A-DAFX: ADAPTIVE DIGITAL AUDIO EFFECTS

Verfaille V., Arfib D.

CNRS - LMA
31, chemin Joseph Aiguier
13432 Marseille Cedex 20
FRANCE
(verfaille,arfib)@lma.cnrs-mrs.fr

ABSTRACT

Digital effects are most of the time non-adaptive: they are applied with the same control values during the whole sound. Adaptive digital audio effects are controlled by features extracted from the sound itself. This means that both a time-frequency features extraction and a mapping from these features to effects parameters are needed. This way, the usual DAFx class is extended to a wider class, the adaptive DAFx one. Four A-DAFx are proposed in this paper, based on the phase vocoder technique: a selective time-stretching, an adaptive granular delay, an adaptive robotization and an adaptive whisperization. They provide interesting sounds for electroacoustic and electronic music, with a great coherence between the effect and the original sound.

1. ADAPTIVE DIGITAL AUDIO EFFECTS (A-DAFX)

1.1. Definition

Adaptive digital audio effects are effects driven by parameters that are extracted from the sound itself [1]. The principle of this effect's class is to provide a changing control to an effect. This gives life to sounds, allowing to re-interpret the input sounds with a great coherence in the resulting sound.

The first examples of A-DAFx already known are the effects based on an evaluation of the dynamic properties of the sound (noise gate, expander, compressor, etc.). We generalize adaptive effects to those based on the spectral properties of the input signal. We will present the adaptive granular delay, the selective time-stretching, the adaptive robotization and the adaptive whisperization, all of those being implemented thanks to the phase vocoder technique.

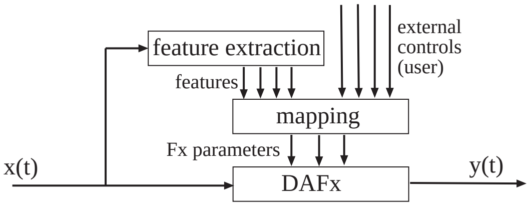

Figure 1: *Structure of an adaptive DAFx (A-DAFx), with $x(t)$ the input signal and $y(t)$ the output signal. Features are extracted, and then a mapping is done between these features and user controls as input, and effect parameters as output.*

To provide an A-DAFx (cf. fig.1), three steps are needed:

1. the analysis / feature extraction part ([3]);
2. the mapping between features and effects parameters;
3. the transformation and re-synthesis part of the DAFx.

1.2. Database sounds for our study

These effects will in particular be applied to gliding sounds (all sounds with rapid and continuous pitch changes). These sounds can come from an instrumental technique (vibrato, portamento, glissando) as well as from a whole musical sentence or from electro-acoustic sounds. The interest of gliding sounds for our study is that they provide an evolution of a great number of features. Moreover, as input sounds they produce by themselves interesting perceptive effects (example: vibrato, cf. [4]; transitions, cf. [5]), that is why they are usually used in electroacoustic effects.

2. FEATURE EXTRACTION

We consider two approaches for features extraction: the global features extraction (extraction thanks to a phase vocoder analysis and rather simple calculi, as a guarantee of real-time processing), and the high level features extraction (from a spectral line extraction).

2.1. Global features

The global features we extract are the voiced/unvoiced indicator, the fundamental frequency, the centroid, the energy (with an RMS method), the low/high frequency indicator, the odd/even indicator ([2]). On figure 2, we can see four global features (the centroid, the energy, the voiced/unvoiced indicator, the fundamental frequency) extracted from a sung voice.

2.2. High level features

Using the spectral lines model (sines + residual, [6]) to describe a sound, the features extracted are: partial frequencies and modulus, harmonics' and residual's energy, centroid of the harmonic part and the residual part. We will soon add other features, such as harmonics synchronism ([8]), harmonicity of partials, energy of the three sound components (sines + transient + residual, [7]).

