



Audio Engineering Society Convention Paper

Presented at the 123rd Convention
2007 October 5–8 New York, NY, USA

The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Realtime auralization employing a not-linear, not-time-invariant convolver

Angelo Farina¹, Adriano Farina²

¹ University of Parma, Industrial Eng. Dept., Parco Area delle Scienze 181/A, 43100 PARMA, ITALY
farina@unipr.it

² Liceo Ginnasio Statale G.D.Romagiosi, via Maria Luigia 1, Parma, Italy

ABSTRACT

The paper reports the first results of listening tests performed with a new software tool, capable of not-linear convolution (employing the Diagonal Volterra Kernel approach) and of time-variation (employing efficient morphing among a number of kernels).

The listening tests were done in a special listening room, employing a menu-driven playback system, capable of presenting blindly sound samples recorded from real-world devices and samples simulated employing the new software tool, and, for comparison, samples obtained by traditional linear, time-invariant convolution. The listener fills up a questionnaire for each sound sample, being able to switch them back and forth for better comparing.

The results show that this new device-emulation tool provides much better results than already-existing convolution plugins (which only emulate the linear, time-invariant behavior), requiring little computational load and causing short latency and prompt reaction to user's action.

1. INTRODUCTION

Not-linear convolution, based on Diagonal Volterra Kernels, was pioneered by one of the authors [1,2], who developed the theory and performed off-line tests, which did already prove how the usage of this technique can outperform traditional linear convolution (and also the "dynamic convolution" implemented by the IR-switching technique [3]).

The first realtime plugin implementing Volterra Kernel technology, named NEBULA, has been recently released by a small Italian startup company, Acustica Audio [4]: it also includes capabilities of emulating not-time-invariant systems, which are sampled over time. The time variance of their transfer function is later reconstructed by means of a very smooth interpolation technique. The time variation can be driven by the gain envelope of the signal (compressors, expanders, etc.), or can vary cyclically (phasers, etc.).

In some cases, the variation over time is due to the user, who is manually acting on pedals, knobs or sliders.

Also spatial variations can occur (for example, a binaural impulse response changes if the listener rotates his head). All the above cases are rendered without problems in this new software tool, as the variation of the transfer function can be controlled by the envelope of the signals, by an internal periodic cycle (the frequency can be user-controlled), or by an external MIDI control connected to knobs, sliders, keyboard, or even to an head-tracker.

The authors conducted three separate experiments: the first was aimed to assess the emulation of the nonlinearities of devices such as tube preamplifiers and spectral enhancers, under substantially time-invariant conditions.

The second experiment was aimed to the emulation of time-variant devices, such as compressors and flangers.

The third experiment was related to the real-time responsiveness, employing the plugin in user-controlled mode, emulating musician's effects such as a guitar pedal, and employing it for binaural rendering with headtracking (in this case, the user rotates his head, and the plugin changes smoothly the HRTF being convolved, following the head movement, and providing the illusion that the virtual position of the sound source is stable in space despite the rotation of the listener's head).

All the experiments, except the latter one (binaural), were blind and the listener had to compare the sound recreated by the not-linear convolution plugin with the one created by the real device or sound system being emulated. Also traditional linear, time-invariant emulation was included in the comparison.

The results demonstrated that the perceptions of the sound created by means of the new not-linear convolver are much closer to the real-world than that created by means of the traditional linear convolver.

Regarding the third experiment (realtime responsiveness), it was seen that this becomes good enough for satisfying a musician while playing its instrument only by setting the internal latency of the software, and the buffer size of the hardware (sound card) to very little values.

This can be problematic if the computer is not very powerful and well configured, or if it is being performing also other tasks. This means that this technology, in its current implementation as a VST plugin, is still not stable and responsive enough for being employed, for example, by performers during a concert, as the risk of signal dropouts is still present with these extremely-low-latency settings.

On the other hand, the binaural experiments did prove instead that, for this degree of responsiveness, the plugin is perfect even with much more conservative latency settings, providing a very nice headphones listening, definitely better than without head-tracking.

And the progressive and smooth interpolation manages very well also HRTF datasets with limited angular resolution, reconstructing properly the transfer functions even for intermediate angles, where no binaural IR was measured.

In conclusion, the new software tool can be seen as the current state of the art about emulation of real-world devices, including sound systems, room acoustics, and taking into account without limitations not-linear systems, moving sources or receivers, and any other usual cause of time variation.

2. DIAGONAL VOLTERRA KERNEL CONVOLUTION

This chapter provides a quick review of the digital-filtering method known as Diagonal Volterra Kernel convolution (also known as Vector Volterra Kernel). A more extended explanation is given in [2] and [5].

