

Software implementation of an MLS analyzer, with tools for convolution, auralization and inverse filtering

Angelo Farina, Fabrizio Righini

*Department of Industrial Engineering, University of Parma,
Via delle Scienze, I-43100 PARMA, Italy
tel. +39 521 905854 - fax 905705
E-MAIL farina@pcfarina.eng.unipr.it - [HTTP://pcfarina.eng.unipr.it](http://pcfarina.eng.unipr.it)*

ABSTRACT

The paper describes a suite of software tools, which have been developed as extensions to the most diffused shareware program for waveform editing on Win-32 PCs (Cool Edit 96 by David Johnston). For each tool a brief explanation of the underlying theory is given, along with an application example based on experimental work.

CONTENTS

These tools allow for the generation of MLS (Maximum Length Sequence) and IRS (Inverse Repeated Sequence) signal, which are used for the excitation of the system under test. Two corresponding modules make it possible to deconvolve the system's Impulse Response from the signal sampled at the output of the system.

Various processing can be done with the measured impulse responses: another module acts as a software convolver, making it possible to use the impulse responses as linear FIR filters. Thus, for example, it is possible to add reverberation, spatial impression and tonal coloration to a "dry" (anechoic) signal.

Other two modules are for the computation of inverse filters: given an impulse responses, it is possible to create another impulse response which has a frequency response which is the reciprocal of the first one, and this can be useful for perfectly equalizing a reproduction system. The second module computes an inverse filter which do not simply invert the frequency response, but also the phase response: this could be used, in principle, for dereverberating signals recorded in a reverberant space, but it can be used also for the time re-alignment of multi-way loudspeakers.

If the impulse responses refer to a room, it is possible to compute the most common acoustical parameters with a dedicated module, according to the definitions given in the ISO3382 standard.

The last two modules are for file format conversions: the first makes it possible to read and write impulse response data in the format used by the MLSSA system. The second converts energy impulse responses coming from room acoustics programs, such as Ramsete [1], in standard pressure impulse responses; also the opposite conversion is possible, for direct comparison of experimental results with computed impulse responses, making use of the graphical post-processing tools of Ramsete.

1. INTRODUCTION

In the last few years, many new instruments have been introduced for audio and acoustics measurements, based on the properties of advanced excitation signals. All these systems allow for the determination of the Impulse Response (IR) of a system (assumed linear and time-invariant), which usually includes loudspeakers, room response and microphones: the most famous are TDS (Time Delay Spectrometry), in which the excitation signal is a sinusoidal sweep, and MLS, in which the excitation signal is a periodic sequence of pseudo-random binary digits. After the measurement of the IR has been accomplished, a large number of derived quantities can be computed through proper post-processing tools, such as the system's Frequency Response, the reverberation time, the Nyquist plot, the Energy-Time-Frequency plot, and dozens of different acoustical parameters.

These instruments are usually computer-based, but require dedicated hardware for the generation of the signal and in some cases for the measurement of the system's responses: this is impractical and expensive. So the authors developed a set of "virtual instruments", which are pieces of software capable of emulating the main features of the existing audio analyzers. New possibilities have also been added, due to the extremely simple and modular approach, and other are under development now, extending the possibilities well beyond those of the hardware-based analyzers.

As host system, a standard multimedia PC, equipped with a low-cost, 16-bit, Windows-compatible sound board was used: this means that these new software modules are ready to run on more than 10 millions of computers worldwide, requiring nothing more than proper cabling to the transducers. The software environment employed is a widely diffused, ultra-low-cost waveform editor, called CoolEdit96: it can be freely downloaded from the Internet, and the registration cost is only US \$ 50. Everyone can write additional modules for CoolEdit, because the author, David Johnston, freely distributes an advanced API (Application Programming Interface). This makes it possible to create software DSP modules with a very little programming effort. The power of CoolEdit is not simply in this expandability option: this program already contains a lot of useful things, such as FFT analysis, spectrograph analysis, any kind of filtering and panning, interesting sound effects such as echo and reverb, noise removal, signal generation, and much more.

So, with a little number of new modules, CoolEdit turns to be the most powerful audio workstation around. And when a new measurement technique or a new analysis tool appears, it can be implemented with a minimum effort, maintaining the system always on the edge.

The only known problem with this completely software, modular implementation is about the analogue quality of the PC sound boards: many of them are awful, many barely acceptable, and only a very few ones have a quality comparable with instrumentation-grade acquisition boards. This is certainly a problem when the measurements have to comply with international standards or national laws, but often the sound analyzer is used for research or technological development, or only comparative results are required. Provided that a preliminary check ensures that the measurement errors caused by the sound board are below the acceptance level, almost any 16-bit sound board can be used.

The last problem is about the full-duplex capabilities of today's sound boards: although in many of them there is no hardware constraint impeding to simultaneously generate a signal and record the system's response, many software drivers do not allow for this. In these case the measurement requires a temporary storage of the excitation signal, or of the system's response, on a digital recorder (usually a DAT deck). This could cause a further degradation of the measurement chain, but it has the advantage of eliminating the need for a wired

connection of the microphone with the loudspeaker, and this is very useful when measuring room's responses in large spaces such as theatres, sport arenas, and similar.

In the following chapters each group of modules is separately explained, with a brief theoretical explanation, and a description of the use through an applicative example based on experimental work.

2. GENERATION OF THE EXCITATION SIGNALS

Two modules were developed for the generation of excitation signals: the first one produces MLS sequences of very high order and with different tap positions. Besides the possibility of using MLS signals, also IRS (Inverse Repeated Sequences) can be generated, obtaining a significant reduction of the odd-order distortion artifacts, but doubling the measurement time.

The MLS signal is well known since at least two decades [2,3,4]: it is a binary sequence, in which each value can be simply 0 or 1, obtained by a shift register as the one shown in fig. 1. The obtained signal is periodic, with period of length L given by:

$$L = 2^{N-1} \quad (1)$$

in which N is the number of slots in the shift register, also called the order of the MLS sequence. Thus an order N=16 means a sequence with a period of 65535 samples. The software MLS generator can produce sequences up to order 21, which correspond to 2,097,151 samples, or 47.55 s at a sampling rate of 44.1 kHz.

Another very important point is the position of the tap inside the shift register: it is possible to generate MLS sequences also with multiple taps, and the position of the taps influences the behavior of the sequence, particularly when it is used for the excitation of system which are not perfectly linear. The work of Vanderkooy [5] suggests that some sequences are better than others, reporting a list of known taps position which works reliably for various orders of the MLS sequence. All these are included in this software module, and the user can choose between the various available sequences for any order.

As shown by the work of Vanderkooy, the MLS excitation technique can cause the appearance of spurious peaks in the deconvolved IR, if the system under test exhibit a significant amount of distortion. Although this problem can be identified and removed by using two different MLS sequences of the same order but with different tap positions, it is also possible to use a more advanced excitation signal, called IRS (Inverse Repeated Sequence): this is made of two identical MLS sequences, being the second one an inverted copy of the first (this means replacing zeroes with ones, and vice-versa).

The use of this IRS signal should avoid, in principle, any artifact due to odd-order distortion, because they simply cancel out from the result: the even order distortion, anyway, remain unchanged, and thus this signal is useful only when the lot of the distortion is concentrated in odd-order harmonics.

For creating the excitation signal, the user has simply to go at the Generate menu of CoolEdit, and then choose either MLS signal or IRS signal. At this point a dialog box appears, for the choice of the proper sequence order and tap positions: after the choice is made, the required signal is generated, with an amplitude of +/-16384. The amplitude of the generated signal is not set at the maximum allowable, but 6 dB lower, because on many sound boards the use of the last bit of the D/A converter causes distortion in the analog part of the circuitry. Fig. 2 shows an example of an MLS signal of order 5.

After the signal has been generated, it can be played continuously placing CoolEdit in “loop mode”, and leaving it running while another instance of CoolEdit is sampling the system’s response. Obviously the signal can be modified after generation, acting with the CoolEdit Transform options, changing for example the amplitude of the signal, or applying a frequency equalization.

An alternative way of producing MLS signals, originally suggested by Mommertz [6], will be explained in the next chapter.

3. DECONVOLUTION OF THE SYSTEM’S RESPONSE

After having sampled the response of the system to the excitation signal for a reasonable time (usually at least 10 repetitions of the original sequence), this response has to be processed, for extracting the system’s IR. In any case, the number of sequences sampled are first averaged, yielding a single sequence of the proper length, with an increased signal-to-noise ratio.

Then the process is quite different depending on the kind of excitation signal, and thus two separate modules have been developed.

When the signal was MLS, the deconvolution of the IR can be made with the well known Fast Hadamard Transform (FHT), as originally suggested by Alrutz [2], and clearly explained by Ando [3] and Chu [4]. The Ando formulation was employed here. The process is very fast, because the transformation happens “in place”, and requires only addition and subtractions. The computations are done in floating point math, and the computed IR is then converted in 16-bit integer format by scaling to the maximum value. This means that the absolute amplitude information is lost.

Furthermore, being the sampling process completely asynchronous with respect to the signal generation, no “absolute zero time” exists, and the computed IR happens to be circularly folded along the time axis. So it is unfolded, placing the maximum value always at the sample number 1500.

