

---

**Model-Based Optimization/Estimation and Analysis: Paper  
ICA2016-530****A new toolbox for the identification of diagonal  
Volterra kernels allowing the emulation of  
nonlinear audio devices****Thomas Schmitz<sup>(a)</sup>, Jean-Jacques Embrechts<sup>(b)</sup>**<sup>(a)</sup>University of Liège, Belgium, T.Schmitz@ulg.ac.be<sup>(b)</sup>University of Liège, Belgium, jjembrechts@ulg.ac.be**Abstract**

Numerous audio systems are nonlinear. It is thus of great importance to study them and understand how they work. Volterra series model and its subclass (cascade Hammerstein-Wiener model) are usual ways to modelize nonlinear systems. However the identification methods of these models are still considered as an open topic. Therefore we have developed a new optimized identification tool ready for use and presented as a Matlab toolbox. This toolbox provides the parameters of the optimized sine sweep needed for the identification method, it is able to calculate the parameters of the Hammerstein model and to emulate the output signal of a nonlinear device for a given input signal. To evaluate the toolbox, we modelize a guitar distortion effect (the *Tubescreamer*) having a total harmonic distortion (THD) comprised in the range 10-23%. We report a mean error of less than 0.7% between the emulated signal and the signal coming from the distortion effect.

**Keywords:** nonlinear model, sine sweep, Volterra & Hammerstein model

# A new toolbox for the identification of diagonal Volterra kernels allowing the emulation of nonlinear audio devices

## 1 Introduction

The modelization of nonlinear systems has been a central topic in many engineering areas, as most real-world devices exhibit nonlinear behaviors. The study of these topics is of great interest as they include a large range of applications such as industrial processes, control systems, economic data, life sciences, social systems, physics and many more. This project focuses on one particularly interesting application: the modelization of the audio chain of a guitar player. Such a chain is usually composed of compressors, distortion effects, preamplifiers, amplifiers, loudspeakers and microphones. Most of these elements exhibit nonlinear characteristics. The idea is to replace these devices by a computer emulation that could be suitable for guitar players in many situations [1].

Most weakly nonlinear systems can be approximated by linearization. In that case, the modelization can be achieved by the convolution technique, as for classical Linear Time Invariant (LTI) systems. However, stronger nonlinearities need more specific models in order to keep an accurate approximation of the real system.

In our research, we focus on a technique based on the Volterra series model [2, 3], which recently received an increased interest with the improvement of computer's performance. Considering only static (memoryless) nonlinearities, this model can be simplified into a Hammerstein model (see Fig. 1) which is a cascade of  $M$  parallel (the system's order) branches where each branch  $m$  is composed by a static monomial nonlinearity (putting the input signal to the power  $m$ ) followed by a linear filter  $h_m[n]$  which is the  $m^{\text{th}}$  Hammerstein kernel (also called diagonal Volterra kernel) [3, 4, 5].

In the last few years, a nonlinear convolution method [6] has been elaborated with the aim of emulating nonlinear audio systems such as loudspeakers [1, 7], guitar distortion generators [8, 9] (the famous *Tube-Screamer* is often taken as an example) and compressors [10]. The key of the method is the identification of the diagonal Volterra kernels by means of the synchronized sine sweep technique [9]. However, the practical implementation of this identification technique is complex due to many possible sources of error. The nature of these errors will be explained