This method can be seen as a “reduced version” of the general Volterra representation of a not-linear system: while the general approach requires the usage of matrix kernels, having a dimensionality equal to the order of the kernel (so the second-order Volterra kernel is a square matrix, the 3rd-order one is a cubic matrix, and so on), the Diagonal Volterra Kernel approach employs mono-dimensional vectors at any orders.

The general Volterra formulation represents the output y of a not-linear system, being fed with the input signal x , as:

$$y(n) = \sum_{i_1=0}^{M-1} h_1(i_1) \cdot x(n-i_1) +$$

$$\begin{aligned}
 & + \sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} h_2(i_1, i_2) \cdot x(n-i_1) \cdot x(n-i_2) + \\
 & + \sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} \sum_{i_3=0}^{M-1} h_3(i_1, i_2, i_3) \cdot x(n-i_1) \cdot x(n-i_2) \cdot x(n-i_3) + \dots
 \end{aligned} \tag{1}$$

In this formulation, it is assumed that all the kernels are of the same size (M samples). The input signal x is linearly-convolved with the first-order kernel h₁, the squared values of the input samples are convolved with the second-order matrix h₂, and so on.

This approach provides, in theory, a complete representation of the not-linear behaviour of the system, under the hypothesis that it is still time-invariant, and that the Volterra expansion is reaching an order high enough and that the kernel size M is large enough for capturing the whole “memory” of the system.

However, the computational load required tends to explode beyond any reasonable limit as the order of the expansion is more than 3, whilst it is known that at least 5 orders are required for perceptually-acceptable simulation of harmonic-distorting devices such as loudspeakers or musical instruments.

The simplified formulation does only take into account the Volterra coefficients located along the diagonal of the higher-orders matrixes, so that the computational load now scales linearly with the order up to which the not-linear simulation is performed:

$$\begin{aligned}
 y(n) = & \sum_{i=0}^{M-1} h_1(i) \cdot x(n-i) + \sum_{i=0}^{M-1} h_2(i) \cdot x^2(n-i) + \\
 & + \sum_{i=0}^{M-1} h_3(i) \cdot x^3(n-i) + \dots
 \end{aligned} \tag{2}$$

In principle, this could provide a strong degradation in the accuracy of the simulation. In practice, instead, most of the Volterra coefficients outside the diagonal are usually very close to zero, and hence this “reduced” Volterra convolution is generally perceptually very good. In most cases, the advantage provided by the possibility to extend the simulation to higher orders outperforms the error due to discarding the coefficients locate outside the diagonal.

Furthermore, as explained in [2], the experimental measurement of the diagonal Volterra kernels can be performed easily, employing as a test signal an

Exponential Sine Sweep, and performing the deconvolution through aperiodic linear convolution with a suitable inverse filter. Very simple linear math is required for transforming the measured impulse responses in the Volterra vectors.

The capability of analyzing the not-linear properties of a system employing the Exponential Sine Sweep signal were first described by Gerzon [6], and only much later “rediscovered” by Griesinger [7] and Farina [8].

Hence, we provide here only a very quick description of the method: the test signal x is defined as:

$$x(t) = \sin[\omega_{var}] = \sin \left[\omega_1 \cdot T / \ln \left(\frac{\omega_2}{\omega_1} \right) \cdot \left(e^{\frac{t}{T} \ln \left(\frac{\omega_2}{\omega_1} \right)} - 1 \right) \right] \tag{3}$$

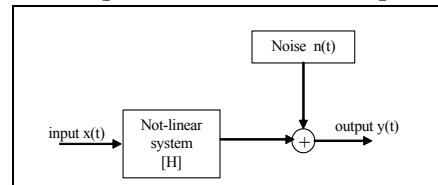


Figure 1. Flow diagram of the not-linear system

When the signal is introduced in the non-linear system, its output also contains harmonic distortion products, as shown here:

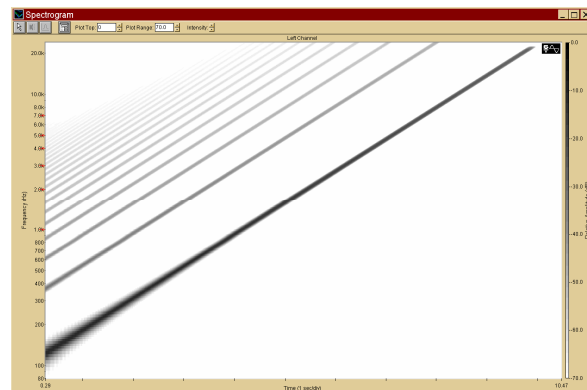


Figure 2. Spectrogram of the system’s response

It is possible to deconvolve the impulse response by applying to this response, by convolution, a proper inverse filter, which is simply the time-reversal of the excitation signal (3), equalized with a slope of 6dB/oct (time-reversal mirror plus whitening filter). This is the result:

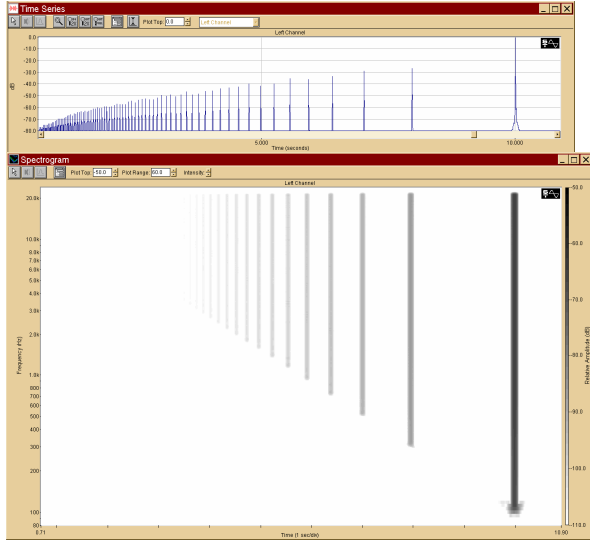


Figure 3. Spectrogram of deconvolved impulse responses

This process also removes almost completely the measurement noise, which is rejected typically by more than 70 dB.

The rightmost impulse response is the linear one, which is preceded by the second-order harmonic response, and so on. The measured impulse responses are not directly the Volterra kernels, but these are easily computed by solving a linear equation system.

The solution of this system allows for the computation of the unknown diagonal Volterra kernels h_i starting from the measured ordered impulse responses h'_i .

This is the solution for the first 5 orders, formulated in frequency-domain:

$$\begin{cases} \bar{H}_1 = \bar{H}'_1 + 3 \cdot \bar{H}'_3 + 5 \cdot \bar{H}'_5 \\ \bar{H}_2 = 2 \cdot j \cdot \bar{H}'_2 + 8 \cdot j \cdot \bar{H}'_4 \\ \bar{H}_3 = -4 \cdot \bar{H}'_3 - 20 \cdot \bar{H}'_5 \\ \bar{H}_4 = -8 \cdot j \cdot \bar{H}'_4 \\ \bar{H}_5 = 16 \cdot \bar{H}'_5 \end{cases} \quad (4)$$

After computing the values of the kernels from the measured multiple impulse responses, the non-linear convolution can be efficiently implemented following eq. 2.

3. EFFICIENT PARTITIONED CONVOLUTION

For real-time applications, indeed, the computational load required for performing all the multiplications and sums required by eqn 2 is still too large for an entry-level PC. However, a very efficient convolution scheme is known, making use of frequency-domain processing, which is very convenient when employing highly-optimized FFT subroutines.

By proper partitioning the kernels to be convolved, it is possible to simultaneously reduce the computational load and to meet the expectations of a reasonably low processing latency.

Some years ago, it was thought that the “optimal” partitioned strategy had to be searched with the goal of minimizing the total number of multiplications, as at that time this was the most costly operation for a digital processor. Under these assumptions, three different algorithms were proposed (and patented) by David McGrath [9], Bill Gardner [10] and Guillermo Garcia [11]. However, one of the authors subsequently proved [12] that these “advanced” algorithms, based on the non-uniform partitioning of the kernel, are systematically outperformed, on modern CPUs, by the previously-known, “suboptimal” algorithm initially proposed in 1966 by Stockham [13], which makes use of uniform partitions, thanks to the “trick” suggested by Barry Kulp in 1988 [14], consisting in performing the accumulation of the results of the convolution of the N blocks in frequency domain, prior of a single IFFT operation.

4. TIME-VARIANT SYSTEMS

Real-world systems often exhibit some degree of time-variance. Acoustic propagation in rooms, for example, is affected by air temperature and air movement; loudspeakers change their radiation properties due to temperature variations, and most electronic devices contains “adaptive” circuitry performing various form of automatic gain control, compression, expansion, limiting, soft and hard clipping. In some cases the variation of the system’s transfer function is driven by the amplitude of the signal, in others it is driven by an external control, which can be manual or based on an oscillator (for example, in a flanger an oscillator periodically changes the tuning of a parametric filter).

The most obvious approach for emulating time-variant devices is the usage of a convolver employing variable filtering coefficients. This can be seen to have some

point in common with the approach known as “dynamic convolution”, developed by Mike Kemp [3] with the goal of representing the behaviour of not-linear device. This approach revealed to be suboptimal for the original goal, but it is indeed a great idea for emulating linear, time-variant system: if each sample being processed is convolved with the “proper” set of filtering coefficients, this can produce a faithful emulation of a time-variant system.

