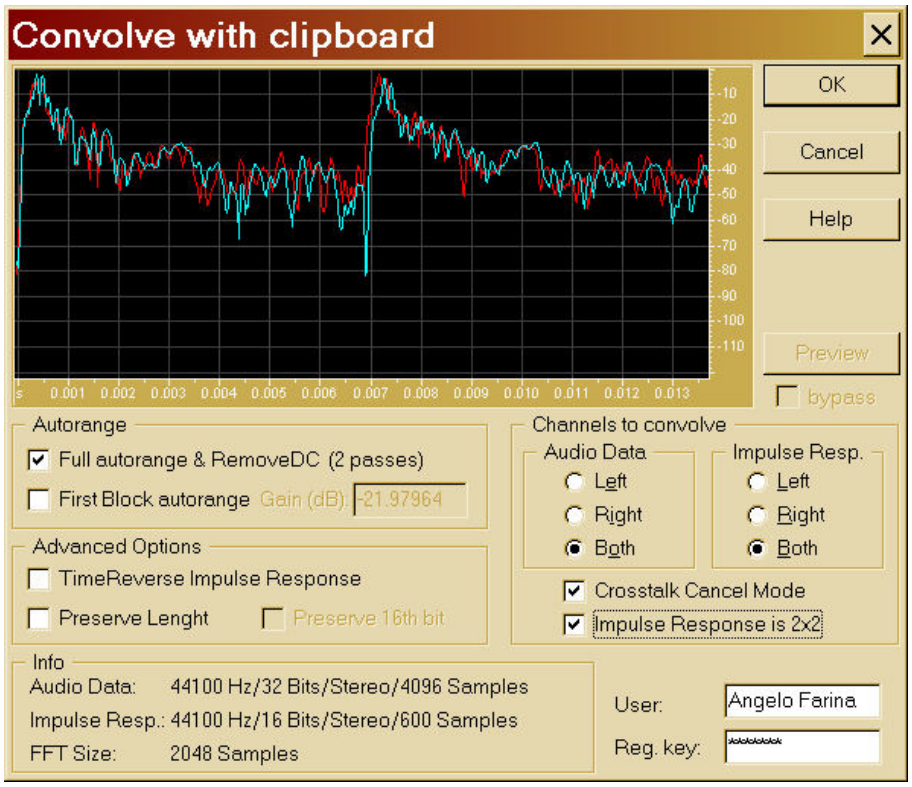


The inverse filters are now in the Windows clipboard, form where they can be pasted to a new waveform:

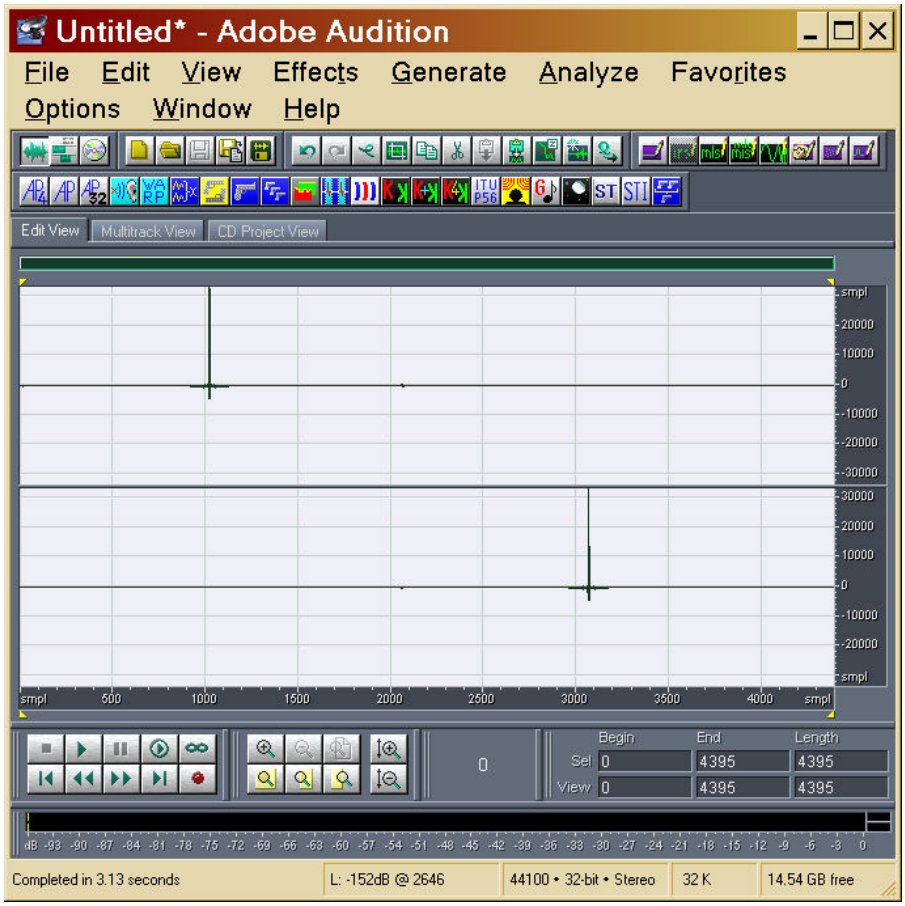


We can now check how good these cross-talk cancelling filters are, by convolving them with the original set of measured HRTF.

We use the **Linear Convolution with Clipboard** module for this:



And this is the result:


Untitled* - Adobe Audition
File Edit View Effects Generate Analyze Favorites
Options Window Help
Edit View
Multitrack View
CD Project View
smpl
20000
10000
0
-10000
-20000
-30000
500
1000
1500
2000
2500
3000
3500
4000
Begin
End
Length
Sel 0 4395 4395
View 0 4395 4395
Completed in 3.13 seconds
L: -152dB @ 2646
44100 • 32-bit • Stereo
32 K
14.54 GB free

$ K Cross Functions

This module computes the most usual cross-functions between channel left and right: autocorrelations, cross correlation, impulse responses h1, h2 and h3.