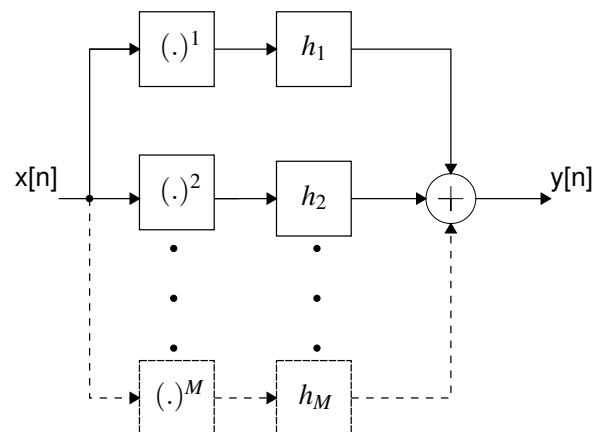


Figure 1: The Global Polynomial Hammerstein model

in details in a journal paper entitled *Optimized Hammerstein Kernels Identification by Means of a Sine Sweep Technique Applied to Nonlinear Audio Devices Emulation* which is presently submitted for publication. This paper describes how to accurately measure the Hammerstein kernels using our simple *Matlab* toolbox program[11]. This toolbox has been evaluated on some real nonlinear audio devices.

The paper is organized as follows : the basic identification technique is summarized in section 2. In section 3, we present the features of our *Matlab* toolbox. In section 4, we provide a step by step example of the Hammerstein kernels identification of a guitar distortion effect (the *Tubescreamer*). Finally, section 5 presents our conclusion about this toolbox.

## 2 Basic identification method

This section briefly presents the principles of Volterra series and Hammerstein model. The full development of the method will be shortly given in another paper [12].

### 2.1 Volterra Series

Considering time invariant nonlinear systems, the Volterra series expresses the relationship between an input signal  $x(t)$  and its corresponding output signal  $y(t)$  as an infinite sum of multiple convolutions [2, 3]:

$$y(t) = \sum_{m=1}^{\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} v_m(\tau_1, \dots, \tau_m) \times \prod_{i=1}^m x(t - \tau_i) d\tau_1 \dots \tau_m \quad (1)$$

Where  $\{v_m(\tau_1, \dots, \tau_m)\}_{\forall m \in \mathbb{N}^+}$  are the Volterra kernels characterizing the system.

### 2.2 Cascade of Hammerstein Models

It can be shown that any continuous nonlinear system can be represented by a set of M parallel branches composed by a static non linearity  $P_m(\cdot)$  surrounded by two linear filters [13]. The cascade Hammerstein model is a subclass of this general model where we consider a static nonlinear monomial function followed by a linear filter  $h_m(t)$  (the Hammerstein kernels) as shown in Fig.1. Considering a nonlinear discrete time causal system with a finite order M, we can express the input/output relationship as:

$$y[n] = \sum_{m=1}^M h_m[n] \otimes x^m[n] \quad (2)$$

We can see from Eqs. (1) and (2) that the Hammerstein model corresponds to the Volterra series model if the Volterra kernels are diagonal (ie. the Volterra kernels differ from zero only for the values  $\tau_1 = \tau_2 = \dots = \tau_M$ ). That is why the cascade of Hammerstein models is also called a diagonal Volterra model.

## 2.3 Hammerstein Kernels Identification

In order to extract the Hammerstein kernels, we need to excite the nonlinear system with a suitable input signal. Such a signal should allow us to separate the different Hammerstein kernels from each other after the deconvolution by its inverse filter. The next two subsections explain which suitable input signal is chosen and how it can be used to compute the Hammerstein kernels.

### 2.3.1 Exponential Sine Sweep Phase Properties

Farina et al. [6] proposed to use the properties of the Exponential Sine Sweep (*ESS*) signal in order to identify the Hammerstein kernels. The goal is to separate the harmonic responses (created by the harmonic distortion of the nonlinear system) from each other.

From now on, we propose to develop all our equations in discrete time for a better correspondence with the practical implementation. Let  $ss[n]$  be the *ESS*:

$$ss[n] = A \sin(\phi[n]) \quad (3)$$

The phase  $\phi[n]$  grows exponentially such that :

$$\phi[n] = \omega_1 \frac{R}{f_s} \cdot (e^{\frac{n}{R}} - 1) \quad \forall n \in [0, N - 1] \quad (4)$$

Where:

- $R = (N - 1) \cdot (\log \frac{\omega_2}{\omega_1})^{-1}$  is the inverse frequency changing rate.
- $N$  is the length of the sweep in samples.
- $\omega_1, \omega_2$  are the initial and final angular frequencies respectively.
- $f_s$  is the sample rate.

