CNAQ: a MATLAB acquisition tool for computing transfer functions over audio cards

Guillaume Pellerin

CNAM Paris, 5, rue du Vert Bois, 75003 Paris, France

guillaume.pellerin@cnam.fr

Since transfer function measurements require very precise buffer management, it is not easy to find any free and/or educational software which processes them with consumer audio cards, cheaper than professional ones but useful. Some people often use musical sequencers to generate and acquire signals but data cannot be easily analyzed in other specific engineering codes because the lengths of the samples cannot be easily fixed for example. Then, we decided to develop a specific software for general audio electro-acoustical measurements with these main constraints : choose a programming language compatible with existing educational and professional tools, promote multi-platform interpreted languages, be able to use consumer audio cards, publish the source code under the terms of a free software license. CNAQ provides a graphical interface under Matlab to process sweep sine measurements for any kind of acoustical or electronical application in the audio range. The current features of the software and some development ideas like implementing the convolution based Farina's method will be also presented.

1 Introduction

Many students and reseachers have already written some code to compute transfer functions and impulse responses of many dynamical systems. Although many results are obviously published in papers, it is really hard to find some non-proprietary tools which allow to edit and modify some analysis functions. In this situation, we decided to start the development of an open source project called CNAQ which could propose a shared versioned development space [1] and then some useful tools to make this kind of analysis. For pedagogical reasons, the Matlab language has been chosen since it is widely used in schools and laboratories even if the main Matlab software is not open sourced. Some libraries are then used to control the audio hardware which can be a simple sound card.

After introducing the Exponential Sine Sweep method (ESS) which gives some very good results compared to the Maximum Length Sequence (MLS) and Inverse Repeated Sequence (IRS) [6, 9, 10], the classical dividing method and the deconvoluting one are exposed. In many cases, we need to adapt the analysis with a temporal masking technique to avoid some oscillating artefacts around the two frequency extrema of the frequency response. This is needed also to avoid a pre-ringing phenomenon appearing at low frequencies on the impulse reponses. A view of the graphical interface is then presented and some future development ideas are proposed.

2 The exponential sine sweep (EES) method

The main drawback of common measurement method which use random or pseudo-random frequency distributions is that these type of signal contains very high instantaneous level variations. Although some natural or musical signals can contain some powerful transients, it is often a bad idea to measure dynamical systems with non-continuous signals because they make some typical non-linear systems generate distortions and then artefacts on impulse responses. This has been very well describe by Stan et al. [10].

Moreover, random based signals do not contain much energy on each frequency band. Thus, it is often necessary to compute averaged responses from many measurements. It is then a better solution to compute frequency responses using exponential sine sweep signals so that the energy could be concentrate on each instantaneous frequency value. In this situation, the analyzed

components are only feed with some sine variation. A. Farina [6, 7] proposes the swept-sine signal describe at equation (1) where T is the total time of the signal, f_1 the first frequency value (in Hz) and f_2 the last frequency value at T.

$$s(t) = \sin \left[\frac{2\pi f_1 T}{\ln \left(\frac{f_2}{f_1} \right)} \left(e^{\frac{t}{T} \ln \left(\frac{f_2}{f_1} \right)} - 1 \right) \right] \qquad (1)$$

The relation between the time and the frequency is given by the equation (2) so that it is easy to obtain any frequency value along the sweep.

$$t = T \frac{\ln \left(\frac{f}{f_1} \right)}{\ln \left(\frac{f_2}{f_1} \right)} \qquad (2)$$

Whereas this type of excitation signal provides some good signal-to-noise ratios, it also allow to compute harmonic distortion levels easily thanks to the Volterra kernels [8].

3 Computing the transfer functions

In order to obtain the transfer functions, two methods are proposed. The first is the classical division method which gives some good results until the exciting signal contains very low values. In this particular case, the transfer function can tend to infinity and the deconvolution method is needed.

3.1 Division method

If $x(t)$ is the exciting signal where t is the time vector, $y(t)$ the measured signal, $X(f)$ and $Y(f)$ their Fourier transforms respectively where f is the frequency vector f defined between f_1 and f_2, the transfer function $S(f)$ can be expressed like this :

$$S(f) = \frac{Y(f)}{X(f)}. \qquad (3)$$

In this case, one need a course to be careful that $X(f) \neq 0$.