This module can also operate advanced processing of signals coming form a pressure-velocity (intensity) probe, with the aim of computing the sound absorption coefficient by means of the new Intensity/Density method.

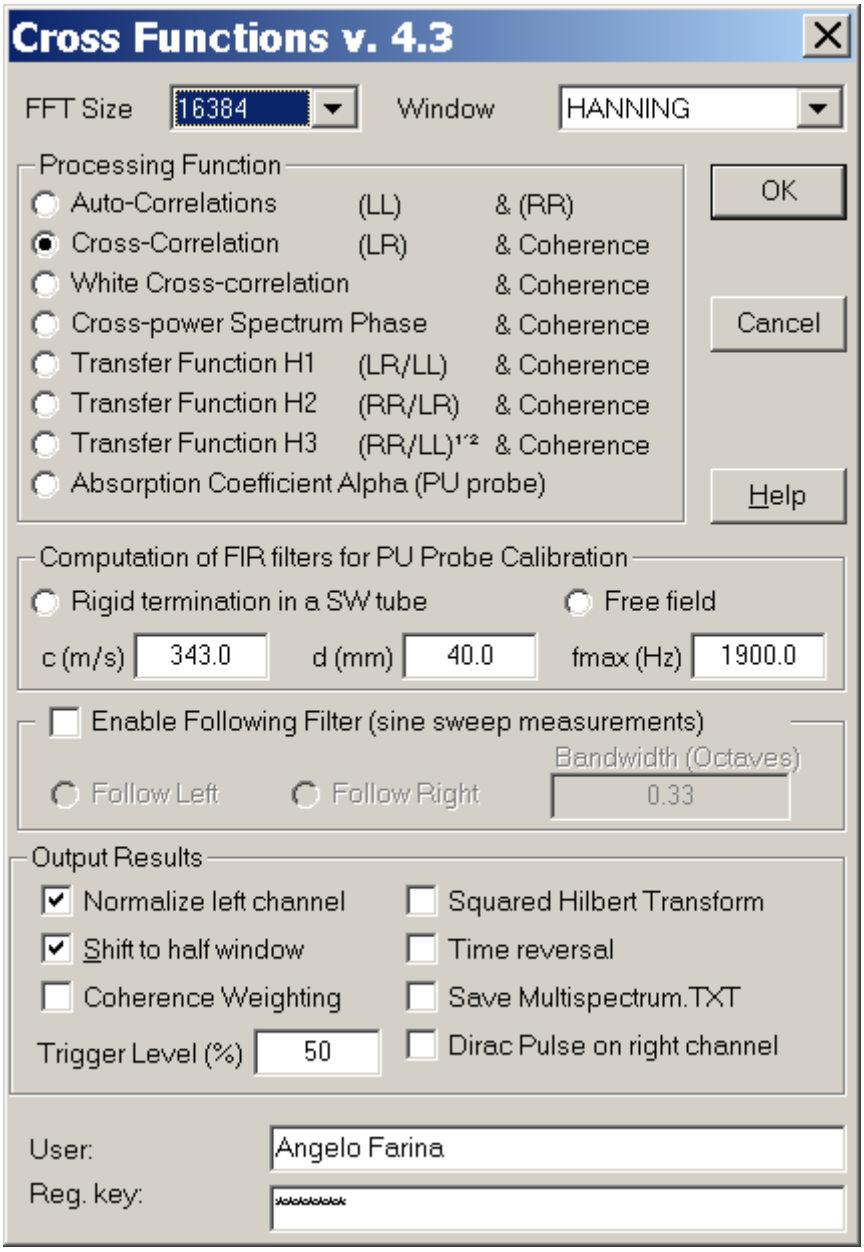
The result can be optionally Hilbert-transformed, for showing the magnitude envelope. During the processing a "following filter" can be applied to the signal, tracking the peak frequency of one of the two channels. This can be very useful for improving the S/N ratio in measurements conducted with the sine sweep method.

This is the user's interface of the Cross-functions module:

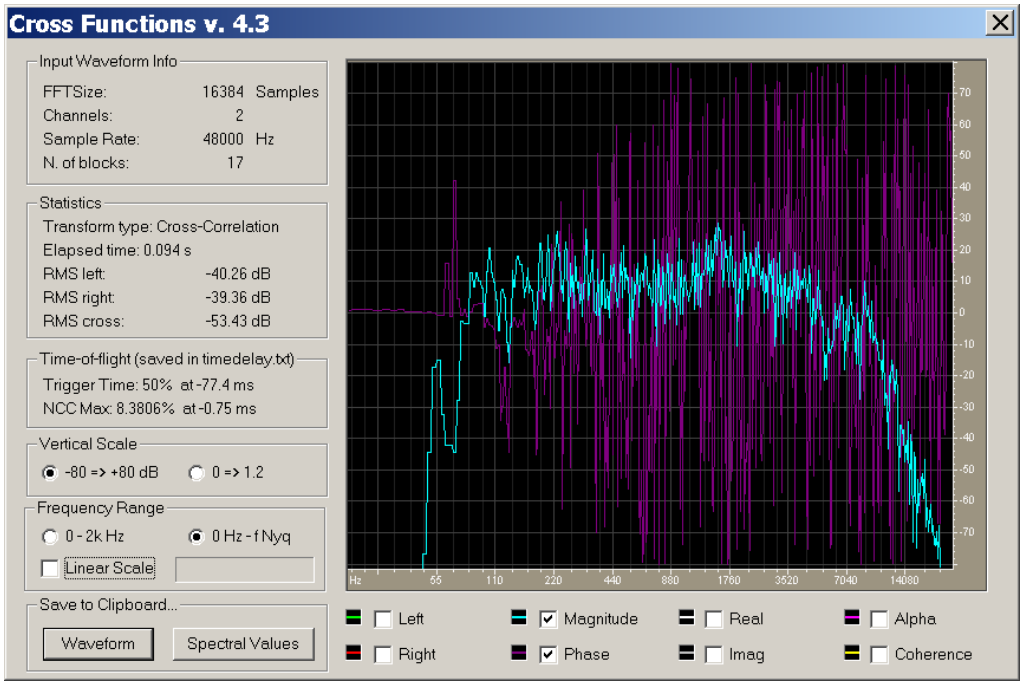
[#] XFUNC

[$] Cross Functions

[K] Cross Functions



When the computation is done, the results are displayed in Frequency Domain, with the possibility to show then as Magnitude & Phase or as Real & Imag, as shown here:



The results can now be exported to the clipboard, either as Frequency Domain data, or as time-domain waveform (after a proper IFFT is performed for going back to time domain)

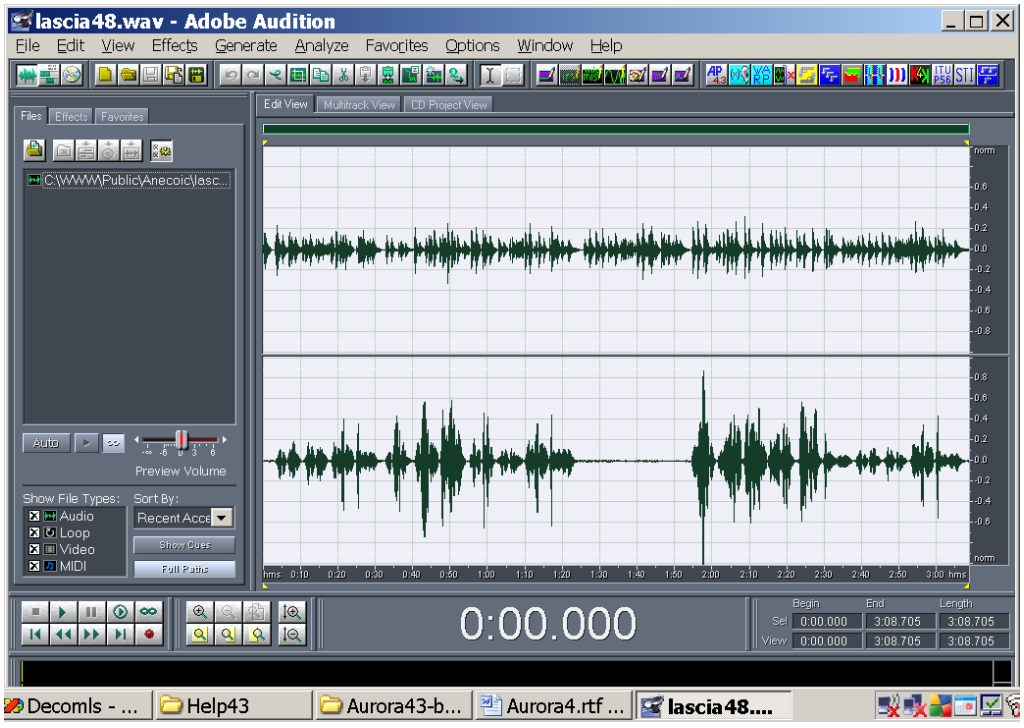
A complete understanding of the functionality of this module would require a complete textbook, so it cannot be explained here…

$ K ITU P56 Active Speech Level

This module performs the analysis of the temporal fluctuactions of the level of a recorded sound. It has basically three possible functions:

- measurement of the Active Speech Level of the human voice signal according to ITU-P56. The ITU frequency-limiting curve can be applied.
- measurement of standard acoustical parameters Leq, Lmax (with time weightings Slow, Fast, Impulse) and detection of impulsive events (whenever LmaxImpulse exceeds LmaxSlow by more than 5 dB). If the recording was linear, the IEC "A-weighting" curve can be applied.
- analysis of the reaction of the human body to whole-body vibrations, by applying the generic-axis acceleration weighting curve according to ISO 2731, and then looking at the Max Slow level.

We start loading a stereo recording of speech, as shown here:

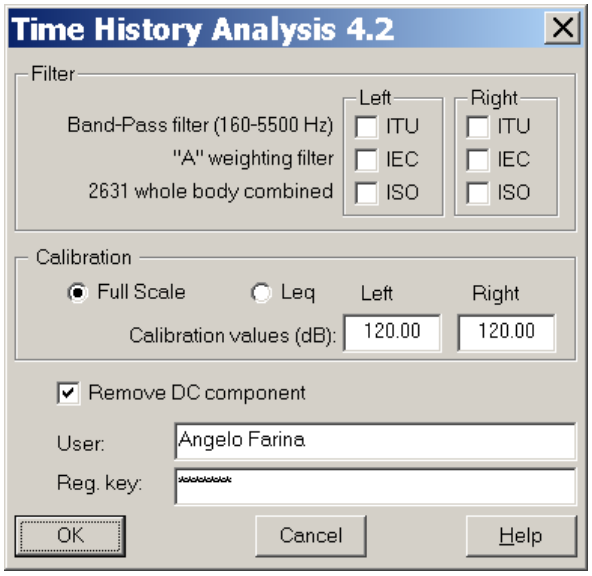


Now we invoke the ITUP56 module, and the following window appears:

ITUP56

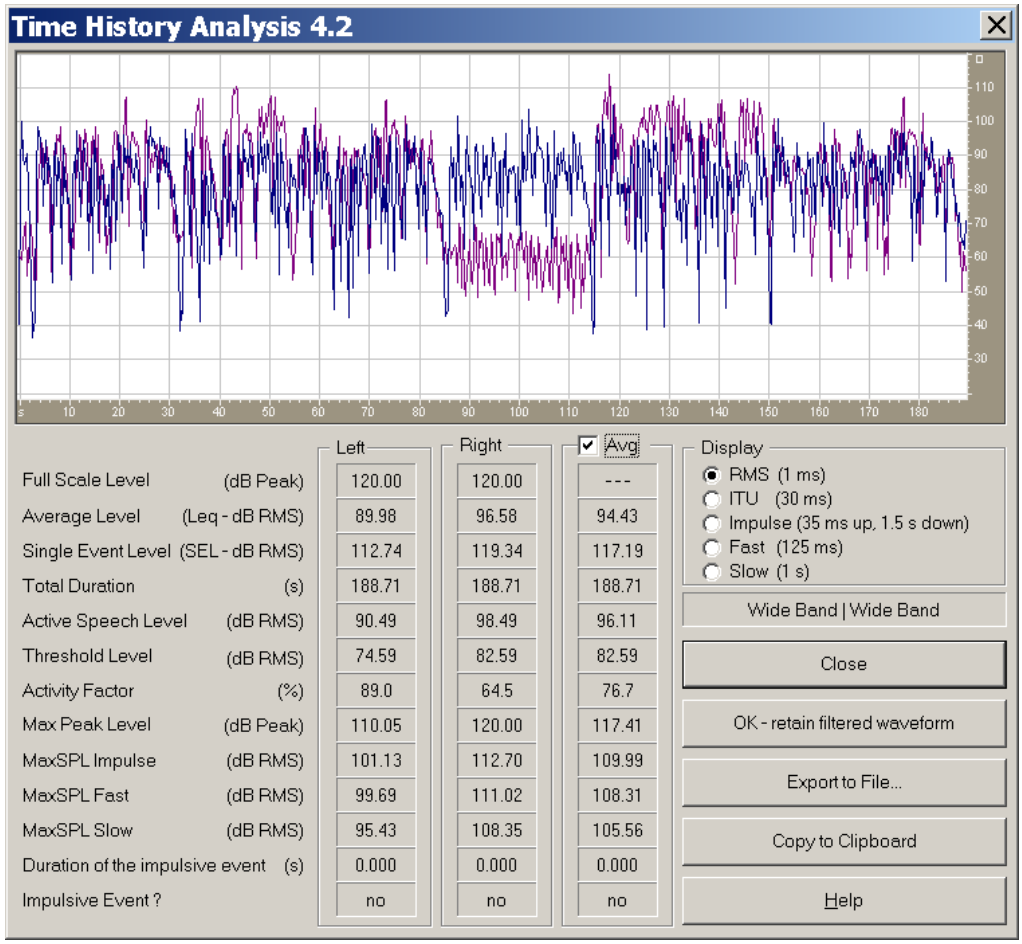
$ Active Speech Level according to ITU P56

K ITU P56 Active Speech Level



Please notice that this module allows for calibration: if the true SPL (Leq) value of the recording is known (typically because a recording of the calibrator is being analyzed), then the user can specify this value as Leq (typically 94 dB). The following times the module is employed again, for analyzing real recordings, the full-scale value will remain set to the correct value, ensuring that the displayed values of SPL are the correct ones.

Pressing OK, the result window appears:


Time History Analysis 4.2
Left
Right
Avg
Display
RMS (1 ms)
ITU (30 ms)
Impulse (35 ms up, 1.5 s down)
Fast (125 ms)
Slow (1 s)
Full Scale Level (dB Peak) 120.00 120.00 ---
Average Level (Leq - dB RMS) 89.98 96.58 94.43
Single Event Level (SEL - dB RMS) 112.74 119.34 117.19
Total Duration (s) 188.71 188.71 188.71
Active Speech Level (dB RMS) 90.49 98.49 96.11
Threshold Level (dB RMS) 74.59 82.59 82.59
Activity Factor (%) 89.0 64.5 76.7
Max Peak Level (dB Peak) 110.05 120.00 117.41
MaxSPL Impulse (dB RMS) 101.13 112.70 109.99
MaxSPL Fast (dB RMS) 99.69 111.02 108.31
MaxSPL Slow (dB RMS) 95.43 108.35 105.56
Duration of the impulsive event (s) 0.000 0.000 0.000
Impulsive Event ? no no no
Wide Band | Wide Band
Close
OK - retain filtered waveform
Export to File...
Copy to Clipboard
Help

[#] [$] [K] STI (Speech Transmission Index)

This module computes the Speech Transmission Index (STI), according to the standard IEC 60268-16 (2003). The measurement of STI requires substantial expertise, and special equipment, such as an artificial mouth (a special loudspeaker system emulating the directivity and spectral emission of the human voice).

There are basically two ways to measure STI:

- measurement of a "noisy" impulse response employing MLS and no averaging
- measurement of a "noiseless" impulse response, employing MLS or, better, Sine Sweep, and separate measurement of the octave spectra of the Signal and of the Noise

STI is dominated by the Signal-To-Noise ratio, and any factor altering this delicate balance will consequently result in wrong values of the STI parameter.

Let us assume to use the second technique ("noiseless" IR and separate estimation of the spectra of Signal and Noise), as this provides much better repeatability and requires less effort.

We assume that the reader has already mastered the usage of Aurora for measuring impulse responses, either with the MLS or the Sine Sweep approaches.