2.3. Guided features extraction

We used a computer assisted partial tracking (fig.3), for harmonic analysis as well as inharmonic partial tracking. Since we are using gliding sounds, we want an easy-to-parameterize program, in

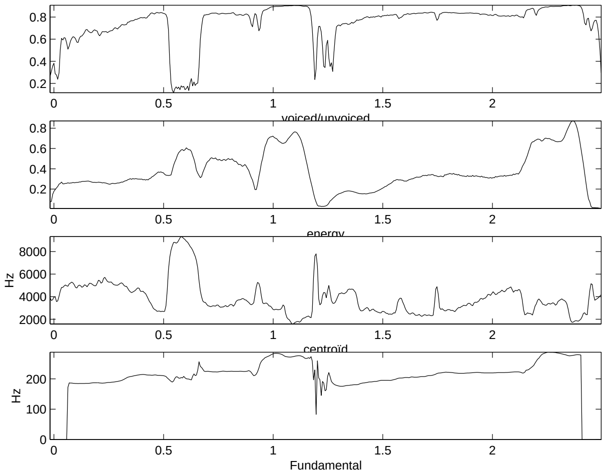

Figure 2: *Example of curve extraction for a sung voice, from up to down: voiced/unvoiced, energy, centroid and fundamental frequency, versus time (s).*

order to define how partials or harmonics evolve in time. The user plots on a sonagram an approximated frequency trajectory of a partial, thanks to segment lines. Then, an estimated fundamental $\hat{f}_1$ is calculated. The corresponding fundamental is computed in the following way: first, we look for the maximum of the magnitude spectrum in a track around $\hat{f}_1$, and from the maxima of two following frames, we calculate the real frequency in the phasogram.

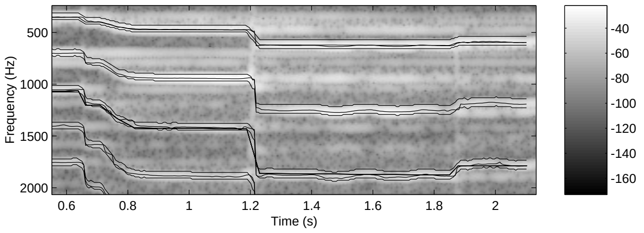

Figure 3: *Computer assisted partial tracking: from a segment line plotted by the user, the program extracts partials inside a track around the segment line. That way, reverberations are ignored.*

The track is defined as the interval: $[\hat{f}_i - \frac{\hat{f}_i}{i\ \alpha}; \hat{f}_i + \frac{\hat{f}_i}{i\ \alpha}]$. The greater the parameter α, the narrower the track. For higher harmonics, we use a weighted algorithm with the same idea of track, taking into account the estimated fundamental $\hat{f}_1$ and the calculated lower harmonics f_i and the weighted modulus:

$$\hat{f}_i(t) = (1-\beta) i \hat{f}_1(t) + \beta \sum_{k=1}^{i-1} \frac{f_k(t)}{k} \frac{\rho_k(t)}{\sum_{l=1}^{i-1} \rho_l(t)} \qquad (1)$$

This method allows first to take into account the slight inharmonicity of partials (thanks to the width of the track) and to avoid irrelevant values for the tracking of weak partials (thanks to the weight of the estimated fundamental). Moreover, it allows to track partials in sounds with a lot of reverberation.

3. MAPPING

The evolution of these parameters with time is depicted to the user by means of curves. Then, different kinds of mapping between extracted features and effect parameters are tested, depending on two concepts: the connection type and the connection law.

3.1. Connection type the connection law

Connections type between features and effect parameters can be simple, that is to say one to one or two to two. In a general case, it will be a N to M connection. The connection law is linear or non linear, for each connection. That means than we can combine linear and non linear laws in a mapping. The mappings we tried are linear, several to one parameter, or non linear several to one. From now on, we are using a linear combination of three features to one parameter, to which we apply a non linear function. Finally, the parameter obtained is fitted to the range of the effect parameter.

3.2. First step: linear combination three to one

Let us consider $k = 3$ features, namely $\mathcal{F}_k(t),\ t = 1, \ldots, NT$. First, we do a linear combination between one up to three features after normalizing them between 0 and 1. Noting $\mathcal{F}_k^M = \max_{t \in [1;NT]} (|\mathcal{F}_k|(t))$ and $\mathcal{F}_k^m = \min_{t \in [1;NT]} (|\mathcal{F}_k|(t))$, we obtain a weighted normalized feature:

$$\mathcal{F}_g(t) = \frac{1}{\sum_k \gamma_k} \sum_k \gamma_k \frac{\mathcal{F}_k(t) - \mathcal{F}_k^m}{\mathcal{F}_k^M - \mathcal{F}_k^m} \qquad (2)$$

with γ_k the weight of the k^{th} feature.