These are without doubt limits of the software implementation in comparison with already existing hardware-based MLS systems. But the sound boards are stereo, so the two channels can be used for working around these limits: one of the input channels can be wired directly to the signal output, while the second channel captures the microphone signal. In this way, the almost perfect Dirac’s delta function deconvoluted by the first channel is placed at the sample N. 1500, with full-scale amplitude, and the system’s response appears on the other channel, with lower amplitude and a correct delay. In case of different measurements to be compared, being the excitation signal always the same, the relative amplitude and delay between the system responses are correctly maintained. This workaround sacrifices one of the two channels, but in many case the absolute amplitude and delay are meaningless, and thus both the input channels can be used for capturing microphone signals, making it possible, for example, to measure binaural impulse responses simultaneously.

The process is different with IRS signals: in this case, the very efficient FHT algorithm cannot be used, and the traditional FFT approach has to be used for deconvolving the impulse responses. This can be slow on some low-power computers when the sequence order was very high, but with today’s fast processors the deconvolution of sequences up to order 18 can be done almost instantaneously. Apart the difference in the deconvolution scheme, the only other important difference between MLS and IRS is the fact that with the latter there is uncertainty about the sign of the deconvolved IR: in fact the software cannot know if the sampling was

started during the playback of the first half of the IRS signal, with has correct polarity, or during the second half, which has inverted polarity. So it happens that the deconvoluted IRs are “upside down”, but this can be easily corrected by the CoolEdit editing functions.

Figure 3 shows a sampled signal, obtained in a room where a loudspeaker was playing continuously an MLS sequence signal. At this point, the Transform/Deconvolve MLS signal is invoked: a dialog box appears, for choosing the proper MLS order and tap position, after which the deconvolution process starts: at the end of the process, a dialog box informs the user that the computed IR has been stored in the Windows clipboard, from where they can be easily recovered through an Edit-Paste command. Figures 4 and 5 show the final dialog box and the deconvoluted IR.

The deconvolve MLS signal module can be used also for a different task: generating an MLS-like signal, but having a predefined spectral content, instead of being white. This idea was first published by Mommertz [6], and it is based on the reversibility of the Hadamard transform.

For generating an excitation signal with a pre-defined spectrum, a Dirac’s delta function, followed by a large number of zeroes, is first generated. Then it is processed, applying a frequency filter. At this point, an artificial impulse response has been created, with the wanted spectral behaviour. The steps necessary to transform it in an MLS-like signal are the following:

1. Reverse it on the time axis
2. Invoke the Deconvolve MLS Signal module, with the required order and tap position
3. Reverse again the result on the time axis
4. The obtained signal can be continuously looped for exciting the system under test
5. When the response of the system is deconvolved again, the result obtained is the convolution of the original artificial impulse response with the system’s IR

Another application of this technique is the pre-equalization of the measurement chain, including the transducers: let we assume that the loudspeaker and microphone response was first measured in anechoic conditions. An inverse filter of such a response can be created with the module described in chapter 5. This inverse filter can be used as the starting point of the above process, creating an MLS-like signal which compensates for the uneven frequency response of the transducers. When this excitation signal is employed for room acoustics measurements, the deconvoluted IR is already free of the effect of the transducers, and contains only the room-related information, as if the trasnducers were “perfect”.

3. FAST and REAL-TIME CONVOLUTION

An Impulse Response is a linear filter, and it can be applied to any kind of input signals through a linear convolution process. This principle is known since a lot of time, and the fast convolution algorithms are on Digital Signal Processing books since more than 20 years [7]. Nevertheless, such a simple algorithm is usually not found in any waveform editing program, nor it is inserted along with the post processing capabilities of the audio or acoustic analyzers.

The convolution module developed for CoolEdit can be used in two different ways: it can make a batch processing on existing waveforms, which are filtered and saved again on the hard disk. Or it can do a real-time convolution, sending directly the result to the audio board for listening, but without saving it.

The second mode is useful for listening quickly at the results, after which the process can be started again in the first mode, saving the result.

The actual implementation of the algorithm makes use of the select-save algorithm, as described in [7], as explained here

The convolution of a continuous input signal $x(t)$ with a linear filter characterized by an impulse response $h(t)$ yields an output signals $y(t)$ by the well-known convolution integral:

$$y(\tau) = x(\tau) \otimes h(\tau) = \int_0^{\infty} x(\tau - t) \cdot h(t) \cdot dt \quad (2)$$

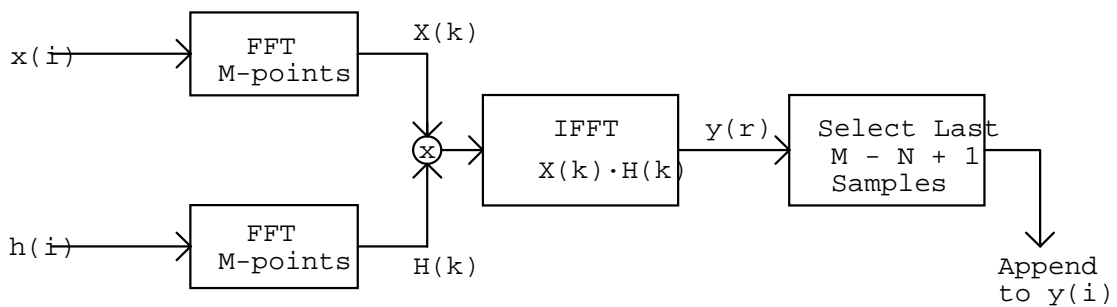
When the input signal and the impulse response are digitally sampled ($t = i \cdot Dt$) and the impulse response has finite length N , such an integral reduces to a summation:

$$y(i) = \sum_{j=0}^{N-1} x(i - j) \cdot h(j) \quad (3)$$

The sum of N products must be carried out for each sampled datum, resulting into an enormous number of multiplications and sums! These computations need to be made with float arithmetic, to avoid overflow and excessive numerical noise. For these reasons, the real time *direct true convolution* is actually restricted to impulse response lengths of a few hundredths points, while a satisfactory description of a typical concert hall transfer function requires at least $N=65536$ points (at 44.1 kHz sampling rate).

However, the convolution task can be significantly simplified performing FFTs and IFFTs, because the time-domain convolution reduces to simple multiplication, in the frequency domain, between the complex Fourier spectra of the input signal and of the impulse response. As the FFT algorithm inherently suppose the analyzed segment of signal to be periodic, a straightforward implementation of the Frequency Domain processing produces unsatisfactory results: the periodicity caused by FFTs must be removed from the output sequence.

This can be done with two algorithms, called *overlap-add* and *select-save* [7]. In this work the second one has been implemented. The following flow chart explain the process:



As the process outputs only $M-N+1$ convolved data, the input window of M points must be shifted to right over the input sequence of exactly $M+N-1$ points, before performing the convolution of the subsequent segment.

The tradeoff is that FFTs of length $M > N$ are required. Typically, a factor of 4 ($M=4 \cdot N$) gives the better efficiency to the select-save algorithm: if N is 65536 (2^{16}), one need to perform FFTs over data segments of length $M=65536 \cdot 4=262144$ points! This requires a very large memory allocation, typically 1 Mbyte, for storing the input sequence or the output signal. The overall memory requirement for the whole select-save algorithm is thus several Mbytes! This is the reason for which this algorithm cannot be easily implemented on widely

diffused low cost DSP boards, which are not equipped with such a storage capability. Only dedicate DSP units can afford this task [8].

On the other hand, today's personal computers have plenty of RAM, and enough floating-point computing power for keeping-up in real time even with very long IRs: a Pentium Pro running at 200 Mhz can convolve a mono input signal (sampled at 44.1 kHz) with a mono IR of more than 800,000 points, and with a stereo (or binaural) IR of more than 200,000 points/channel. Further improvements will be possible with the new Pentium II, and the future is pink, because the computing power is doubling each year...

The only problem with the Select-Save algorithm is that it causes a substantial delay, typically 4 times the IR length. A zero-delay algorithm has been proposed by Gardner [9], and is actually implemented (and patented) in hardware DSP convolvers [10], but at the expense of a noticeable increase in the number of computations required. For these reasons, a new algorithm is under development, which will make it possible to listen immediately at the convoluted signal as soon as the "listen" button is pressed, but maintaining the computational needs at a level only slightly larger than the original Select-Save algorithm.

Here is how the user accesses the convolution module: first the original signal (possibly anechoic) is loaded in Cooledit, from a hard disk file or a CD. At this point the Transform-Convolver option is chosen, and the dialog box shown in Figure 6 appears. Here the user can select up to two stereo IRs, loading them from the clipboards or from files. How the input channels are processed through the filters is chosen by pressing the push-buttons on the right: in the most complex case, a stereo input signal is passed through two stereo IRs, so that each input signal is convolved with two IRs, and the resulting "Left" and "Right" channels are mixed with the corresponding results of the other input channel. This is particularly useful when implementing a cross-talk cancellation system, for listening at binaural material through a pair of loudspeakers, as it will be explained at chapter 6.

The lower-right section of the dialog box allows to start the processing, and the insertion of an overall scale factor, for adjusting the playback level during real-time listening (this control can be accessed also during the playback); the overall scale factor is disabled during batch processing, because in this case the result is always normalized to the maximum value before conversion to the integer format.

Being the entire process made with floating-point math, the computed result has a sonic quality and a dynamic range far better than those achievable with fixed-point DSP units.

4. CREATION OF A FREQUENCYRESPONSE EQUALISING FILTER

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.

The algorithm employed is the one proposed by Neely and Allen [11]: the FFT of the original impulse response is taken, then the frequency response is properly smoothed, and the phase of the complex spectrum is thrown away. At this point an IFFT yields the required inverse filter.

It must be noted, anyway, that the non-minimum phase component is left not-inverted: as this component carries the reverberation, the equalized signal has a proper spectral frequency response (depending on the degree of smoothing applied), but it still suffers of echoes or reverberation present in the original signal. Thus this module is ideal for creating an

equalizing filter taking into account only the direct wave coming from a loudspeaker with poor frequency response.