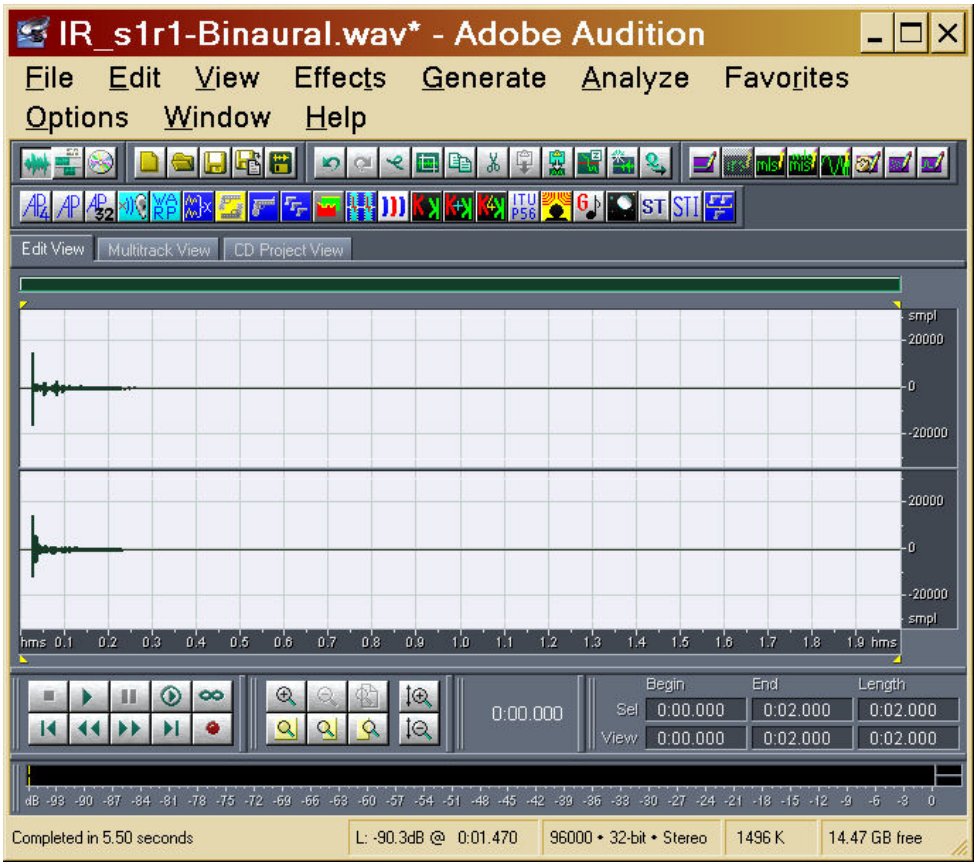
As the IR measurement is noiseless, we can avoid the troubles caused by setting the mouth simulator for radiating the "official" SPL value (62 dBA at 1m from the lips) and the "official" male or female spectra. Instead, we can employ the mouth simulator at maximum available power, and with flat equalization (or even no equalization at all). In fact, the goal of IR measurement is just to estimate reverberation and echoes, and not the Signal-to-Noise ratio, which will be measured separately.

So we play the test signal through the mouth simulator, we record the room response through the monoaural or binaural microphone, and we deconvolve the room impulse response. Let's assume that this way we obtained a binaural IR, as shown here:

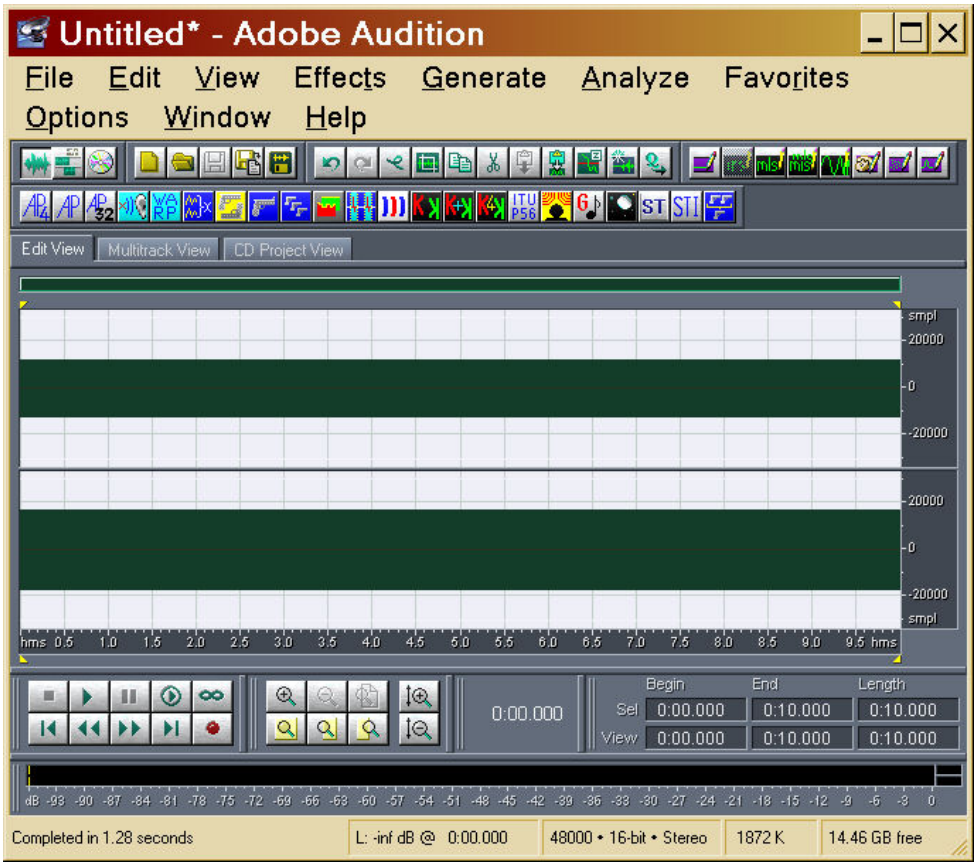
[#] STI

[$] Speech Transmission Index (STI, RASTI)