3.3. Second step: including a non-linearity

Then, we map the weighted normalized feature according to a linear or non linear mapping function $\mathcal{M}$, and we obtain a mapped curve.

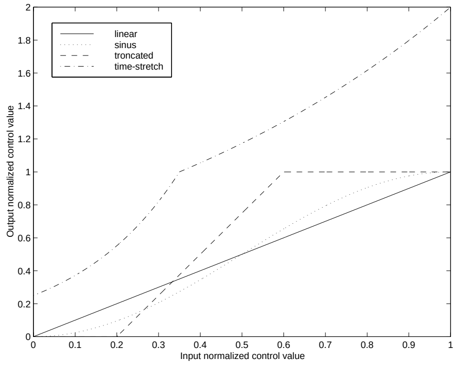

Figure 4: *Four different mappings: i) linear, ii) sinus, which increases the proximity to 0 or to 1 of the control value, iii) truncated between $t_m = 0.2$ and $t_M = 0.6$, which allows to select a portion of interest in the control curve, iv) time-stretched, contracted from $s_m = 1/4$ to 1 for a control value between 0 and $\alpha = 0.35$, and dilated from 1 to $s_M = 2$ for a control value between α and 1.*

The different mapping functions m we used are:

$$\mathcal{M}_1\left(\mathcal{F}_g(t)\right) = \mathcal{F}_g(t), \text{ linear} \qquad (3)$$

$$\mathcal{M}_2\left(\mathcal{F}_g(t)\right) = \frac{1+\sin(\pi(\mathcal{F}_g(t)-0.5))}{2}, \text{ sinus} \tag{4}$$

$$\mathcal{M}_3\left(\mathcal{F}_g(t)\right) = \frac{trunc}{t_M - t_m}, \text{ truncated} \tag{5}$$

$$trunc = t_m \mathbb{1}_{\mathcal{F}_g(t)<t_m} + t_M \mathbb{1}_{\mathcal{F}_g(t)>t_M} + \mathcal{F}_g(t) \mathbb{1}_{t_m<\mathcal{F}_g(t)<t_M}$$

$$\mathcal{M}_4\left(\mathcal{F}_g(t)\right) = s_m^{\left(\frac{\alpha-\mathcal{F}_g(t)}{\alpha}\right)} \mathbb{1}_{\mathcal{F}_g(t)\leq\alpha} + s_M^{\left(\frac{\mathcal{F}_g(t)\alpha}{1-\alpha}\right)} \mathbb{1}_{\mathcal{F}_g(t)>\alpha}, \text{ time-stretched} \tag{6}$$

with $\mathbb{1}_a$ the indicator function (which value is 1 if the test a is true and 0 if the test is false), $[t_m; t_M] \in [0;1]$ for the truncated mapping and $s_m \leq 1$ the contraction factor and $s_M \geq 1$ the dilatation factor of the time-stretched mapping. The time-stretched parameter $\alpha \in [0:1]$ divides the segment $[0;1]$ into two parts: the lowest $[0;\alpha]$ will be contracted, the upper $[\alpha;1]$ will be dilated. An example is given fig.4.

3.4. Third Step: fitting the mapped curve to the effect parameter's boundaries

Finally, we fit the mapped curve between the minimum Δ_m and maximum Δ_M values of the effect parameter. The effect parameter is given by:

$$\Delta(t) = \Delta_m + (\Delta_M - \Delta_m)\, \mathcal{M}\left(\mathcal{F}_g(t)\right),\ t = 1,..,NT \tag{7}$$

3.5. Interest of the proposed mappings

The sinus mapping increases the proximity to zero or to one of the curve. The truncated mapping allows to select a portion of interest in the curve. The time-stretching mapping is conceived for the time-stretching effect, and permits to choose the portion of the curve which correspond to contraction and to dilatation. The choice of perceptively interesting mappings is done by listening to each of several sounds of a database.

4. EFFECTS IMPLEMENTED AND RESULTS

We have already implemented four adaptive effects[1], based on the phase vocoder technique: a selective time-stretching, an adaptive robotization, an adaptive whisperization and an adaptive granular delay.