3.2 Deconvolution method

When the exciting signal reaches values near zero - for example when an electroacoutical source is placed at a node of an acoustical mode - one can use a temporal

method based on deconvolution. The impulse response of the system $h(t)$ is then obtained by convolving $y(t)$ with a proper filtering impulse response :

$$h(t) = y(t) \otimes w(t), \tag{4}$$

where $w(t)$ is based on the time reversed vector of $x(t)$ divided by f to compensate the falling down of -3 dB/octave of the exciting signal :

$$w(t) = \frac{x^T(t)}{f}. \tag{5}$$

Practically, the convolution product can be very expensive in terms of computation time beacause the $w(t)$ must have the same length as the excitation signal. In Matlab, it seems faster to compute the convolution in the frequency domain. Hence, we can mofify the equation (4):

$$h(t) = \mathrm{Re}\left[\mathrm{FFT}^{-1}[\mathrm{FFT}(y).\mathrm{FFT}(w)]\right] \tag{6}$$

where Re is the real part function, FFT the Fast Fourier Transform and FFT^{-1} the inverse Fast Fourier Transform.

3.3 Temporal masking

One of the most important problem of the ESS is the generation of edge effects, i.e. bad oscillations, at each frequency extrema of the frequency response. The figure 1 shows this phenomenon on the basis of a numerical lowpass filter response compared to the theoritical one. In order to avoid it, CNAQ implements a masking function which imposes both a lower and a higher limit than the frequency range given by the user and then applies a fade in and a fade out to the excitation signal based on a Blackman window. Thus, all sweeps are generated between fixed frequencies $f_0 < f_1$ and $f_3 > f_2$. The duration T_1 of the fade in between f_0 and f_1 is then given by:

$$T_1 = \frac{T}{\frac{\ln\left(\frac{f_2}{f_0}\right)}{\ln\left(\frac{f_1}{f_0}\right)} - 1} \tag{7}$$

and the duration T_3 of the fade out between f_2 and f_3 by:

$$T_3 = T\left(\frac{\ln\left(\frac{f_3}{f_1}\right)}{\ln\left(\frac{f_2}{f_1}\right)} - 1\right). \tag{8}$$

The plot of the temporal mask applied to the fixed swept-sine is shown at figure 2.

4 Audio management

To ensure the management of the audio hardware directly in side Matlab, pa-wavplay [2] has been included in the package. This library uses the Port-Audio API which potentially allows a good compatibility with many devices playing multi-channel audio. At the moment, pa-wavplay has been compiled to support ASIO, DirectSound and Windows drivers. Many improvements of this solution have to be set in order to reach a full multi-platform compatibility (see 6.2).
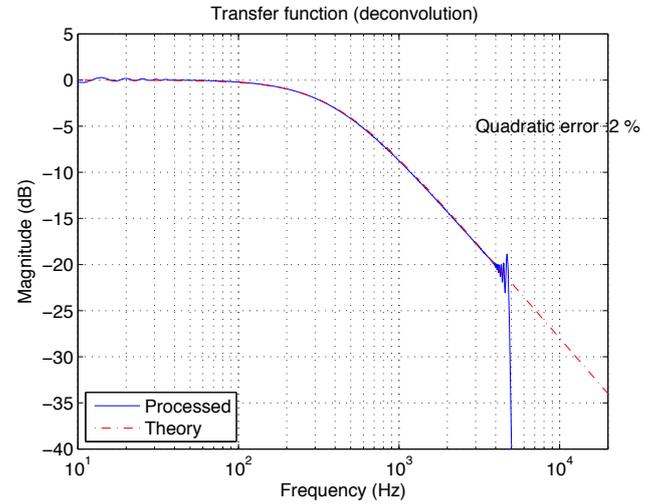
Figure 1: Transfer function of a numerical lowpass filter compared to the theoritical one ($f_c = 5000$ Hz).
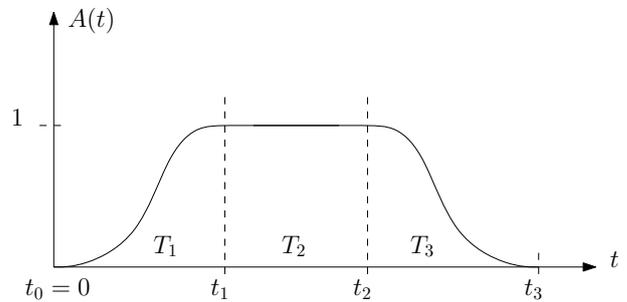
Figure 2: Temporal mask used for synthetizing excitation signals.