An interesting property of the *ESS* [1] is that the  $m^{th}$  harmonic of the *ESS* is the *ESS* itself delayed by  $\Delta_m$ , as:

$$m \cdot \phi[n] = \phi[n + \Delta_m] - B(m - 1) \quad \forall n \leq N - 1 - \Delta_m \quad (5)$$

Where:

$$\begin{cases} B = \frac{\omega_1 \cdot R}{f_s} \\ \Delta_m = R \cdot \log(m) \end{cases} \quad (6)$$

Choosing  $B \in 2k\pi$ , with  $k \in \mathbb{N}$  allows to neglect the  $B$  term in Eq. (5) as it just adds a multiple of  $2\pi$  to the sine phase. Note that the values  $\Delta_m$  are real. As our problem is discretised for practical implementation, this could lead to phase mismatches between the different harmonic impulse responses. From now on, consider that  $\Delta_m \in \mathbb{N}$  until Eq. (14) where this problem will be discussed.

The property of the *ESS* in Eq. (5) is interesting because the contribution of an harmonic  $m$  in the impulse response will be delayed by  $-\Delta_m$  from the first order impulse response, after its deconvolution by the *ESS* inverse filter  $\overline{ss}[n]$ :

$$\begin{cases} ss(\phi[n]) \otimes \overline{ss}(\phi[n]) = \sum_{k=0}^{N-1} ss[k] \cdot \overline{ss}[n-k] = C \cdot \delta[n-n_0] \\ ss(m \cdot \phi[n]) \otimes \overline{ss}(\phi[n]) = D \cdot \delta[n-(n_0-\Delta_m)] \end{cases} \quad (7)$$

Where  $C$  and  $D$  are constant amplitude factors.  $n_0$  is the position along the time axis of the first order (linear) impulse response which depends on the *ESS* length ( $n_0 = N - 1$ ). The inverse *ESS* can be calculated in several ways [12]. In this work, we use the time reversed and amplitude corrected version :

$$\overline{ss}[n] = ss[N-1-n] \cdot \left( \frac{\omega_2}{\omega_1} \right)^{\frac{-n}{N-1}} \quad (8)$$

### 2.3.2 Power sine properties

It can be shown [12] that the power formulas of the *ESS* signal (defining  $SS^n$  as the Fourier transform of  $(\sin^n(x))$ ) in the positive frequency domain are equal to :

$$\begin{cases} SS^{2n+1} = SS \cdot \frac{-1^n}{4^n} \sum_{k=0}^n (-1)^k C_{2n+1}^k e^{j\omega\Delta_{2n+1-2k}} = a_n \cdot SS \quad \forall n \in \mathbb{N} \\ SS^{2n} = j \cdot SS \cdot \frac{-1^n}{2^{2n-1}} \sum_{k=0}^{n-1} (-1)^k C_{2n}^k e^{j\omega\Delta_{2n-2k}} = b_n \cdot SS \quad \forall n \in \mathbb{N}_0 \end{cases} \quad (9)$$

Where the DC component has been neglected.

### 2.3.3 Deconvolution Process

Including the *ESS* signal (3) as an input to the model (2) gives:

$$y[n] = A \sin(\phi[n]) \otimes h_1[n] + A^2 \sin^2(\phi[n]) \otimes h_2[n] + \dots + A^m \sin^m(\phi[n]) \otimes h_m[n] \quad (10)$$

Applying (9) for sine powers, the deconvolution of the Hammerstein model in the frequency domain leads to:

$$Z(j\omega) = Y(j\omega) \overline{SS}(j\omega) = H_1(j\omega) + \sum_{m=1}^{MM} [a_m H_{2m+1}(j\omega) + b_m H_{2m}(j\omega)] \quad (11)$$