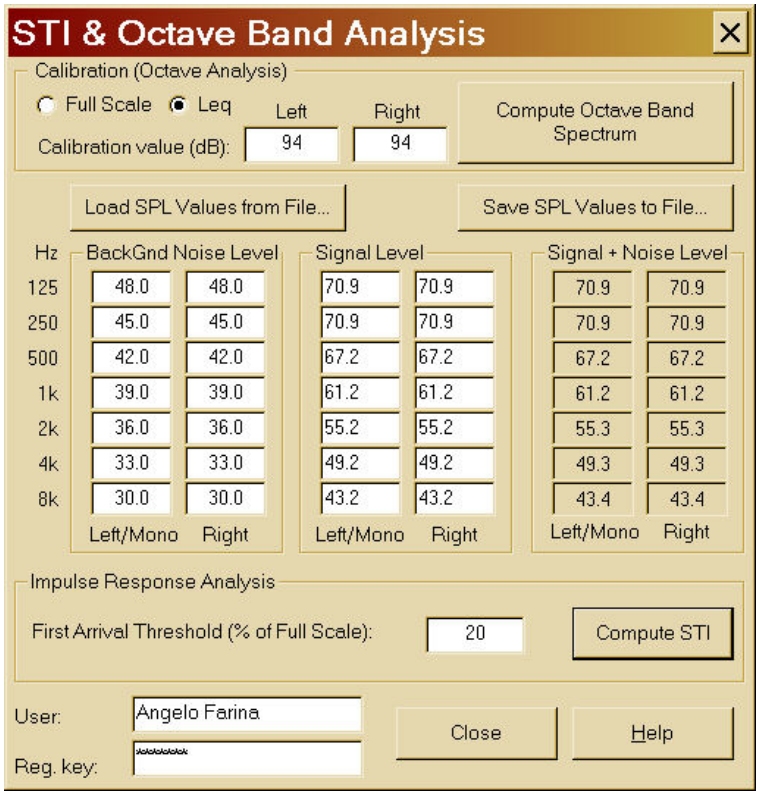
[K] Speech Transmission Index



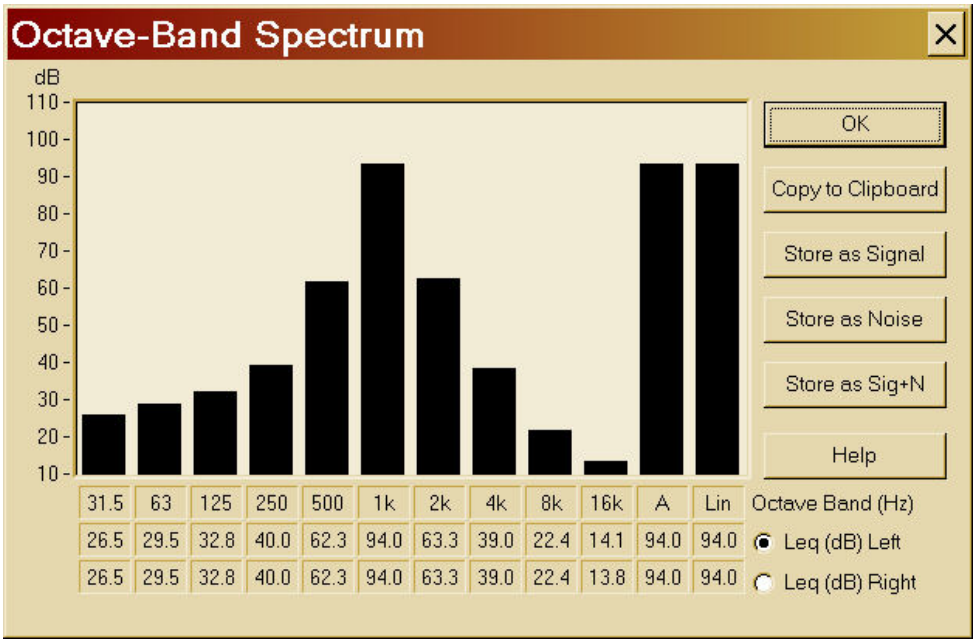
Now we need to make a recording of the background noise: but first we record 30s of calibration signal (at 94 dB SPL), as shown here:



And we calibrate the amplitude scale of the STI module. We invoke it:



And we specify the Leq value for each of the two channels - then, pressing the button labelled "Compute Octave Band Spectrum", the following result appears:



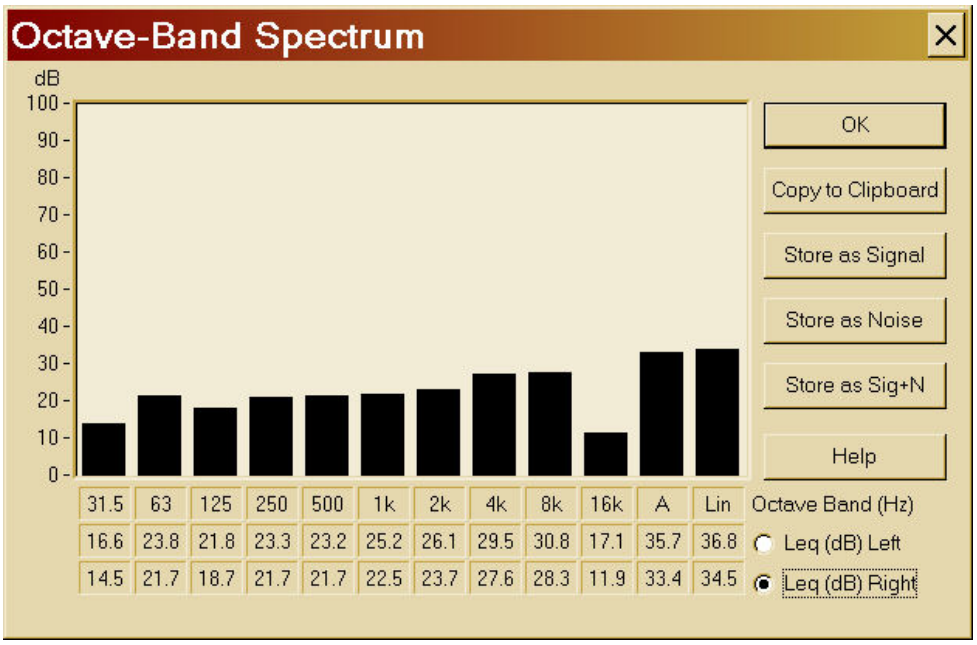
We can press OK. Now the amplitude scale of the STI module is correctly calibrated, and the next time we will not need anymore to specify the Leq value.