For example, let us equalize the anechoic IR of a loudspeaker, which has been measured with the MLS signal generator and deconvolver modules: it is shown in Figure 7 both as time signature and frequency response magnitude.

Now we can press the Flatten Spectrum button, and the Dialog Box shown in figure 8 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 more accurate is the frequency equalization. Then we have the Windowing function to be used before the FFT is applied to the original data.

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 Inverse Filter is stored in the Windows Clipboard. Opening another instance of CoolEdit, and pasting from the Windows Clipboard, the signal is recovered, as shown in figure 9 both in time and frequency domains.

The latter is roughly the mirror image of the original Impulse Response, and thus the recovered signal is the required equalizing filter.

For verification, if we apply it by convolution to the original signal we obtain the signal shown in figure 10: this is not perfectly flat, due to the limited number of taps employed for the creation of the Inverse Filter.

5. CREATION OF A TRUE 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 solutions are available for this problem. This argument was deeply investigated by Mourjopoulos [12], particularly with the goal of obtaining a digital equalisation 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 $\{k\}$.

$$[R] \cdot \{g\} = \{k\} \quad (4)$$

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 Toeplitz 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 be 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, fig. 11 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 colouration, as shown in the same fig. 11.

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). Morjopoulos 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.

Fig. 12 reports the inverse filter of the room response shown in fig. 11. The verification of the proper working of the filter can be done by convolving the original impulse response with it. This is shown in fig. 13: 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. Morjopoulos already addressed this problem [12], and an enhanced dereverberation module is being developed now, following his principles.

6. CROSS-TALK CANCELLATION

When one listens through loudspeakers at signals convolved with Binaural Impulse Responses (BIRs), the Cross-talk effect cannot be avoided: the signal emitted from the left loudspeaker reaches also the right ear, and vice-versa.

Anyway, a special cross-talk cancelling filters, which must be applied to the original BIRs prior of the convolution with the anechoic signal(s), can be designed.

In the following a single channel of anechoic signal is considered, which has to be convolved with a binaural impulse response and played through two loudspeakers; in the case of a stereo anechoic signal, the same processing has to be done simultaneously on the other channel, and the feed of each loudspeaker is obtained simply summing the results of the two separate processes.

Fig. 14 shows a scheme of the process: instead of the original binaural impulse responses (h_l and h_r), the cross-talk cancelling modules produces two alternative impulse responses (h_l' and h_r'), which include the inverse filters for obtaining the cross-talk cancellation effect. These filters correct also for the fact that the Head Related Transfer Function is already inserted in the IRs used for convolution, but the signal coming from loudspeaker newly interferes with the listener's head, so the head filtering is applied twice!

The construction of the new filters requires that the listening system impulse responses (h_{ll} , h_{lr} , h_{rl} , h_{rr}) have been already measured, possibly making use of the same dummy head employed for the binaural measurement in the concert hall. In mathematical notation, the signals arriving at the Left and Right ear channel entrance of the listener's head can be described as:

$$\begin{aligned} p_l &= y_l \otimes h_{ll} + y_r \otimes h_{rl} \\ p_r &= y_r \otimes h_{rr} + y_l \otimes h_{lr} \end{aligned} \quad (5)$$

Noting that the stereo signal y_l and y_r were derived from a single mono input x , through convolution with the two binaural IRs h_l and h_r , and passing to the frequency domain by FFT, the previous relations can be rewritten as:

$$\begin{aligned} P_L &= X \cdot (H_L \cdot H_{LL} + H_R \cdot H_{RL}) \\ P_R &= X \cdot (H_R \cdot H_{RR} + H_L \cdot H_{LR}) \end{aligned} \quad (6)$$

Now, let us substitute two adequate IRs (h_l' and h_r') instead of the original ones: they must be capable to make the terms in parenthesis equal to the (wanted) IRs H_L and H_R . After a few passages, we get:

$$\begin{aligned} H_L' &= \frac{H_{RR} \cdot H_L - H_{RL} \cdot H_R}{H_{LL} \cdot H_{RR} - H_{LR} \cdot H_{RL}} \\ H_R' &= \frac{H_{LL} \cdot H_R - H_{LR} \cdot H_L}{H_{LL} \cdot H_{RR} - H_{LR} \cdot H_{RL}} \end{aligned} \quad (7)$$

These simple relations permit to evaluate the wanted IRs, by converting them in their equivalent time-domain formulation.

Obviously, when the above formulas are converted in time domain, it means that the numerator is simply the sum of two convolutions, and the result has to be convolved with the inverse filter, obtained applying the previously described module to the denominator. It must

be noted that the denominator is always the same, also, for example, if a second anechoic channel has to be convolved with separate binaural impulse responses: it depends only on the listening system.

The inversion of the denominator presents, in general, the same problems already depicted at the previous chapter. In some cases, it is better to invert it simply in terms of frequency response, leaving untouched the all-pass component of it (which carries the reverberation). In fact, the removal of reverberation is usually difficult, and it can easily produce audible artifacts, whilst the simple frequency equalization is fast, reliable and robust. If the listening environment is significantly less reverberant than the original space in which the binaural impulse responses to be convolved were measured, then the latter almost completely “cover” the small reverberation of it.

One need to measure the four IRs related to loudspeakers and receiving head, whose effect is so completely removed from the audio playback. This means that not only the crosstalk is eliminated, but also unwanted frequency filtering due to loudspeaker response. Furthermore, the head-related transfer function is eliminated too, avoiding the double-filtering stated above. If the head used for the cross-talk cancellation is the same used for the measurement of the original impulse responses, then its effect is almost completely suppressed, and the listener hears “with his ears”.

All the IRs required for cross-talk compensation can be easily measured by employing the software MLS generation and deconvolution, as explained at chapters 2 and 3.

A special software module, called Subtract Convolved has been developed: its goal is simply to convolve a single channel of the input waveform with the 2 channels of a stereo IR, and subtracting the results. This is useful for computing both the two numerators and the denominator of the above formulas.

After the processing, the signal corresponding to the denominator must be inverted through the Inverse Filter module, and saved. This inverse filter must then be convolved with the two numerators, for obtaining the required time-domain, cross-cancelling filters. A new, faster module, which includes all the above operations, is under development now.

For checking the proper working of the Cross-Talk cancellation, binaural measurements was made in a listening room in which a stereo reproduction system was installed. In this way, 2 separate binaural impulse responses were obtained, as shown in fig. 15.

A particular choice of original the original impulse responses h_l and h_r was made: a dirac's delta function as h_l , and zero as h_r . This means that the original signal, being convolved through the cross-talk cancelling filters, and played on the stereo loudspeaker system, should be received only at the left ear, and the right ear should remain muted.

The cross-talk cancelling filters, computed following eq. 7, are reported in fig.16. In this case, the complete inversion of the denominator was employed.

Convolving the cross-talk cancelling filters with the 4 impulse responses of the listening room, the global transfer functions of the equalized system are found: they are reported in fig. 17, and it can be observed that the left channel contains almost perfectly a Dirac's delta function, while the right channel is zero. Also a subjective test in these conditions confirmed the effect: the sound was heard only through the left ear, and it seemed that a plug was inserted in the right one!

7. ACOUSTICAL PARAMETERS

When a room impulse response has been measured, it is possible to compute from it some important acoustical parameters, such as the reverberation time. The derivation of this and other parameters is explained by the international standard ISO 3382.

The reverberation times can be computed at various frequency (usually in octave bands), and taking into account various ranges of the decay curve.

The decay curve is built by the well-known backward integration algorithm, originally introduced by M.Schroeder [13], applied to octave-band filtered versions of the original impulse response. A noise removal optional utility as been added, based on the method suggested on the ISO3382 standard.

At this point, 4 different ranges can be taken into account, for computing the following reverberation times:

- EDT (from 0 dB to -10 dB)
- T10 (from -5 dB to -15 dB)
- T20 (from -5 dB to -25 dB)
- RTuser (the user can choose the start and end point, in dB)

Furthermore, in each octave band, starting from the squared impulse response (which is a sort of energy impulse response) also these other acoustical parameters are computed:

- C50 – Clarity on 50 ms – it is the ratio, in dB, between the “useful energy” coming from the first 50 ms of the impulse response (starting from the direct wave) and the “detrimental energy”, which is the rest of the impulse response
- C80 – Clarity on 80 ms – as above, but the useful energy interval is now 80 ms.
- D50 – Definition (%) – It is the ratio between the useful energy (first 50 ms) and the whole energy of the impulse response.
- Ts – Center time – is the first order momentum of the energy on the time axis.

For more accurate definitions, the reader is referred to the ISO3382 standard.