Where  $MM = (M - 1)/2$  and  $H_m(j\omega)$  is the Fourier transform of  $h_m[n]$ . In the time domain, Eqs. [9,11] show that  $z[n]$  is a superposition of several versions of each Hammerstein kernel  $h_m[n]$  delayed by  $\Delta_k$  ( $e^{j\omega\Delta_k}$  in the Fourier domain). We name  $g_m[n]$  the superposition of the  $h_m[n]$

having the same delay, which gives:

$$\begin{pmatrix} G_1(j\omega) \\ \vdots \\ G_M(j\omega) \end{pmatrix} = \mathbf{T} \begin{pmatrix} H_1(j\omega) \\ \vdots \\ H_M(j\omega) \end{pmatrix} \quad (12)$$

We can develop the elements  $\mathbf{T}_{u,v}$  with  $u, v \leq M$  by developing Eq. (11) in the matrix form and grouping in  $G_u(j\omega)$  the  $H_v(j\omega)$  having the same phase delay.

$$\mathbf{T}_{u,v} = \begin{cases} \frac{(-1)^{\frac{1-u}{2}}}{2^{v-1}} C_v^{\frac{v-u}{2}} & \forall v \geq u \ \& \ \text{mod}\left(\frac{u+v}{2}\right) = 0 \\ 0 & \text{else} \end{cases} \quad (13)$$

Finally the Hammerstein kernels in the positive frequency domain are given by the following relationship:

$$\begin{pmatrix} H_1(j\omega) \\ \vdots \\ H_M(j\omega) \end{pmatrix} = (\mathbf{T})^{-1} \begin{pmatrix} G_1(j\omega) \\ \vdots \\ G_M(j\omega) \end{pmatrix} \quad (14)$$

One can notice that we have made the assumption that  $B \in 2k\pi$ , with  $k \in \mathbb{N}$  and  $\Delta_m \in \mathbb{N}$  which is not necessarily the case. The  $\mathbf{T}$  matrix has to be corrected to take this into account [12].

### 3 Presentation of the toolbox

The toolbox is composed of three parts, (1) the sine sweep generation, where the signal to send to the nonlinear device under test (*DUT*) is generate, (2) the calculation of the Hammerstein kernels through the signal recovered at the output of the *DUT* and (3) the emulation of the nonlinear *DUT*.

#### 3.1 Sine sweep generation

The tab *Sweep generation* (see Fig. 2) is the first step to measure a nonlinear device. In this step a proper *ESS* signal is generated. The parameters of the sweep have to be optimized in order to have the amplitude of the *ESS* last sample as close as possible to zero [14]: this is done using the *Mixed Integer NonLinear Problem* (MINLP) global optimality solver *Scip* [15]. The idea is to give a range of possible values for the *ESS* and let the solver choose the best parameters values. The parameters of the *ESS* are the following :  $f_1$  is the lower bound of the measured bandwidth,  $f_2$  is the upper bound of the measured bandwidth,  $N$  is the duration of the sweep in samples,  $f_s$  is the sampling frequency, *Fade\_In* is the duration in seconds of the fade in of the *ESS* to avoid oscillations in the bandwidth.

Note that the calculation of the Hammerstein kernels is correct only on the domain  $f \in [M * f_1, f_2]$  where  $M$  is the order of the *DUT* [12]. For example, if one wants to calculate the Hammerstein kernels on the domain [100, 20000]Hz up to the 5<sup>th</sup> order, we suggest to choose  $f_1 = 100/5 = 20\text{Hz}$ .

### 3.2 Hammerstein kernels calculation

The second step (shown in Fig. 3) is to calculate the Hammerstein kernels defined by Eq. (14).