We perform now the recording of the background noise, and the recording of the signal.

The first one is done without any signal emitted by the artificial mouth, the second is done emitting a filtered pink noise. The filtering is generally necessary for ensuring that the free-field signal, measured at 1 m in front of the lips of the artificial mouth, corresponds exactly with the prescribed spectrum for the "male" talker specified inside the IEC standard. The total gain has to be adjusted so that the signal is exactly 62 dB(A) at 1m, when measured in free field.

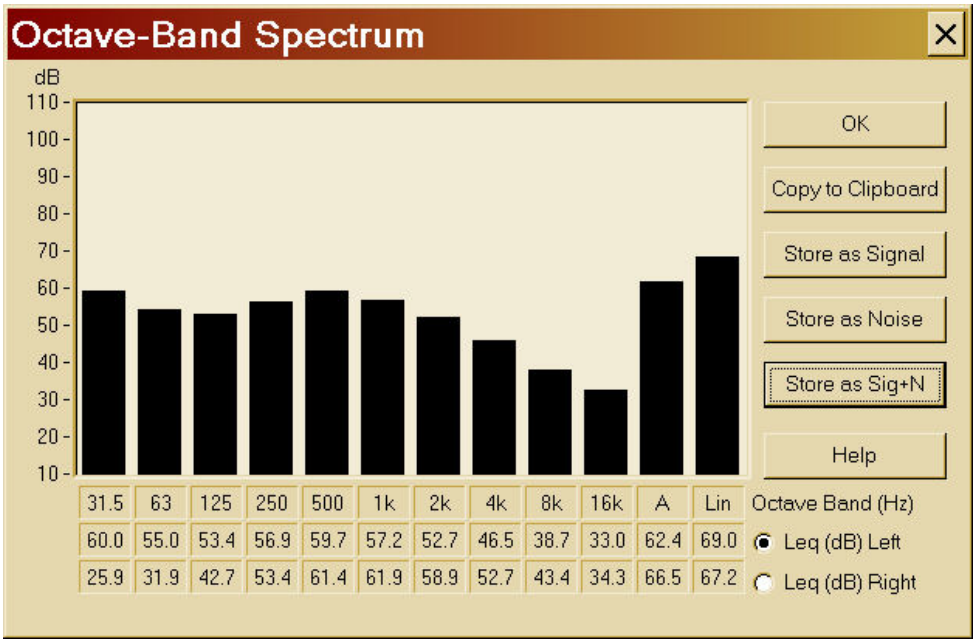
After these two recordings of background noise and of signal are done, we have to invoke again the STI module and perform an octave band analysis of them.

The first time we analyze the background noise, and we obtain this spectrum:



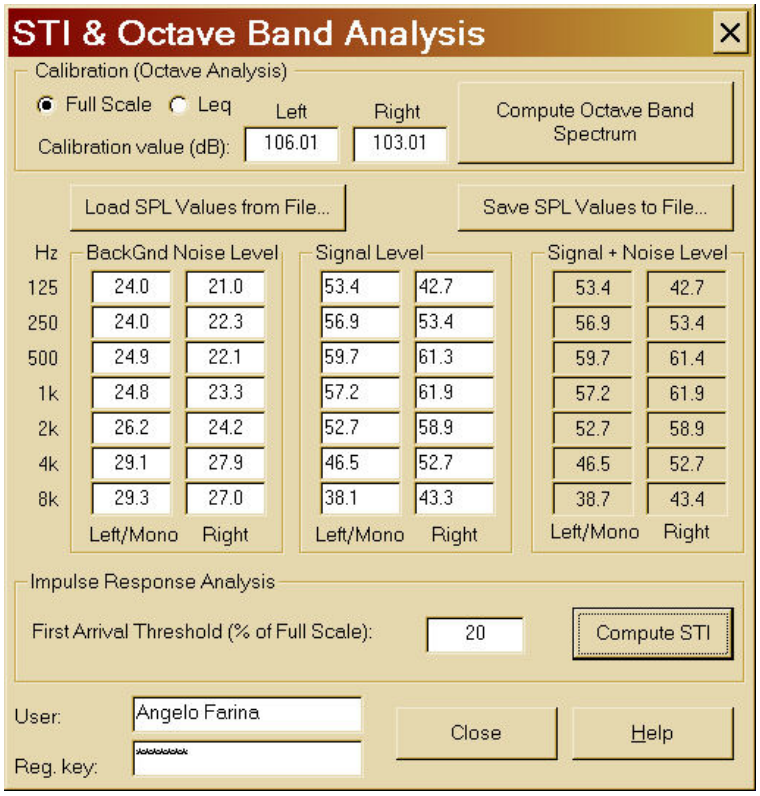
Now we store it as Noise, pressing the corresponding button.