However, “dynamic convolution” requires to employ the old, inefficient time-domain convolution algorithm, whilst we want to employ the computationally-efficient partitioned convolution scheme. However, it is trivial to see the bridge between the two techniques: a time-variant device can be emulated by changing the set of filtering coefficients for each data block being processed, instead of for each sample. If the blocks are small enough, this still allows to emulate faithfully a system which is changing over time.

And here we encounter the most tricky and advanced part of the new Nebula software, the part about which the authors of this article are not knowing all the details, as this part was independently developed by the authors of the Nebula software: as the usage of very small data blocks results in increased computational load, and in any case the sudden change of filtering coefficients can cause some audible artifacts, the authors of Nebula added an effective interpolation algorithm which “smoothes” the transition between blocks being processed with different filtering coefficients. This makes it possible to emulate with a reduced number of “time samples” also devices which exhibit extreme variations, as the binaural head related transfer functions while the head is rotating.

As this part of the Nebula software is not based on the scientific work of the authors of this paper, it will not be further commented here.

5. USING NEBULA SAMPLER TO CREATE A PRESET

This section explains the actual using of the Nebula software in order to measure a not-linear, and, possibly, not time-invariant device. In this example we will measure the dynamic response of a laptop pc audio system. Nebula is made of a sampler and a VST plugin. The sampler generates the test signals and includes a deconvolver (which of course uses the time-inverted

signal). The actual measurement is done by the sampler, while further processing is made by the VST plugin.

5.1. Driver Setup

The first tab in the Nebula sampler requires to setup the drivers and the device properties for the measurement. This is easily done since it supports ASIO drivers, and it is usually enough to select the ASIO4ALL to ensure proper functionality with any sound card. Also in this tab are the latency check and the run button, which starts the measurement once all the settings are complete.

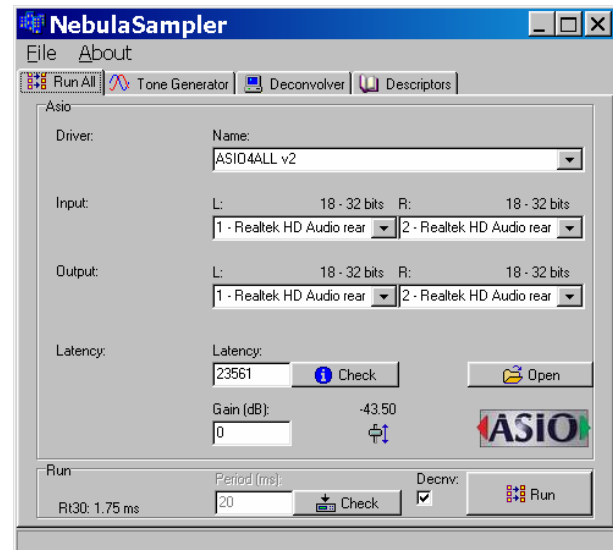


Figure 4. Driver setup in Nebula Sampler

5.2. Tone Generator

The second tab contains the tone settings. If the device requires a “dynamic” response measurement the general type should be set to dynamic (in Nebula, “dynamic” means “time-variant”).

The generated tone is a multiple sweep that starts with a sweep of the given length and then continues with a series of the set number of sweeps, whose amplitude decreases of the set amount of dB (the “step” value, typically 10) at each sweep, with 1 second of silence between sweeps. It is possible to set where the test signal is stored and how it is called. It is also possible to repeat the signal in order to increase the S/N ratio.

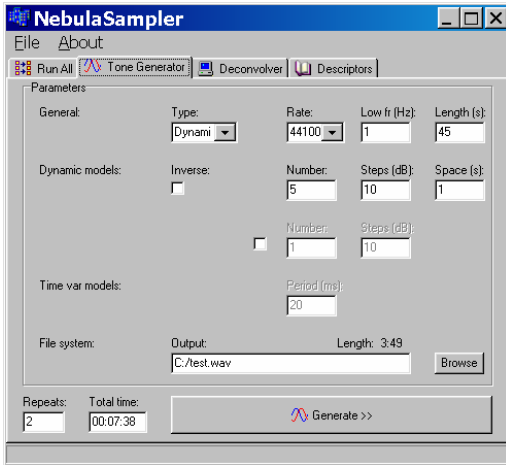


Figure 5. Tone Generator in Nebula Sampler

Here are some info about the generated signal:

	Left	Right
Min Sample Value:	-32768	-32768
Max Sample Value:	32768	32768
Peak Amplitude:	-.01 dB	-.01 dB
Possibly Clipped:	5075	5075
DC Offset:	.101	.101
Minimum RMS Power:	-62.39 dB	-62.39 dB
Maximum RMS Power:	-.05 dB	-.05 dB
Average RMS Power:	-14.02 dB	-14.02 dB
Total RMS Power:	-9.65 dB	-9.65 dB
Actual Bit Depth:	32 Bits	32 Bits

Here you can see the Spectral and Waveform views of the signal.