Let we assume that a room’s impulse response has already been measured, for example the one reported in fig. 5. If the user invokes the Acoustical Parameters module, a dialog box is first displayed, asking for the threshold for locating the direct wave, for the dB-range of the RTuser, and for the noise removal feature to be enabled or not, as it shown in fig. 18. After this, a form is displayed, which shows the results of the computation, as shown in fig. 19. The table can be copied on the clipboard, and in this case the following is obtained:

OCTAVE BAND ACOUSTICAL PARAMETERS

Left Channel

Band	Lin	31.5	63	125	250	500	1K	2K	4K	8K	16K
Parameters											
C50 [dB]	-0.1349	-28.09	-29.91	-23.58	-10.6	-9.256	-5.846	0.6492	1.312	5.547	13.67
C80 [dB]	2.212	-21.17	-19.24	-13.48	-1.766	1.654	-0.6134	2.959	2.137	6.147	13.83
D50 [%]	49.22	0.1549	0.1021	0.4364	8.01	10.61	20.65	53.73	57.49	78.2	95.88
TS [ms]	89.58	303.6	176.	153.4	139.2	122.3	149.9	82.36	76.54	41.74	13.6
EDT [s]	1.602	5.882	2.38	1.831	1.759	2.618	2.378	1.499	1.489	1.274	0.8046
RT20 [s]	1.518	3.006	3.439	2.038	1.714	1.74	1.583	1.572	1.419	0.9592	2.203
r RT20	-0.9975	-0.7664	-0.9618	-0.9815	-0.9884	-0.9819	-0.9834	-0.9949	-0.984	-0.9922	-0.9435
RT30 [s]	1.507	--	--	1.828	1.674	1.644	1.498	1.483	1.379	1.334	2.618
r RT30	-0.998	--	--	-0.982	-0.9921	-0.9827	-0.9865	-0.9971	-0.9902	-0.9763	-0.966
RTU [s]	1.689	5.603	4.225	2.091	1.913	2.28	1.791	1.797	1.044	1.152	0.6691
r RTU	-0.9982	-0.9377	-0.9694	-0.8953	-0.9683	-0.959	-0.9847	-0.9975	-0.9935	-0.9719	-0.9237
Noise C.:	yes	no	no	yes	yes	yes	yes	yes	yes	yes	yes
RTU = RT User (-5. dB, -15. dB)											

Obviously these data can be pasted into a spreadsheet for graphical output or further elaborations.

1. IMPORT/EXPORT FILTERS

The last two modules are for importing/exporting the impulse responses in other file formats. The first module, in particular, is for reading or writing waveform data in the MLSSA .TIM format.

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 an 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 a DOS-based applications, it is still now an unvaluable tool for making computations and extracting information from Impulse Responses.

The MLSSA .TIM file format is quite strange: the inport/export filter for CoolEdit makes it possible to manage .TIM files exactly as if they were .WAV files: this can be useful for importing in CoolEdit Impulse Responses measured with the MLSSA system, or, on the other hand, for exporting to MLSSA Impulse Responses measured with CoolEdit

The other module converts energy impulse responses, coming from the room acoustics simulation program Ramsete, in pressure impulse response very similar to the experimentally measured ones.

Ramsete (as many other Ray Tracing or Beam Tracing programs) produces energy impulse responses of unlimited length, sampled at intervals typically of 10, 5 or 1 ms; a separate response is computed for each octave band (31.5 to 16000 Hz). In this way, the response in one receiving point is a two-dimensional array, with 10 columns (the frequency bands) and some hundreds or thousands of rows (the time intervals).

It is necessary to convert this array in a single pressure response, with a bandwidth of 20 Hz to 20 kHz, sampled at 44.1 or (better) 48 kHz. Usually these impulse responses are stored in standard WAV format, or in proprietary Impulse Response formats as the TIM format of the MLSSA system.

The conversion process begin with the generation of a sample of white noise with proper length and sampling rate, covering the impulse response duration. Ten copies of this white noise signal are made, and each of them is "modulated" with an amplitude envelope obtained by the square root of the energy impulse response in the corresponding frequency band. After this, each modulated white signal is passed through a 6-pole IIR digital octave filter, centered on the corresponding frequency. The ten filtered signals are then summed together, obtaining a wide-band pressure response. Eventually, the result is normalized and converted to 16-bit integer.

Although this process can be criticized in many ways, it is very fast and produces impulse responses that exhibit the same energy/time curves, the same reverberation times, and almost exactly the same objective acoustical parameters as the original energy responses. Obviously this is not enough to be accepted as realistic for audible simulations: a subjective evaluation needed to be made for this, and it gave good results [1].

Fig. 20 shows the user's interface of the Ramsete->WAV conversion module, which includes also a sort of graphic equalizer, for adapting the frequency response of the sound source.

2. CONCLUSION

Joining the speed, ease of use and power of the original waveform editor with the advanced capabilities of the plug-in modules, a standard multimedia PC is transformed into a versatile audio analyzer for electroacoustic measurements and for psycho-acoustics auralization tests.

Some applications of this system are being presented in other papers [14,15], but the possibilities are so wide that it is impossible here to suggest all the fields in which the new software tools could be used. Apart the powerful computational tasks which can be solved, the beauty of the system is in the capability of listening after each manipulation at what happened to the signal: this is important for avoiding logical mistakes, and by the same time it is incredibly useful in understanding the physical meaning of each mathematical modification of the waveform.

The continuously increasing computing power of PCs suggests that in a couple of years all the heavy digital signal processing tasks, which till today required expensive, dedicated hardware solutions, will be solved with software tools. The modular approach launched by CoolEdit, and followed in this work, will be probably supported by many other software developers, and a multitude of new modules is expected in the next months.

In this way it will be possible to implement the most advanced measuring and analysis techniques, with a minimum effort and with consistency of the user's interface; new modules are under development now, for real-time FFT analysis, for underwater acoustics (profiling of the sea bottom) and for real-time deconvolution of the impulse response from the incoming signal.

3. References

- [1] A. Farina, "Auralization software for the evaluation of a pyramid tracing code: results of subjective listening tests" - ICA95 (International Conference on Acoustics), Trondheim (Norway) 26-30 June 1995.
- [2] H. Alrutz, M.R. Schroeder, "A fast Hadamard transform method for evaluation of measurements using pseudorandom test signals" Proc. of 11th International Congress on Acoustics, Paris, 6, pp. 235-238.
- [3] Y. Ando, *Concert Hall Acoustics*, Springer-Verlag, Berlin 1985, appendix
- [4] W.T. Chu, "Impulse response and Reverberation Decay measurements made by using a periodic pseudorandom sequence", *Applied Acoustics*, vol. 29, pp. 193-205 (1990).
- [5] J. Vanderkooy, "Aspects of MLS measuring systems", *JAES* vol. 42, n. 4, 1994 April, pp. 219-231.
- [6] E. Mommertz, S. Muller, "Applying the inverse fast Hadamard transform to improve MLS-measurements", Proc. of ICA95 (International Conference on Acoustics), Trondheim (Norway) 26-30 June 1995, vol. IV, pp. 127-130.
- [7] Oppenheim A.V., Schafer R., *Digital Signal Processing* - Prentice Hall, Englewood Cliffs, NJ 1975, p. 242
- [8] Connolly B. - "A User Guide for the Lake FDP-1 plus" - Lake DSP Pty. Ltd, Maroubra (Australia) - Sept. 1992.
- [9] W.G. Gardner – "Efficient convolution without input-output delay", *JAES* vol. 43, n. 3, 1995 March, pp. 127-136.
- [10] A. Really, D.S. McGrath, "Convolution processing for realistic reverberation", Pre-Prints of the 98th AES Convention, 1995 February 25-28, Paris
- [11] S.T. Neely, J.B. Allen, "Invertibility of a room impulse response", *J.Ac.Soc.Am.*, vol.66, pp. 165-169 (1979).
- [12] J.N. Mourjopoulos, "Digital Equalization of Room Acoustics", *JAES* vol. 42, n. 11, 1994 November, pp. 884-900.
- [13] M. Schroeder, "New method of measuring reverberation time", *J. Ac. Soc. Am.*, vol. 37, pp. 409-412, 1965.
- [14] A. Farina, E. Ugolotti, "Use of digital inverse filtering techniques for improving car audio systems", Pre-prints of the 103rd AES Convention, New York, 26-29 September 1997.
- [15] A. Farina, E. Ugolotti, "Subjective comparison of different car audio systems by the auralization technique", Pre-prints of the 103rd AES Convention, New York, 26-29 September 1997.