We analyse now the recording of the Signal, and we obtain the following spectrum:



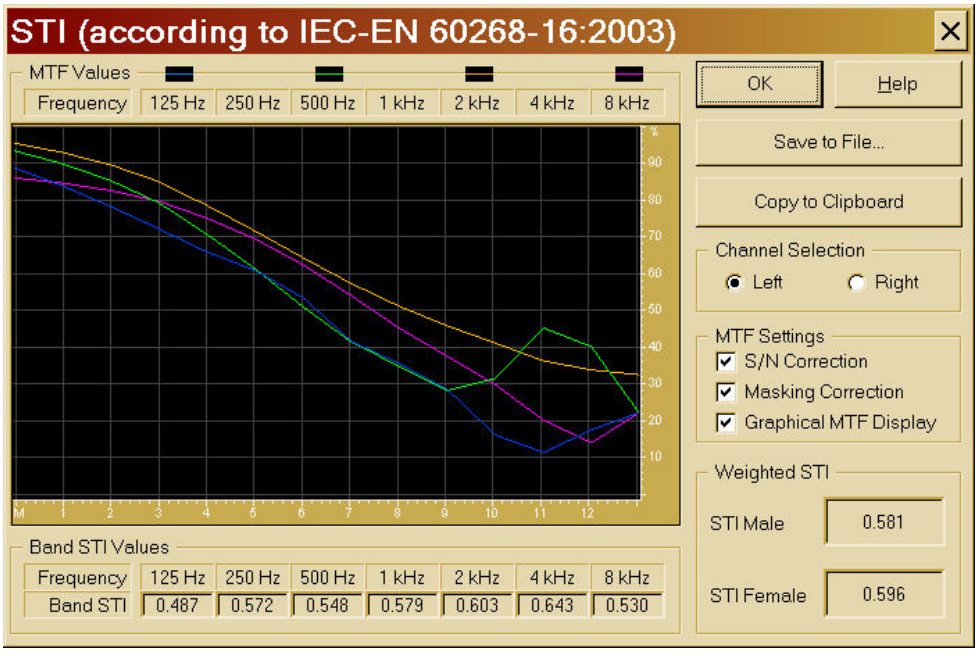
As this measurement was performed in presence of the background noise, we store this as Sig+N. In this way, the module will calculate properly the Signal, subtracting the Noise which was already estimated from the previous recording.

And now, finally, we can process the measured impulse response, for computing the STI. We open it with Audition, and we invoke the STI module:

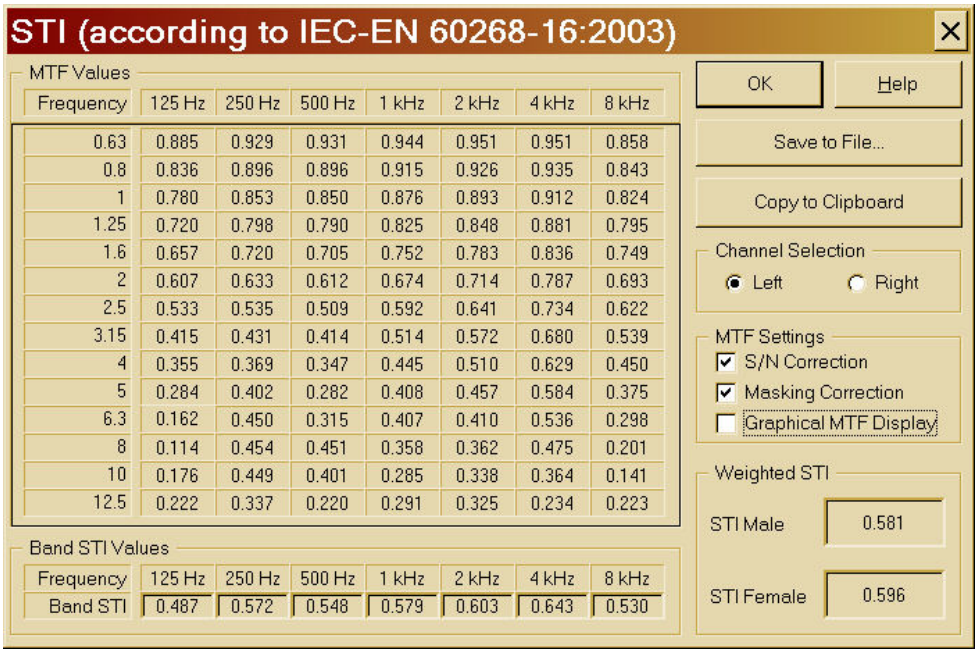


We notice that now both the Full Scale values and the spectra of Signal and Noise are properly set to the correct values.

We press now the button "Compute STI", and the following window appears:



The graphical display of 4 of the 7 octave-band MTF values can be disabled, revealing the complete table of MTF values:



Frequency	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz
0.63	0.885	0.929	0.931	0.944	0.951	0.951	0.858
0.8	0.836	0.896	0.896	0.915	0.926	0.935	0.843
1	0.780	0.853	0.850	0.876	0.893	0.912	0.824
1.25	0.720	0.798	0.790	0.825	0.848	0.881	0.795
1.6	0.657	0.720	0.705	0.752	0.783	0.836	0.749
2	0.607	0.633	0.612	0.674	0.714	0.787	0.693
2.5	0.533	0.535	0.509	0.592	0.641	0.734	0.622
3.15	0.415	0.431	0.414	0.514	0.572	0.680	0.539
4	0.355	0.369	0.347	0.445	0.510	0.629	0.450
5	0.284	0.402	0.282	0.408	0.457	0.584	0.375
6.3	0.162	0.450	0.315	0.407	0.410	0.536	0.298
8	0.114	0.454	0.451	0.358	0.362	0.475	0.201
10	0.176	0.449	0.401	0.285	0.338	0.364	0.141
12.5	0.222	0.337	0.220	0.291	0.325	0.234	0.223

Frequency	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz
Band STI	0.487	0.572	0.548	0.579	0.603	0.643	0.530

We can see how much the S/N Correction or the Masking Correction are affecting the STI values by de-selecting the correspondent options: usually the S/N is very important, whilst the masking has very little effect on the results.

It must be noted that this module also computes the STI Female, even if the measurement has been performed with the STI Male equalization.