For each effect, the sounds is read buffer after buffer with an analysis overlap (cf. fig.5); each buffer is windowed; we then apply an FFT and calculate the features (with a constant buffer size) used for the effects. The synthesis buffer is calculated, windowed, overlapped and added to the output sound. The correction envelope of the output sound is also calculated by adding the power of the synthesis windows. It is not regular, since we may have used a synthesis hop size different from the analysis hop size. At the end of the process, the output sound is corrected sample by sample thanks to this envelope.

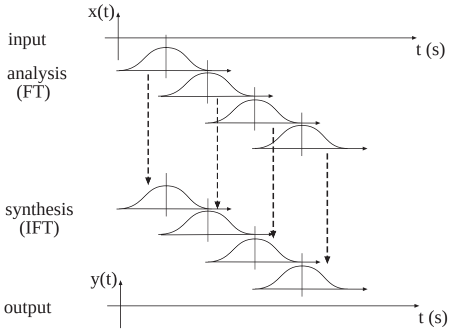

Figure 5: *Phase vocoder technique for analysing and treating the sound.*

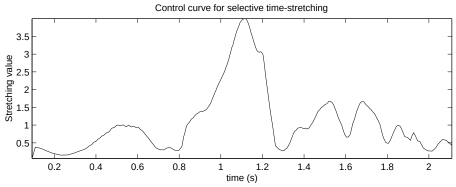

Figure 6: *Control curve for the selective time-stretching (stretching factor), used in the figure 7. It corresponds to the energy, mapped with the time-stretching mapping, and fitted between 0.125 and 4*

4.1. Selective time-stretching

For the selective time-stretching, the effect parameter used is the time-stretching factor. The synthesis hop size is fixed, and we change the analysis hop size, according to the fitted curve value. After windowing the input buffer and according to the time-stretching factor, we calculate the analysis hop size, and then unwrap the phase of the output buffer, before windowing and overlap-adding to the output buffer.

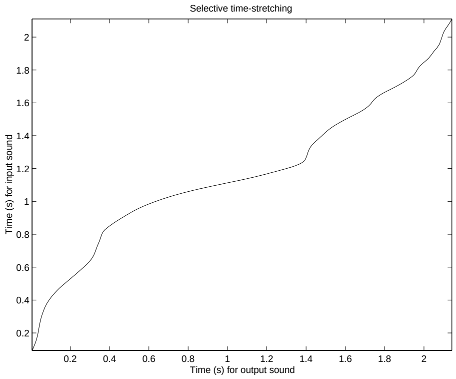

Figure 7: *Input/output time diagram for the selective time-stretching: contractions corresponds to a greater than 1 slope, and a dilatation to a lower than 1 slope*

[1] Three of these effects, namely the time-stretching, the robotization and the whisperization, were presented at the DAFx Working group in Madrid, 6-8 of June, 2001; they are also available on the COST DAFx Web Site at the following address: http://echo.gaps.ssr.upm.es/COSTG6/

Provided that the time-stretch factor varies from a smaller than one value to a greater than one value, it allows to slown down or to accelerate a sound. For example, with the voiced/unvoiced feature, we can slow down only the voicy parts of voice, thus allowing to automatically slow down a singing or spoken voice keeping the intelligibility of the text. In a general way, it allows us to re-interpret a musical sentence.

4.2. Adaptive robotization

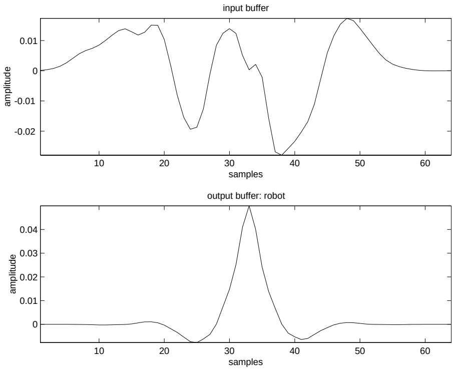

Figure 8: *Example of robotization applied on a 64 samples window: upside figure is the input buffer, downside figure is the output figure. We notice that only one peak is created.*

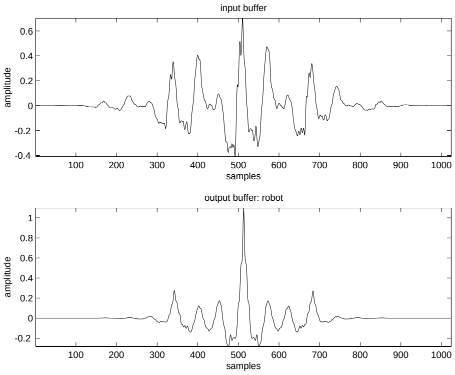

Figure 9: *Example of robotization applied on a 1024 samples window: upside figure is the input buffer, downside figure is the output figure. We notice that several peaks are created.*