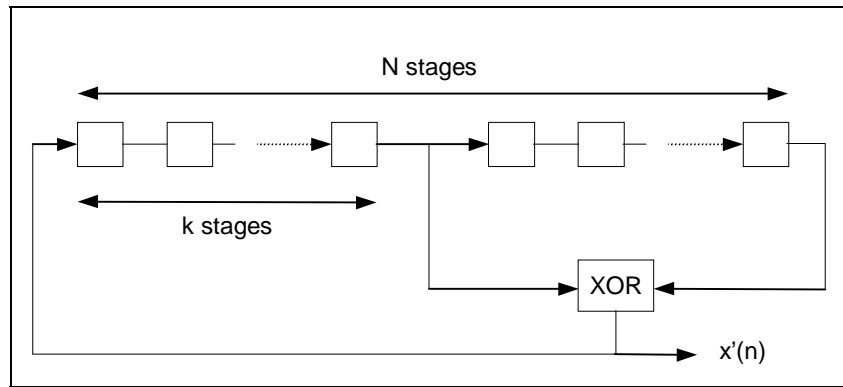


Fig. 1 – Shift register for the creation of the MLS signal

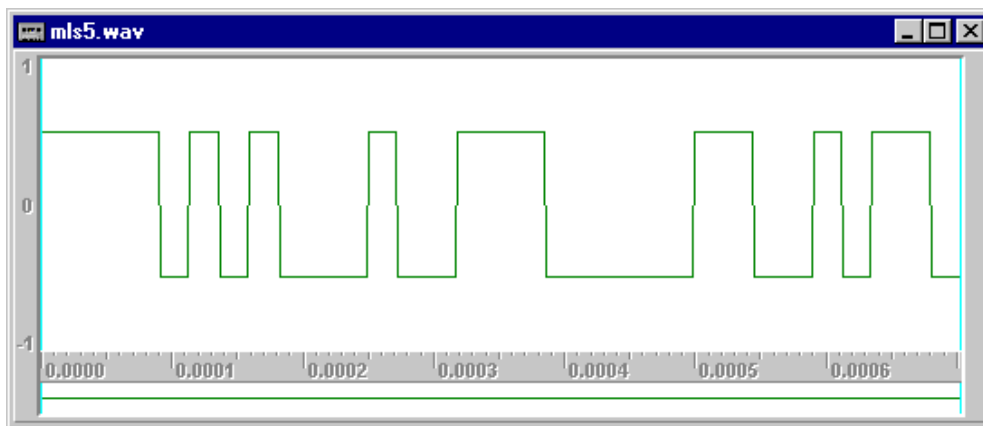


Fig. 2 – MLS sequence of order 5

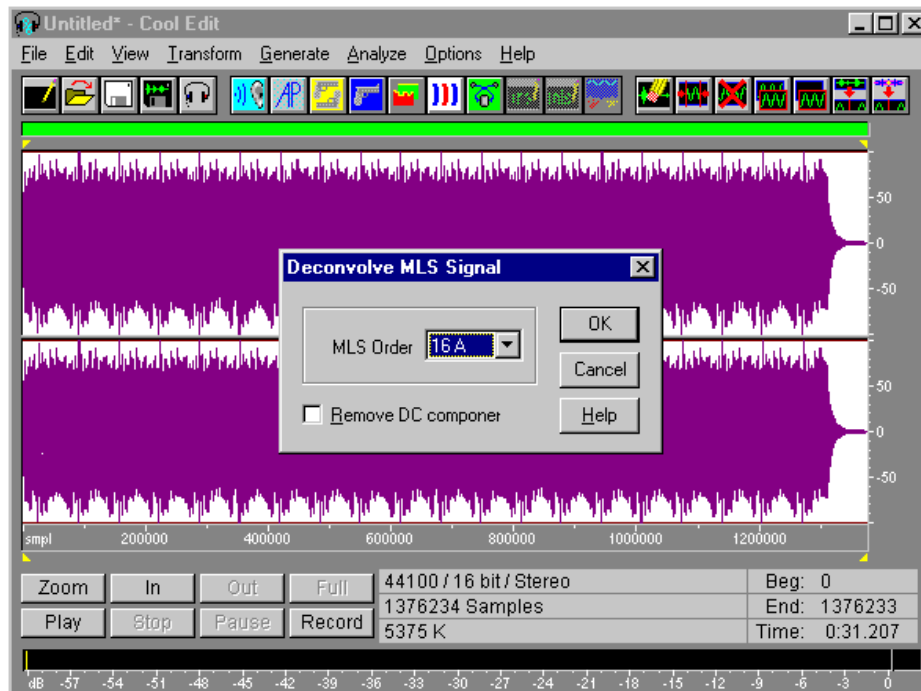


Fig. 3 – Deconvolution of the system's impulse response from the sampled signal

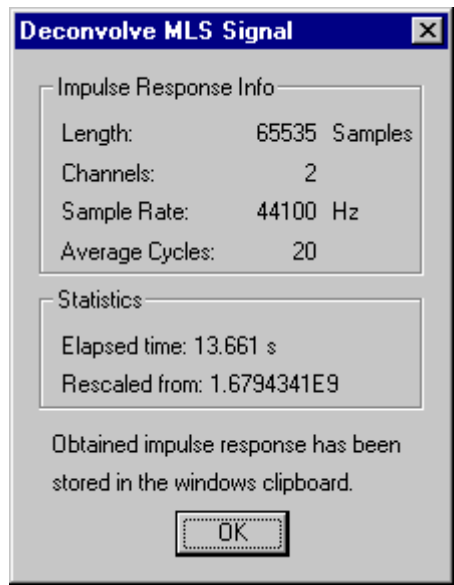


Fig. 4 – Dialog box at the end of the deconvolution

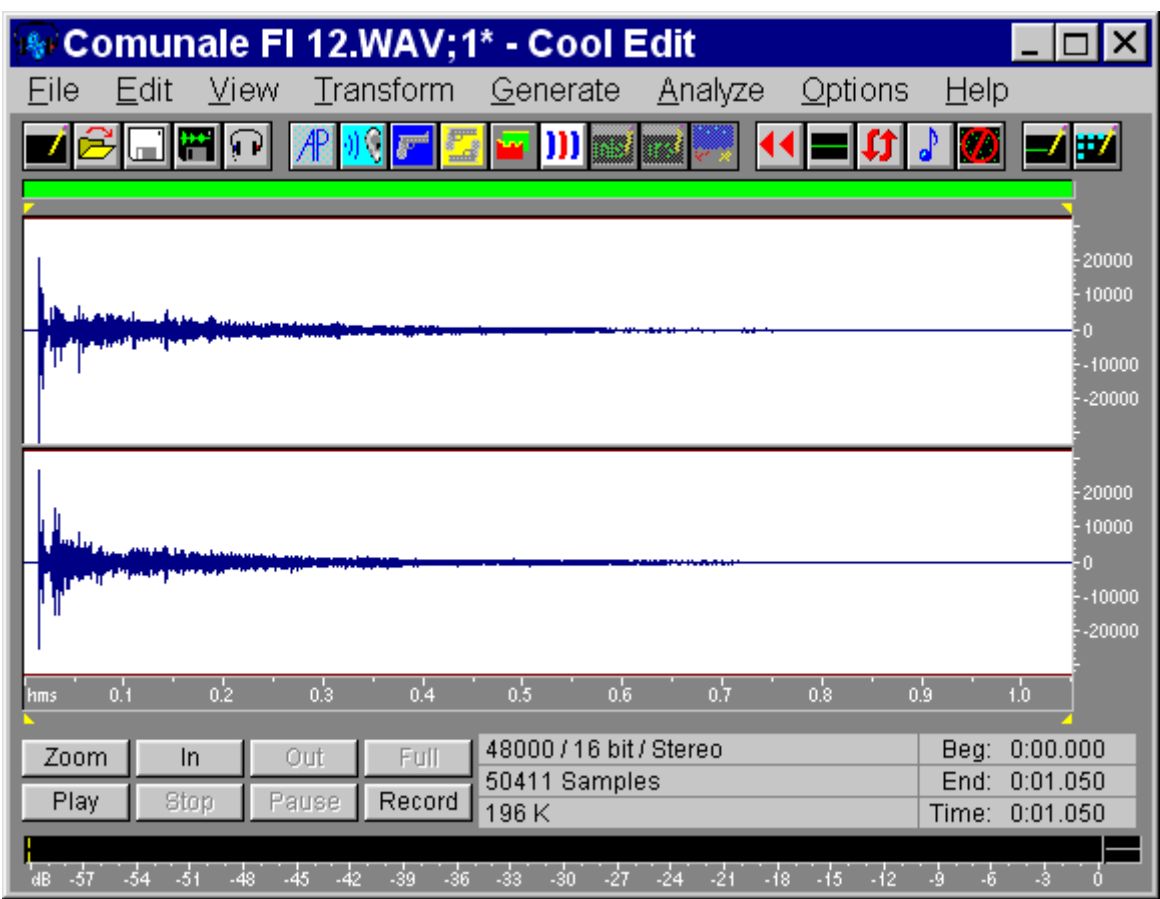


Fig. 5 – deconvolved binaural impulse response

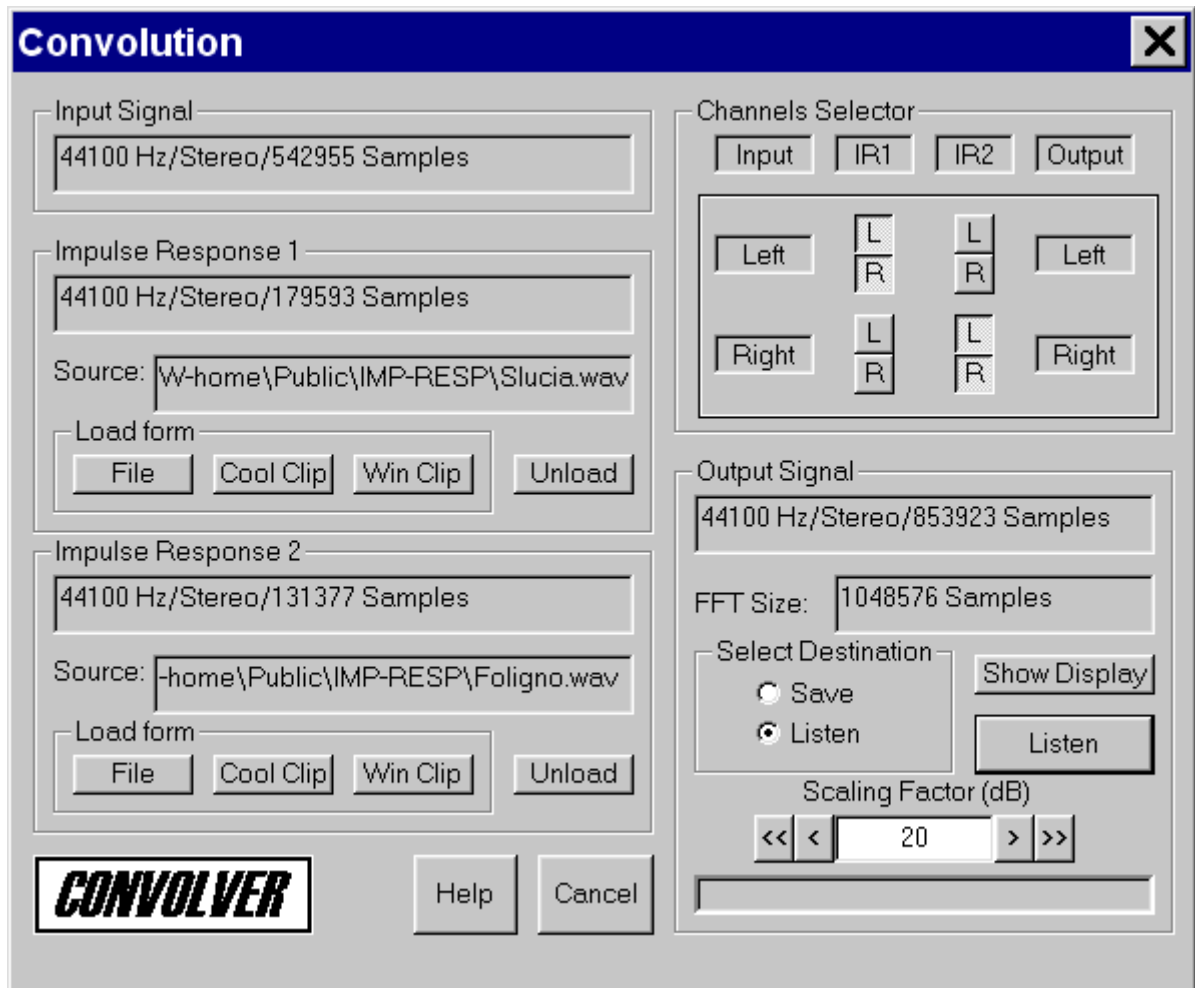