5 Graphical interface

The figure 3 presents a view of the front panel of CNAQ which is composed by seven main blocks :

Metadata: the general data of the next measurement. It includes the username, some comments and the workspace directory where all files will be saved.

Infos: the informations about copyrights and the version of CNAQ.

Monitor: the windows displaying the temporal and frequency representations of the input buffer in real time.

Channels: the sampling frequency F_s in Hz, the number of bits for the quantization, the menus and sliders to choose input and output channel numbers and gains.

Generator: the parameters of the excitation signal (sine, swept-sine, white noise, pink noise). The TRIG button allows to play the given sound.

Measurement: the TRIG button to make the measurement with the excitation signal defined in the Generator block, the ID number of the next measurement which is automatically incremented and

the LOAD and SAVE button to read or write metadata, all parameters and all signals in a .MAT format file.

Analysis: the menus to select which type of measurement is being performed, the method used for computing the transfer functions (division or deconvolution) and the domain to represent the results (time, frequency, time/frequency). An example of a looped measurement is shown at figure 4.

Modulus of the transfert function (group: pellerin, id: 4, channel : 1)
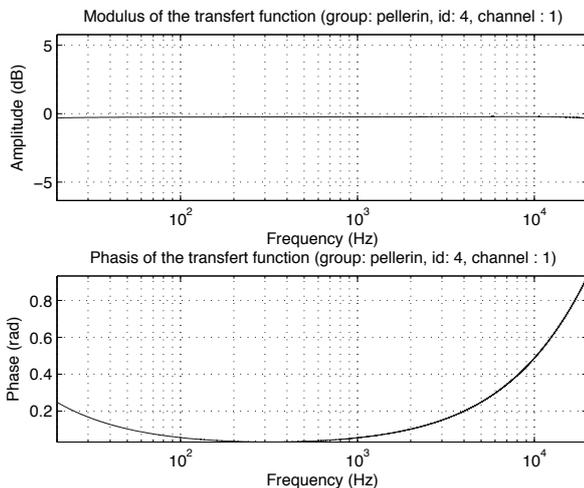
Figure 4: Example of a looped measurement over a sound card ($f \in [20; 20000]$ Hz).

6 Future development

6.1 Use Octave

One of our aims is then to make it compatible with Octave [3] which uses nearly the same syntax as Matlab but does not provide any graphical interface. This could provide a fully open source system to make measurements and analysis with a minimalistic console-based interface.

6.2 Externalize audio management

One of the main change for future versions will be the use of another audio engine. Indeed, many variations of latency time have been detected during tests on Windows systems[1]. This behavior is mainly due to some wrong priorities assigned by the operating system to the devices and it thus produces some phase shifts of the transfer function at high frequencies. To avoid this, it is planned to switch to an independant audio engine which could ensure some complete stability and compatibilty with any device and platform. This will be done with the JACK Audio Connexion Kit [4] controlled by some python scripts.

6.3 Community

You're welcome to download and try this software and give us some feedback. If you want to be involved in

---

[1] You can measure it with the function tools/get_latency.m.

the development or just had some useful functions you have written, feel free to commit them with the Subversion system. For more informations, see the dedicated website [1].

7 Conclusion

CNAQ is a new fresh open source tool to measure transfer functions and impulse responses over audio cards. The main functions has been already coded and tested on students which have scored it very well. Our aim is now to fix remaining minor bugs, implement harmonic distortion calculation thanks to the Volterra kernels and the Farina's method, and build a community of scientists who are welcome to upload some of their best analysis functions. See the website for more informations and to be informed of new releases of the software [1].

Acknowledgments

References

[1] CNAQ: http://svn.parisson.org/cnaq/.

[2] pa-wavplay: http://sf.net/projects/pa-wavplay/.

[3] Octave: http://gnu.org/software/octave.

[4] JACK: http://jackaudio.org.

[5] Qloud: http://gaydenko.com/qloud/.

[6] A. Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. Preprint 5093 presented at the 108th Convention of the Audio Engineering Society, Paris, France, 19-22 February 2000.