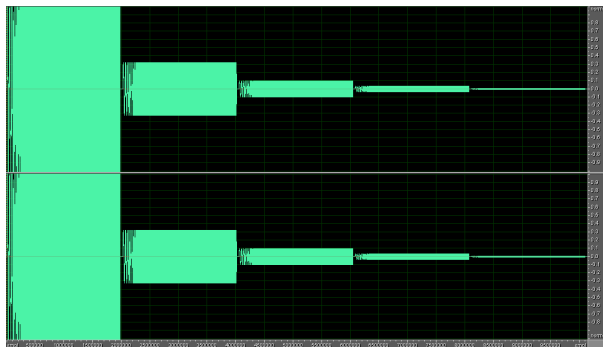


Figure 6. Waveform view of the test signal

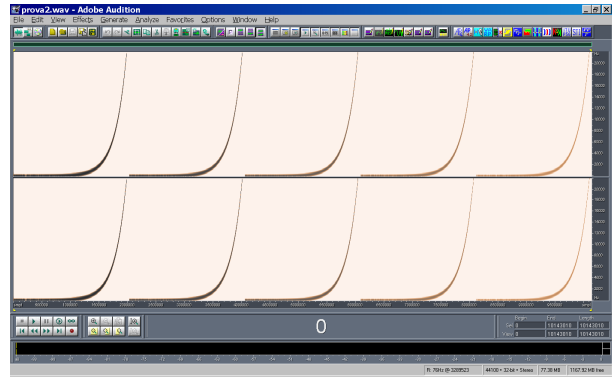


Figure 7. Spectral view of the test signal

5.3. Deconvolver

The third tab contains the deconvolver settings. The parameters must be the same specified for the test signal in the previous tab. It is also possible to select the algorithm. The enhanced one takes longer to compute but gives better results, as it is Volterra kernel-based. In the File system field one must select the multi-sweep generated in the previous tab.

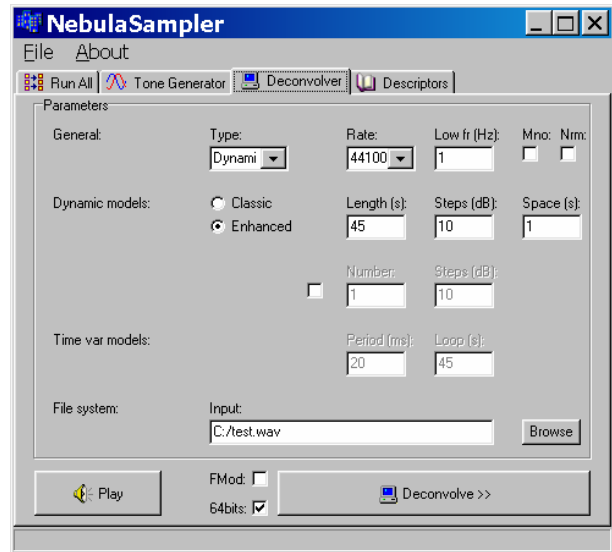


Figure 8. Deconvolver in Nebula Sampler

5.4. Descriptors

The last tab is really easy to use. In the first field one must select a template (which is found in the Template folder of Nebula program files). Since this is a dynamic measurement there are no doubts about selecting the Dynamic template. The following field, called Program

info allows the user to customize the name of the File system and of the Program. The last field provides kernel options. It is generally advised to use high values for H1 (10msc) as the higher the number of milliseconds set the more resources the final preset will use when loaded in the Nebula plugin. To disable a kernel it is enough to type 0 as its value.

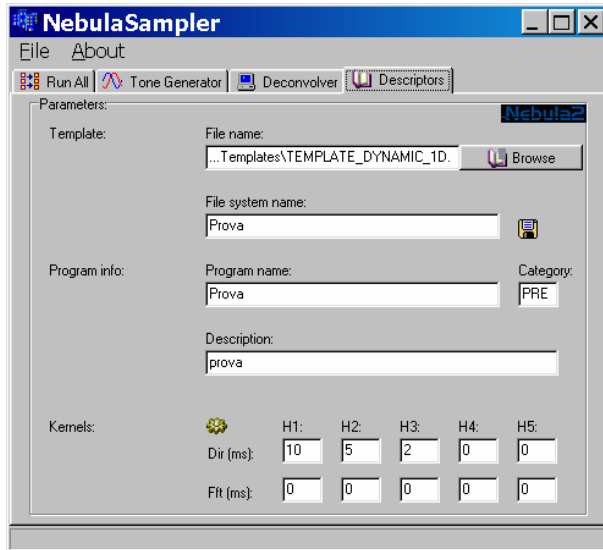


Figure 9. Descriptors tab in Nebula Sampler

5.5. Analyzing the device

Now everything is ready for the measurement to begin. Returning to the first tab we are able to check that everything is ok by pressing the check button. Once the check is complete one can start the measurement by pressing run. The Sampler performs the following actions:

- Checking of the noise floor level
- Checking the RT30
- Recording of a test tone Saving the recorded tone as a backup using the “output” name, so as to enable the user to deconvolve again with different settings without resampling
- Determines the best gain level automatically Deconvolves following the chosen algorithm
- Generates the final “Vector.xml” in each “file system name” directory (as set in the Descriptors tab), using the template file information. It creates a file for each kernel

- Calculates the program xml file, using the previously specified template and the Nebula Sampler information

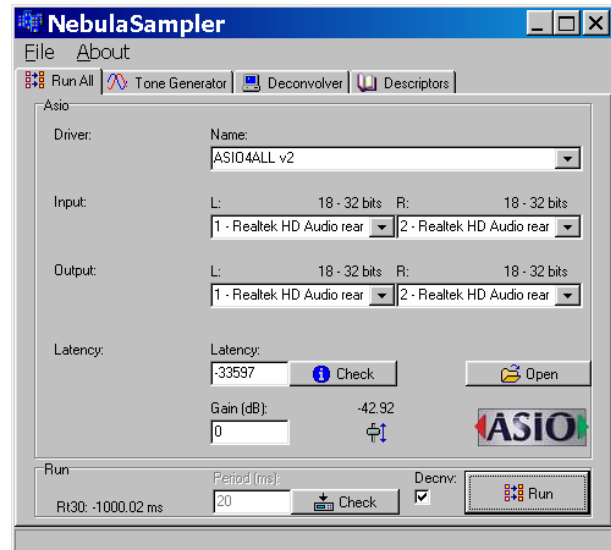


Figure 10. Run All tab in Nebula Sampler

Three files are generated in the previously specified folder. The test file is overwritten by its recording. Two other files are created. Their extension is .N2V and .N2P. The .N2V must be put in the Vectors folder of the nebula program files directory, while the .N2P must be put in the Programs folder of the same directory.

A new preset should now appear in the VST plugin.

6. USING THE VST PLUGIN

Once the preset is made it is possible to employ it as an effect. Specifically, we measured the dynamics of a laptop pc audio system (which has obvious some severe harmonic distortion due to the small loudspeakers, pushed to play a loud signal, and simultaneously also features some automatic gain control, thanks to the enhanced reproduction mode of the AZALIA chipset).

Thence it is possible to modify an audio file in order to make it sound like if was played through that audio system. The measurement procedure, making use of the microphone of the laptop computer, makes so that the resulting file contain the distortion both of the loudspeaker and of the microphone, while a listener hears a sound which is only distorted by the

loudspeaker. However, this fact is of minor importance, as the distortion and frequency filtering of the small loudspeaker are orders of magnitude bigger than those caused by the usage of an electret microphone for sampling the sound.

A first evaluation of the performances of the Nebula plugin can be obtained comparing the signal recorded during the measurement employing the sine sweep with an emulation obtained convolving the original test signal (shown in fig. 6) with the Volterra kernels sampled. The following figures show the sonograms of the real-world measured recorded signal and of the emulation:

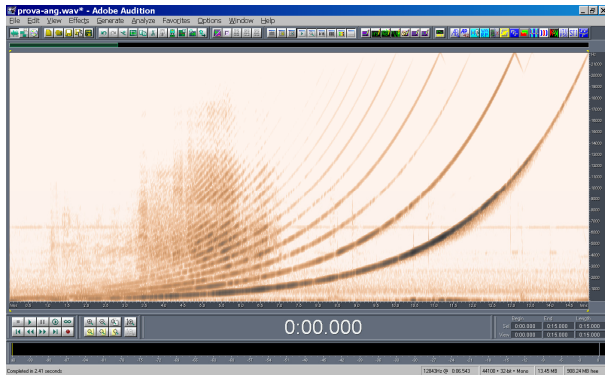


Figure 11. Real-world recording

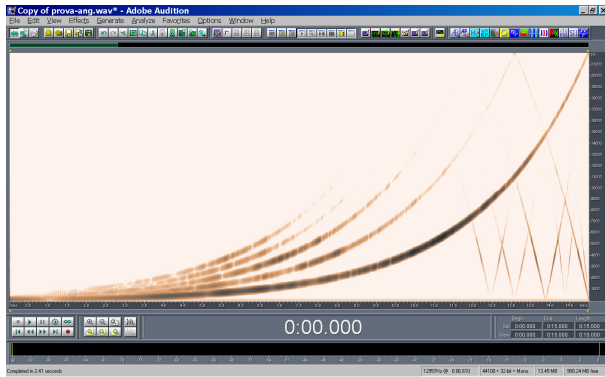


Figure 12. Nebula emulation

It can be seen how Nebula did only emulate the distortion up to 5th order, and that some artifacts do appear at the end of the sine sweep. However, the emulated frequency response is very close to the real-world one, as shown in the following figure:



Figure 13. Real-world and emulated spectra

Also the reverberation sampled in the room where the electro acoustical measurement was performed is faithfully reproduced, as shown in the following two figures:

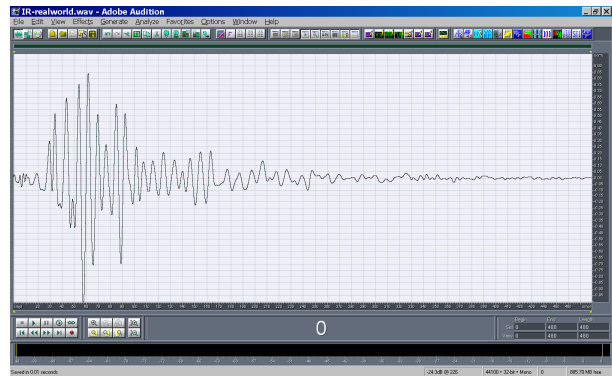


Figure 14. Real-world impulse response

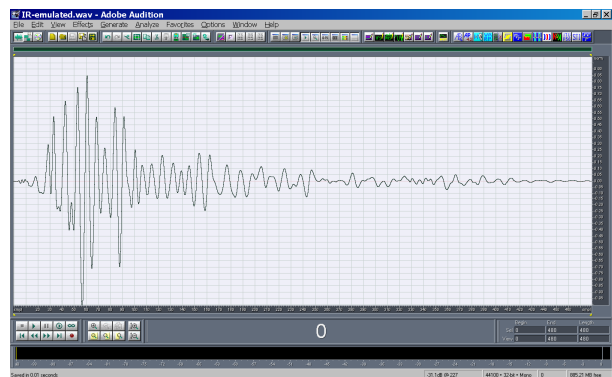


Figure 15. Emulated impulse response

However, the degree of “faithfulness” obtainable with the emulation plugin can only be assessed by psychoacoustical tests with a music sample.

6.1. Psychoacoustic test results

A single piece of music was employed for testing the performances of the Nebula plugin.

The piece was composed on the purpose employing GarageBand on a Mac computer, creating a musical mixture containing good dynamics, extended frequency response, stereo effects and a defined rhythmic base.

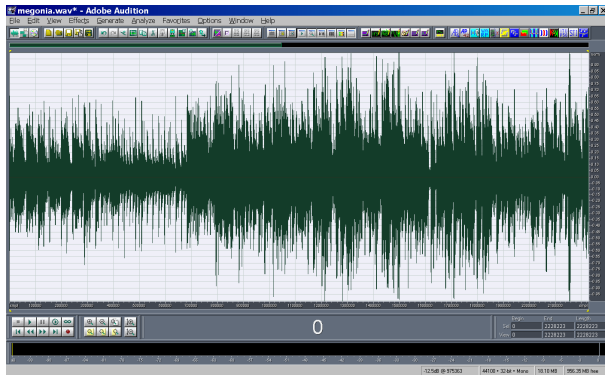


Figure 16. Original music file

The music piece was processed in two ways: “real-world” and “emulation”. The “real world” is obtained simply playing the signal and simultaneously recording, so it passed through the same hardware and acoustical devices already previously measured with nebula Sampler.

The “emulation” sound file is obtained applying the effect of the devices by means of the Nebula plugin:



Figure 17. Nebula 3 plugin

So we have obtained two sound files, one containing the “real-world” recording, and one the “emulation” performed by Nebula, as shown in the following two pictures:

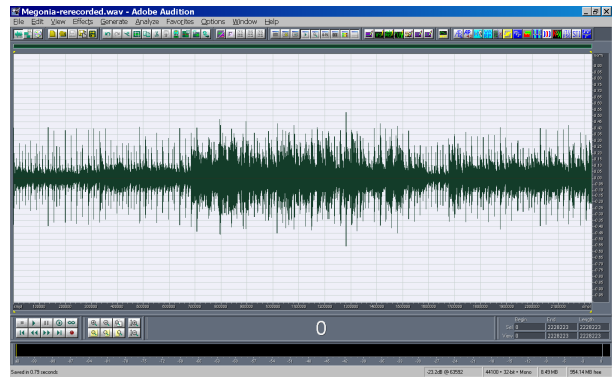


Figure 18. Real-world re-recorded music file

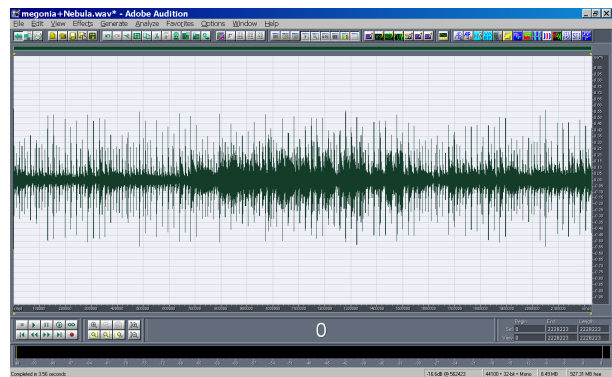


Figure 19. Emulated music file

A psychoacoustic test was performed during which listeners heard the two sound samples through high quality low-distortion headphones. They were then asked to give a decimal mark about the similarity between the two sounds heard (to a higher mark corresponds a higher similarity).

A panel of 13 trained subjects were employed.

The result was an average score of 7.28 with a standard deviation of 1.12, which was considered a good result, in the sense that the emulation is reasonably similar to the real-world.

The results of further psychoacoustical tests will be presented during the 123 AES Conference.

7. ACKNOWLEDGEMENTS

This work was supported by LAE, Laboratory for Acoustics and Electroacoustics, created thanks to funds

made available by Regione Emilia Romagna under the PRRIITT measure 1, action B.

8. NOTICE

The redaction of the present paper is incomplete, due to health problem of one of the authors close to the deadline for submitting. A completed version will be updated on the author's website (www.angelifarina.it) just after the AES conference.

9. REFERENCES

- [1] A. Bellini, G. Cibelli, E. Ugolotti, A. Farina, C. Morandi - "Non-linear digital audio processor for dedicated loudspeaker systems" - Proc. of International IEEE Conference on Consumer Electronics, Los Angeles, June 1998, and in IEEE Transactions On Consumer Electronics, vol. 44, August 1998, pp. 1024-1031."<Click and type/paste Reference text>"
- [2] A. Farina, A. Bellini, E. Armelloni - "Non-Linear Convolution: A New Approach For The Auralization Of Distorting Systems" - Proceedings of the 110th AES Convention, 2001 May 12-15 Amsterdam, The Netherlands
- [3] Michael J Kemp - "Analysis and Simulation of Non-Linear Audio Processes using Finite Impulse Responses Derived at Multiple Impulse Amplitudes", presented at the 106th AES convention in Munich, Germany, 8 - 11th May 1999. preprint number 4919.
- [4] Nebula plugin [HTTP://www.acusticaudio.com](http://www.acusticaudio.com)
- [5] A. Farina, E. Armelloni - "Emulation of Not-Linear, Time-Variant devices by the convolution technique" - Congresso AES Italia 2005, Como, 3-5 November 2005
- [6] M. Gerzon, P. Craven - "Practical Adaptive Room And Loudspeaker Equalizer for Hi-Fi Use" - 92 AES Convention. 24-27 March 1992, Vienna.
- [7] D. Griesinger - "Beyond MLS - Occupied Hall Measurement With FFT Techniques" Presented at the 101st Convention of the Audio Engineering Society Nov 8-11 1996
- [8] A. Farina - "Simultaneous measurement of impulse response and distortion with a swept-sine technique" - 108th AES Convention, Paris 18-22 February 2000
- [9] P. Single, D. McGrath - "Implementation of a 32768-tap filter using real time fast convolution" - 87th AES Convention, New York, 18-21 October 1989.
- [10] Gardner, W.G., - "Efficient convolution without input/output delay." - Journal of Audio Engineering Society, 43(3):127-136, March 1995.
- [11] G. Garcia - "Optimal Filter Partition for Efficient Convolution with Short Input/Output Delay" - 113 AES Convention, Los Angeles, 5-8 October 2002.
- [12] A. Torger, A. Farina - "Real-time partitioned convolution for Ambiophonics surround sound" - 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics - Mohonk Mountain House New Paltz, New York October 21-24, 2001.
- [13] T. G. Stockham Jr. - "High speed convolution and correlation" - AFIPS Proc. 1966 Spring Joint Computer Conf., Vol. 28, Spartan Books, 1966, pp. 229-233.
- [14] Barry D. Kulp - "Digital equalization using Fourier Transform techniques" - Pre-Prints of the 85th AES Convention, Los Angeles, 3-6 November 1988.