Fig. 6 – user's interface of the real-time convolution module

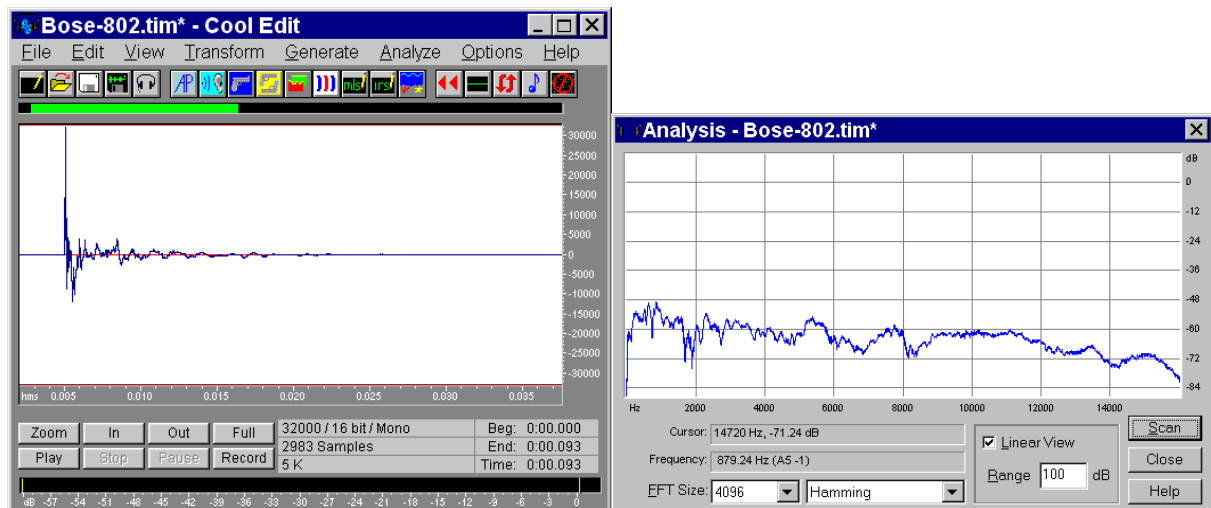


Fig. 7 – Impulse response and frequency response of a loudspeaker

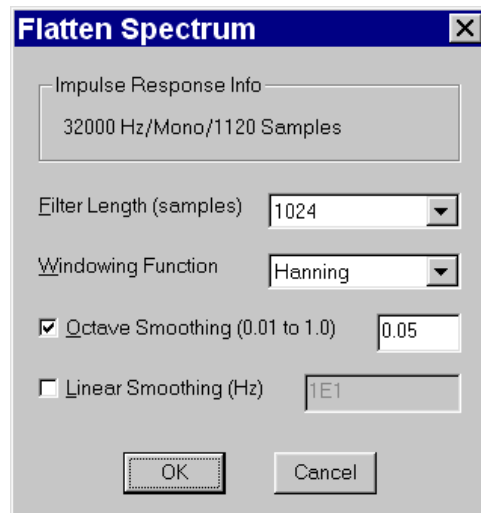


Fig. 8 – Input parameters for the “Flatten Spectrum” module

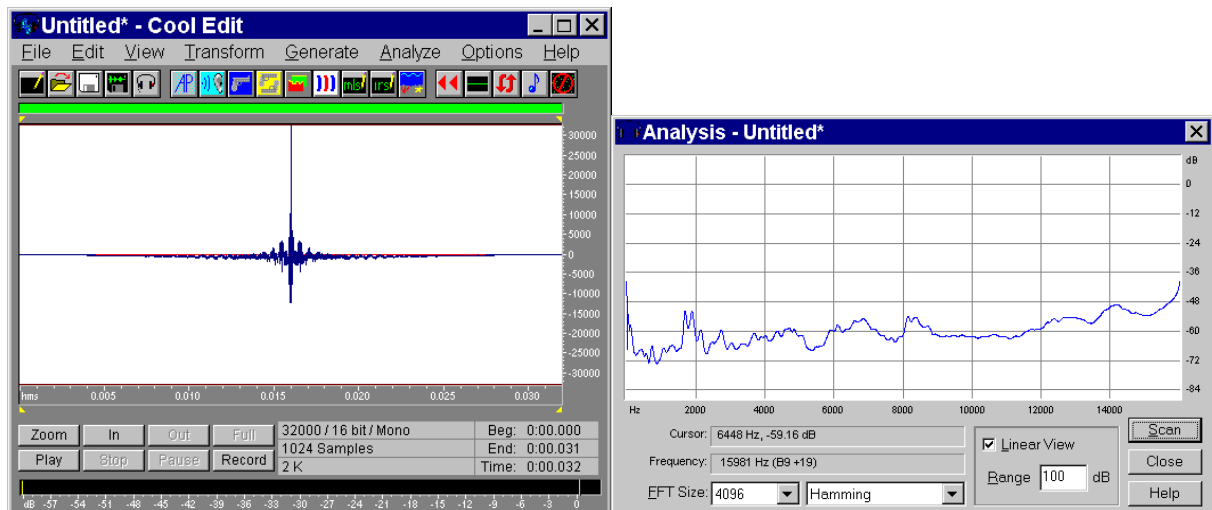


Fig. 9 – Impulse response and frequency response of the equalizing filter

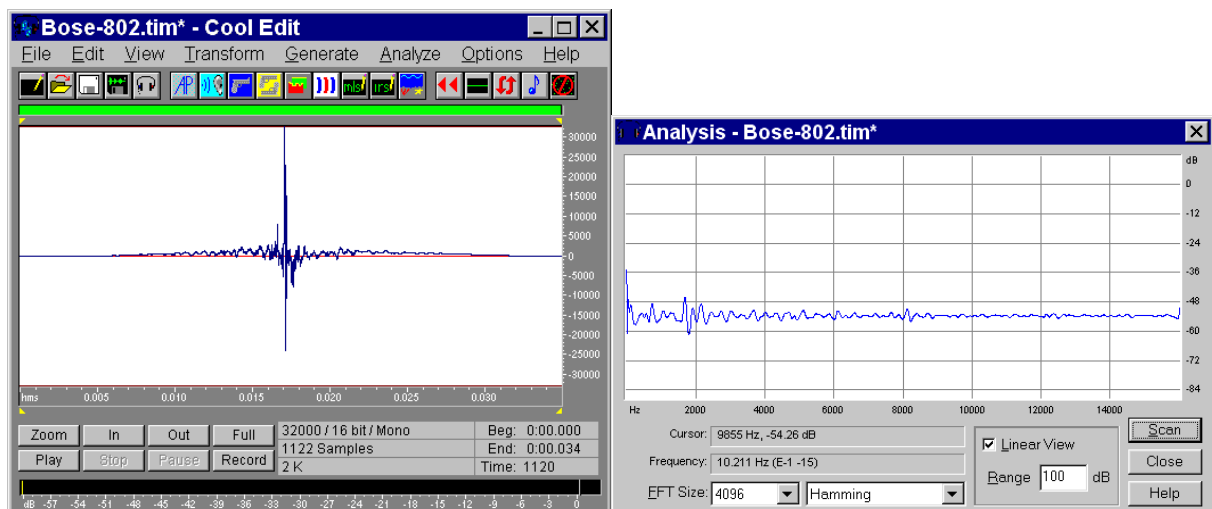


Fig. 10 – Effect of the equalizing filter on the loudspeaker’s response