The regular robotization corresponds to giving a zero value to the each FFT bin's phase. In that case, the reconstructed grain is the sum of cosines with zero phase, then shifted to the middle of the window: this creates a peak. One should use small window length, so that only one period of the pitch of the original sound is taken. On picture 8, we can notice that a small window (64 samples) creates just one peak, whereas a great window (1024 samples, fig.9) creates several peaks.

The pitch of the robot voice is then determined by the synthesis hop size between two synthesis windows. By changing this hop size, we obtain a robot voice with pitch changes, called adaptive robotization. Taking a normal voice, we can obtain a robot voice with the same pitch, as well as musical sentences re-interpreted by a robot with different pitches! This is true only with small windows : for too great windows (taking several periods), a comb filtering effect will appear, since we will have two pitches in the sound: one due to the peaks in each buffer (a pitch coming from the original sound), the other one due to the greatest peak of each group (a lower one). In that case, one should not call this robotization, but it is a perceptively interesting effect, that we can compare with the cross synthesis between the original sound and a robotized version of this sound.

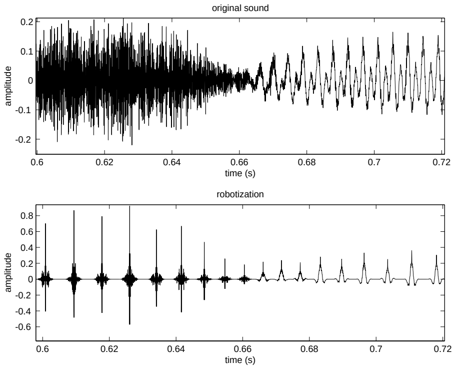

Figure 10: *Example of robotization, with a 256 samples window: i) upside figure: input sound, ii) downside figure: a-robotization.*

4.3. Adaptive whisperization

The regular whisperization is obtained by calculating the FFT of small input buffers (32 or 64 samples), and giving random values to the phase or to the modulus (cf. fig11).

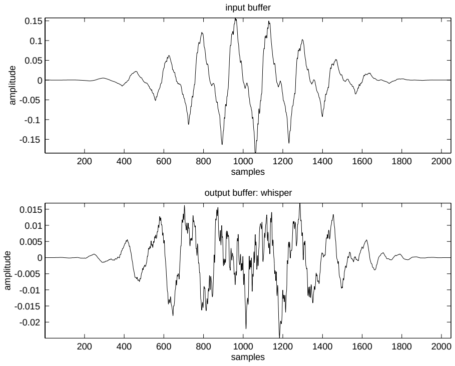

Figure 11: *Example of whisperization applied to a 2048 samples window: it clearly appears that phases are randomized.*

When re-synthesizing the sound with greater windows, we obtain less whispered voices. The adaptive whisperization is easily done by changing the input and output buffer length according to a given parameter, from very small values (whisperization) to great values (8192, for a nearly normal voice, since information reappears in the magnitude), cf. fig.13 left for the sonagram of the sound before this A-DAFx and fig.13 right for the sonagram of the treated sound. The example was obtained thanks to the control curve cf. fig.12: notice that small values on the left of the mapped

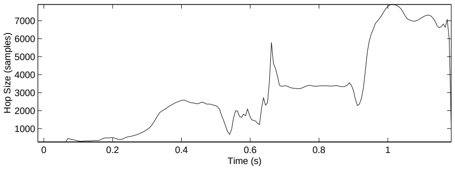

Figure 12: *Control curve for the adaptive whisperization on fig.13.*

curve corresponds to randomized values on the left of the right sonagram fig.13, and great values on the right of the mapped curve corresponds to the right part of the sonagram where harmonics information reappears.