Load the  $y$  response in *.wav* format i.e. the output of the *DUT* when it is driven by the *ESS* at the input. Deconvolve  $y[n]$  with the inverse filter  $\overline{ss}[n]$  in order to obtain the impulse response  $z[n]$ . Choose the number of Hammerstein kernels (the order of the nonlinear *DUT*) and the desired maximum length (in samples) of the Hammerstein kernels (it must be a power of 2 in order to perform *FFT*). The parameter *Decay\_cut* allows to cut the  $g_m[n]$  kernels at positions  $n_0 - \Delta_m + \text{decay\_cut}$  instead of  $n_0 - \Delta_m$ . The purpose of this decay is to perform a better reconstruction of the Hammerstein kernels [12]. The decay must be negative, a value comprised in the range [-100 -1000] seems reasonable. After cutting the  $g_m[n]$ , the Hammerstein kernels can be calculated and plotted in the time and frequency domain. The Hammerstein kernels are automatically saved in the *MATLAB* work space.

### 3.3 Emulation of the *DUT*

The last step is the emulation of the *DUT* for a given input signal. Using Eq. (2) and the Hammerstein kernels previously calculated, the emulated output signal of the *DUT* can be computed for any given input signal (note that the simulation works better if the amplitude of the input signal remains relatively constant and is approximately the same as the one used for the *ESS* signal).

Load the desired input signal in *.wav* format. Perform the nonlinear convolution by clicking on the *Convolve* button. The emulated signal is displayed and can be exported in the work space.

## 4 Example of nonlinear emulation

To verify the efficiency of this toolbox, we evaluate it by emulating the well-known guitar distortion effect the *Tubescreamer* [16]. The button *Tone* is positioned at the middle and the *Overdrive* and *Level* buttons are at their maximum to generate a maximum level of distortion.

The first step is to choose the *ESS* parameters: we want to emulate the *Tubescreamer* between [60 20000]Hz until the 10<sup>th</sup> order (60Hz correspond to a guitar tuned in C). The initial frequency of the sweep will be around  $f_1 = \frac{60}{10} = 6\text{Hz}$ , the final frequency is comprised between 20000Hz and the Nyquist frequency. The length in samples of the *ESS* is chosen to correspond to a duration of about 20 seconds. After clicking the *Optimize* button we obtain the result as shown in Fig.2 and we can generate the file *sweep.wav*.

We can now open an audio sequencer and send the *ESS* (*sweep.wav* file) to the *DUT*. At the output of the *DUT* a signal is received and saved as *yDUT.wav*. On the *Hammerstein kernels calculation* tabs we load the *yDUT.wav* file by clicking the *Load y response* button. After clicking the *Deconvolve* button, we obtain  $z[n]$  a set of impulse responses as illustrated in the Fig.3

Regarding the signal  $z[n]$ , the number of kernels for our model has been fixed to ten kernels ( $M = 10$ ). Clicking the *cut into  $g_m[n]$*  button extracts the ten kernels from  $z[n]$  to  $g_m[n]$ . The *Hammerstein Compute* button is then clicked to calculate the Hammerstein kernels.