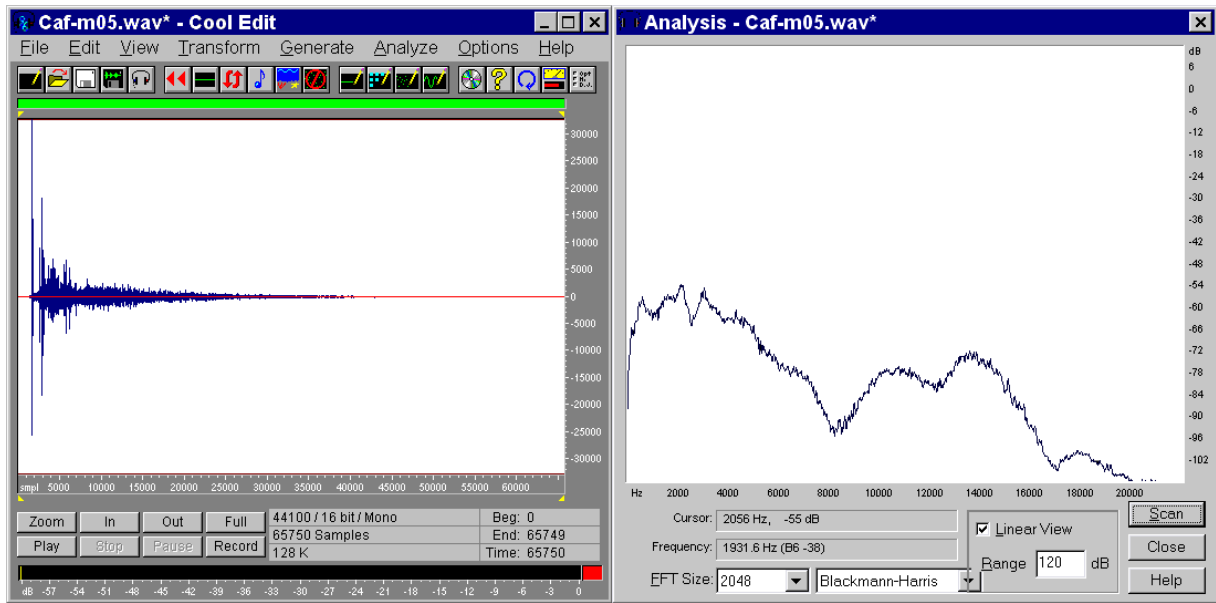


Fig. 11 – Impulse response of a room

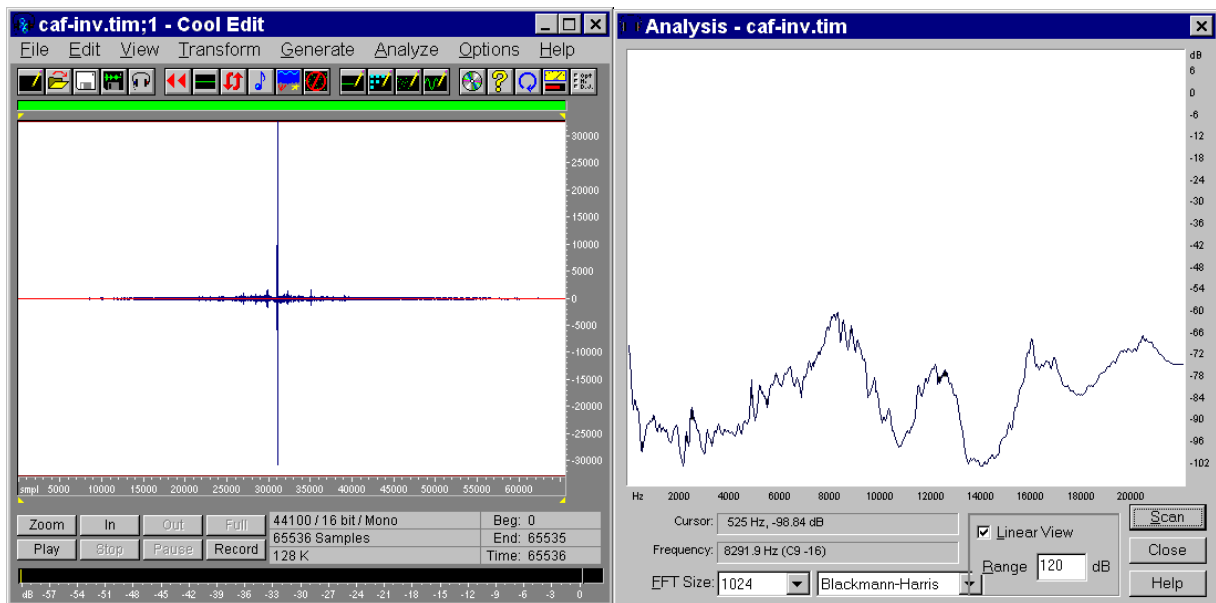


Fig. 12 – Inverse filter computed with the least-squares technique

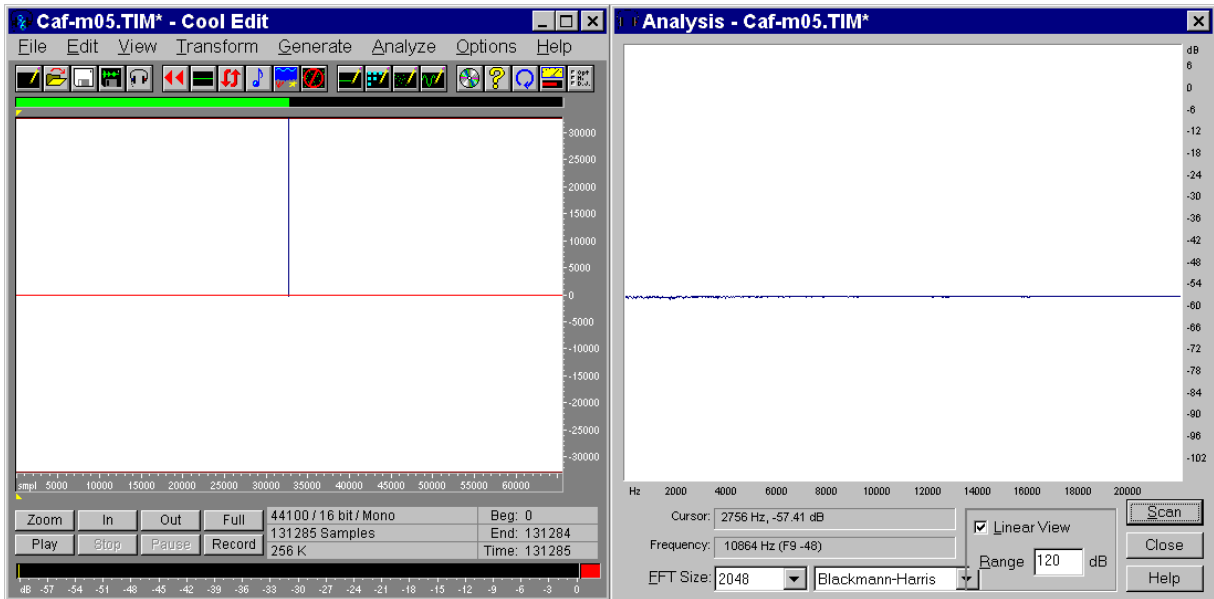


Fig. 13 – Application of the inverse filter to the original room's response

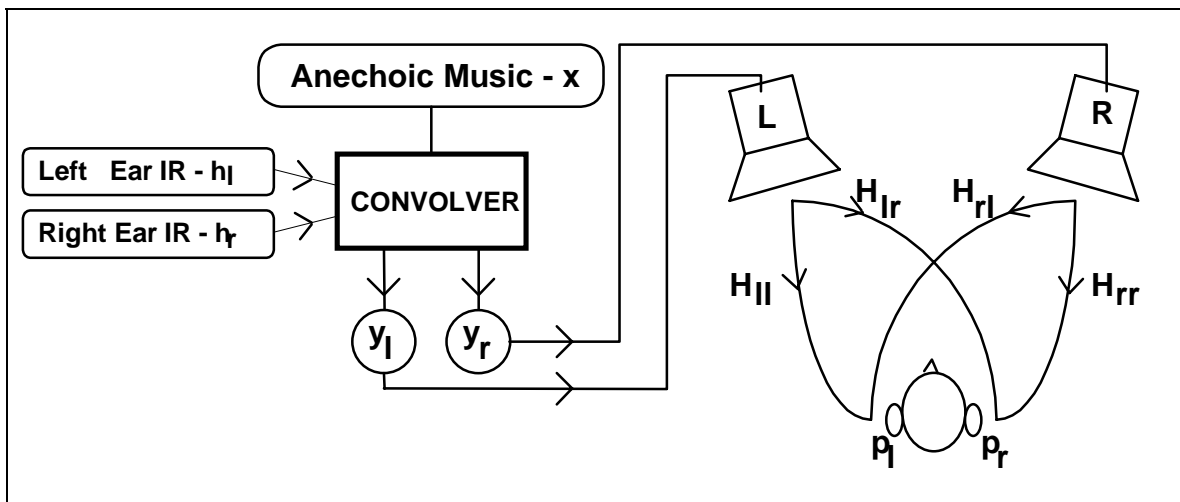


Fig. 14 – Schematic of the cross-talk effect with loudspeakers

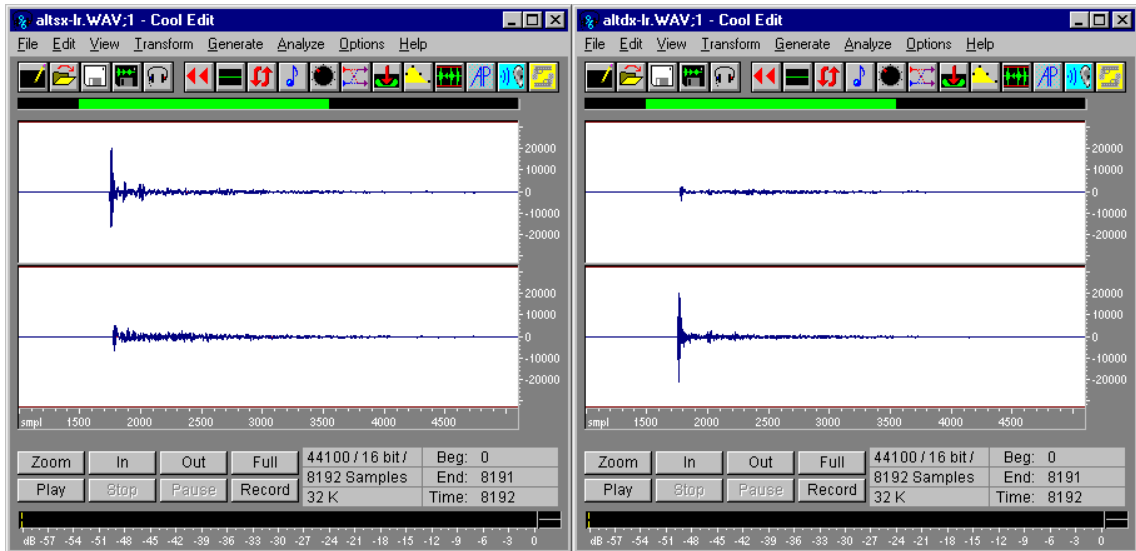


Fig. 15 – Binaural impulse responses of a stereo listening system