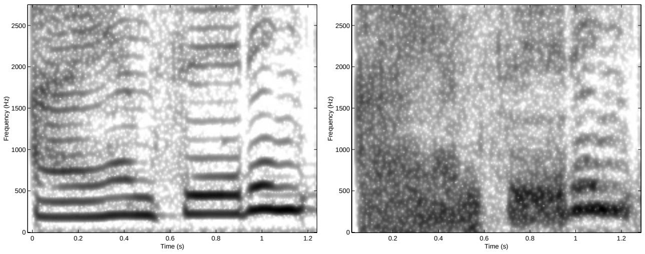

Figure 13: *Sonagram of a sung voice (left figure) and of an adaptive whisperization controlled by the pitch (right figure) with a 2048 points Blackman-Harris window, 256 samples hop size. We can see that harmonics clearly appears on the left, whereas on the right figure, for low pitch, the phases are totally randomized and for high pitch, information of harmonics reappears.*

This A-DAFx permits to have a voice going from a whisper mode to a normal mode, and inversely, and gives a very special timbre to a spoken or sung sentence.

4.4. Adaptive granular delay

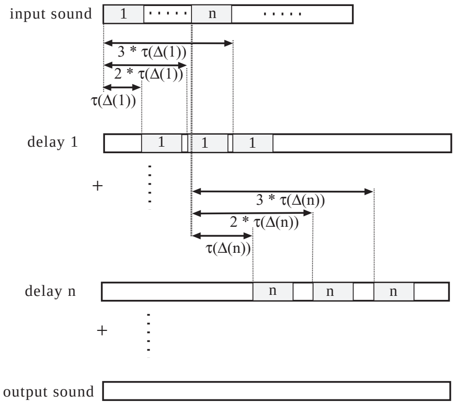

Figure 14: *Diagram of the adaptive granular delay: each frame/buffer is copied and delay, with a gain factor given by the user and a delay time given by the control curve* Δ

The adaptive granular delay is a granular delay changing according to the mapped curve: the output buffer is the same as the input buffer, but differently placed compared to the input sound, with a delay time $\tau(t)$ function of the control curve. Several copies of the original buffer are created, with delay times $k\ \tau(t)$ and gains $G(k) = g^k,\ g \in [0;1[$. An example of this effect is given with the diagram fig.14. One could also affect the mapped curve value to the feedback G gain and not the delay time.

Thanks to such adaptive granular delays, we obtained etheric sounds, with parts of interest (loud parts, harmonic portions in time, for example) appearing more or less present thanks to the repetitions of the grains.

5. CONCLUSIONS, PERSPECTIVES

These A-DAFx implemented with Matlab are extensions of existing effects. Their great interest is to provide possibilities for re-interpreting a musical sentence, changing timbre or presence of the musical events and stretching them. Their perceptive effects are very strong, and permits fine changes in the property of the sounds.

The next steps are: firstly to adapt more usual effects, such as reverberation, chorus and flanging, expander and compressor, equalizer; secondly to provide "cross adaptive effects" (effects applied on any sound and driven by features extracted from gliding sounds and/or a gesture). This will lead us to make a link with musical gesture and gesture controllers.

6. REFERENCES

[1] Arfib, D., "Des courbes et des sons", Recherches et applications en informatique musicale, Herms, pp. 277-286, 1998.

[2] Arfib, D., "Different Ways to Write Digital Audio Effects Programs", Proc. Workshop on Digital Audio Effects (DAFx-98), Barcelona, Spain, pp. 188-191, 1998.

[3] Rossignol, S., Rodet X., Soumagne J., Collette J.-L. and Depalle P., "Feature Extraction and Temporal Segmentation of Acoustic Signals", Proceedings of the ICMC, 1998.

[4] Honing. H. , "The vibrato problem, comparing two solutions". Computer Music Journal, 19 (3), 1995.

[5] Strawn J., "Analysis and Synthesis of Musical Transitions Using the Discrete Short-Time Fourier Transform", Journal of the Audio Engineering Society, volume 35, number 1/2, pp.3-13, 1987.

[6] Serra X., "Musical Sound Modeling With Sinusoids Plus Noise", published in C. Roads, S. Pope, A. Picialli, G. De Poli, editors; "Musical Signal Processing", Swets & Zeitlinger Publishers, 1997.

[7] Verma T., Levine S., and Meng T., "Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals", Proceedings of the ICMC, Greece, 1997.

[8] Dubnov S. and Tishby N., "Testing For Gaussianity and Non Linearity In The Sustained Portion Of Musical Sounds." Proceedings of the Journees Informatique Musicale, 1996.