[7] A. Farina. Advancements in impulse response measurements by sine sweeps. Preprint 7121 presented at the 122th Convention of the Audio Engineering Society, Vienna, Austria, 5-8 May 2007.

[8] A. Farina and E. Armelloni. Emulation of non-linear, time-variant devices by the convolution technique. Annual Meeting the AES Italian Section, Como, Italy, 3-5 November 2005.

[9] S. Müller and P. Massarani. Transfert-function measurements with sweeps. *J. Audio Eng. Soc.*, 49(6):443, 2001.

[10] G. Stan, J.J. Embrechts, and D. Archambeau. Comparison on different impulse response measurement techniques. *J. Audio Eng. Soc.*, 50(4):249, April 2002.
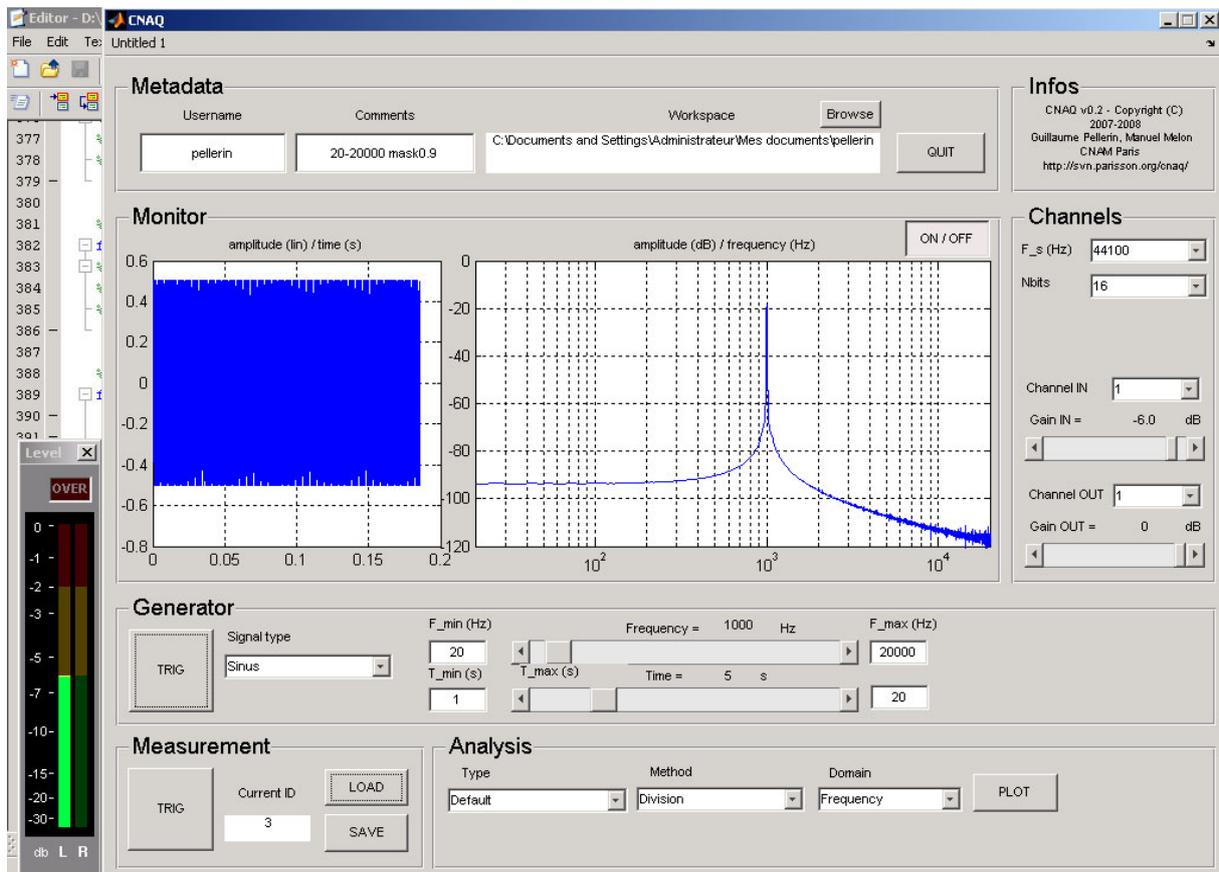
Figure 3: Graphical interface of CNAQ v0.2.