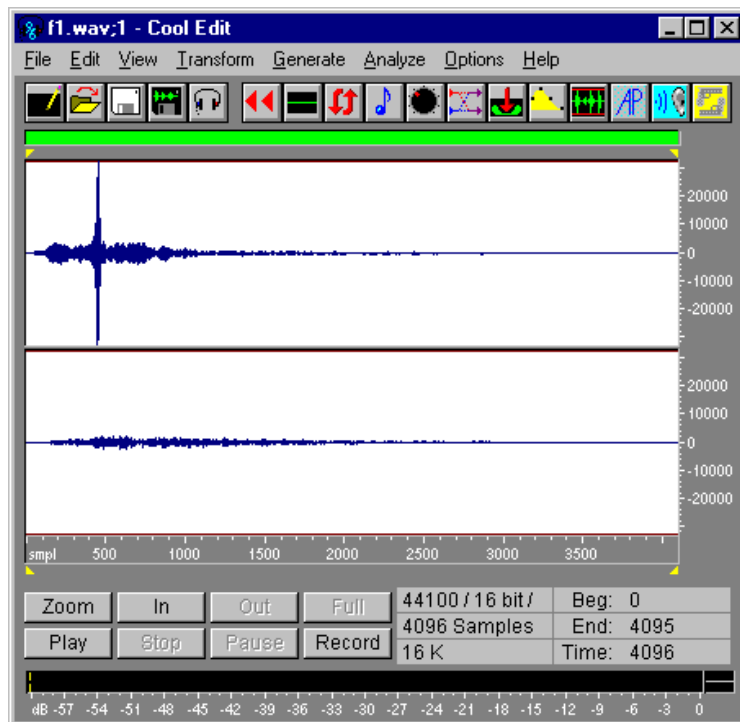


Fig. 16 – Cross-talk cancelling filters

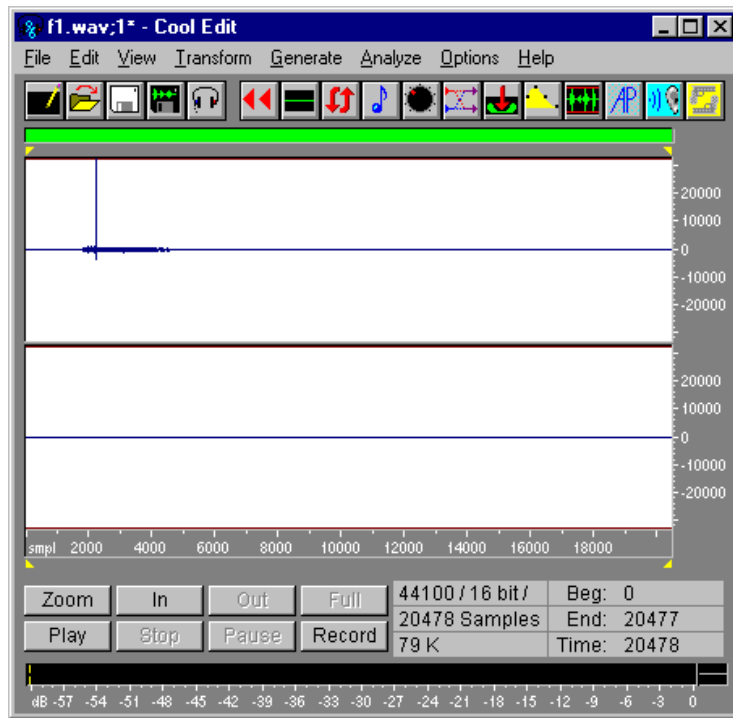


Fig . 17 – system’s response with cross-talk filters inserted

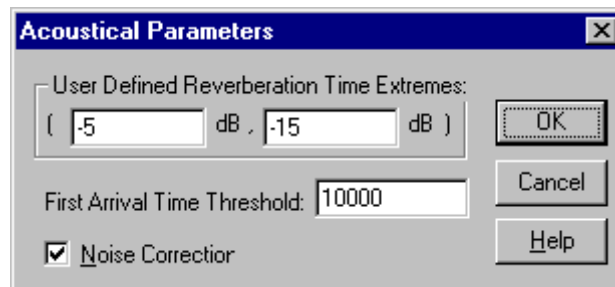


Fig. 18 – Input selection for the Acoustical Parameters module

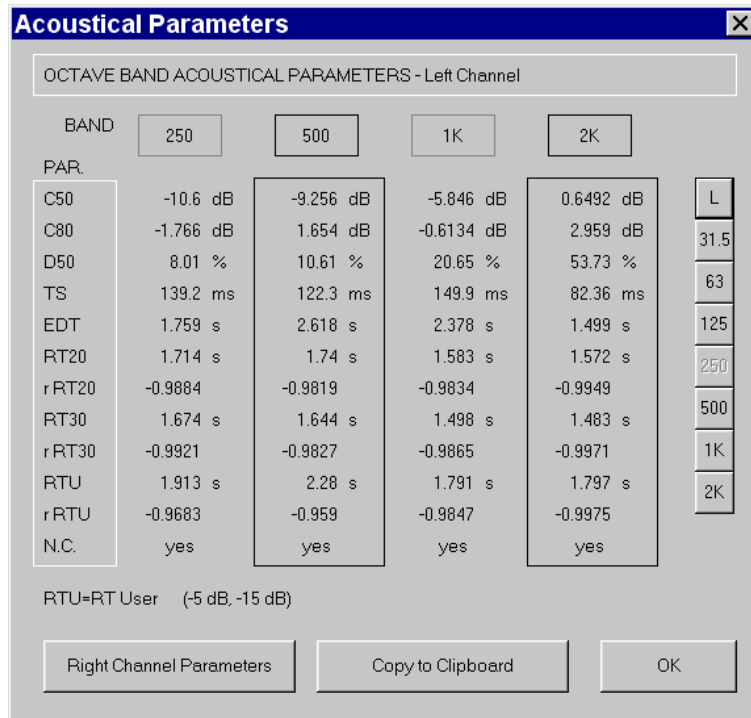


Fig. 19 – Acoustical Parameters form

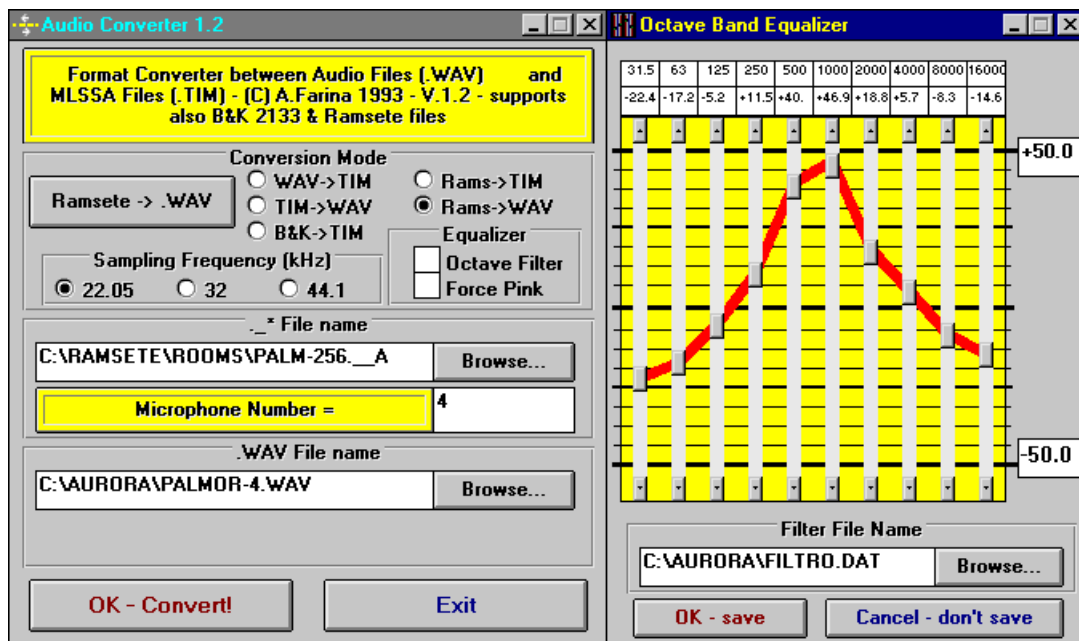


Fig. 20 – user's interface of the conversion module from Ramsete to WAV.