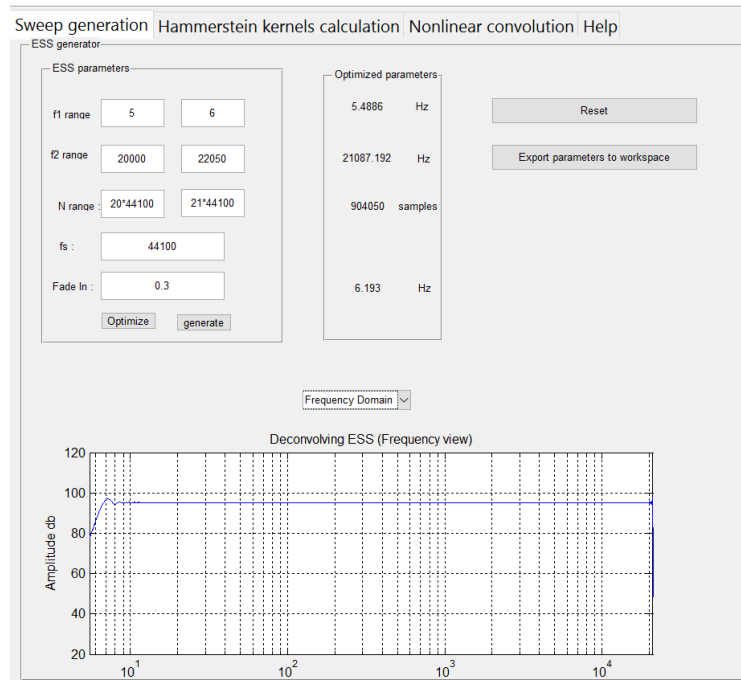


Figure 2: ESS generation

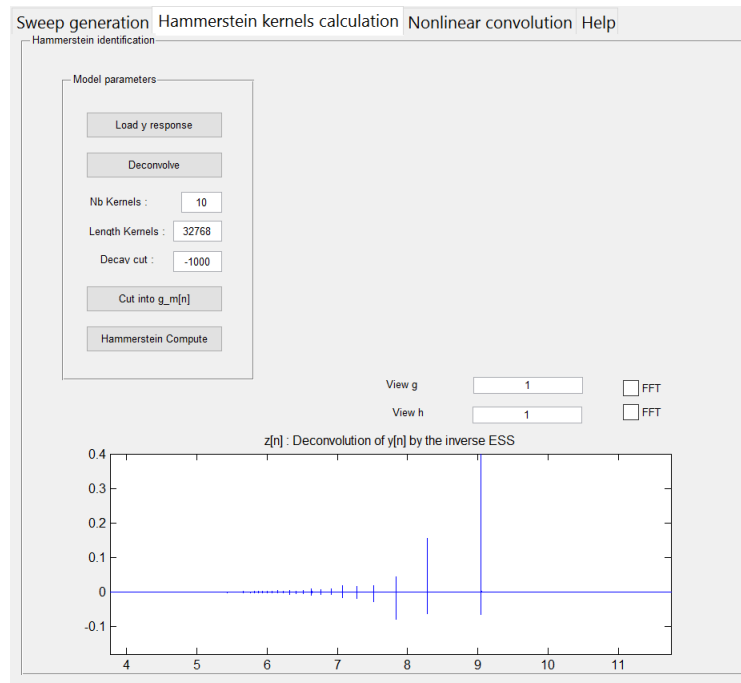


Figure 3:  $z[n]$  the convolution between  $y_{DUT}$  and the ESS inverse filter



Finally, using the computed Hammersteins kernels of the *Tubescreamer*, the emulation can be proceeded. In the *Nonlinear Convolution* tabs the input signal to emulate can be chosen. In this example , we have chosen the *ESS* itself because it sweeps over all frequencies. Clicking on the *Convolve* button launches the generation of the emulated signal (it could take a moment, depending on the length of the input signal). Figs. 4 and 5 present the comparison between the *ESS* through the *Tubescreamer* ( $y_1$ ) and the result of our emulation ( $y_2$ ) around the instantaneous frequencies 100 and 1000 respectively. The mean error (absolute difference between the two signals) is less than 0.7%, which makes the emulated signal almost identical to the original.

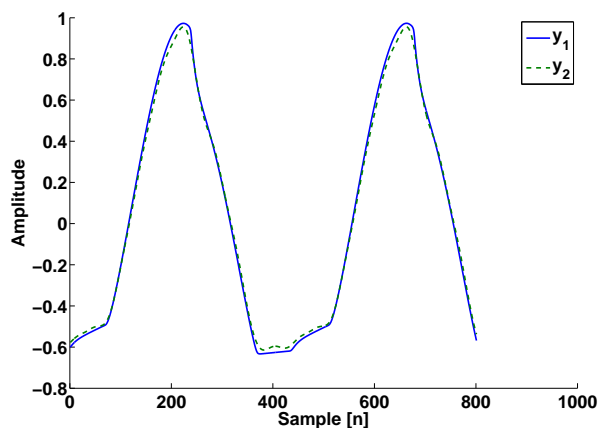


Figure 4: Comparison of the *ESS* through the *Tubescreamer*  $y_1$  and through the emulator  $y_2$  (zoom over the frequency  $f = 100\text{Hz}$ )

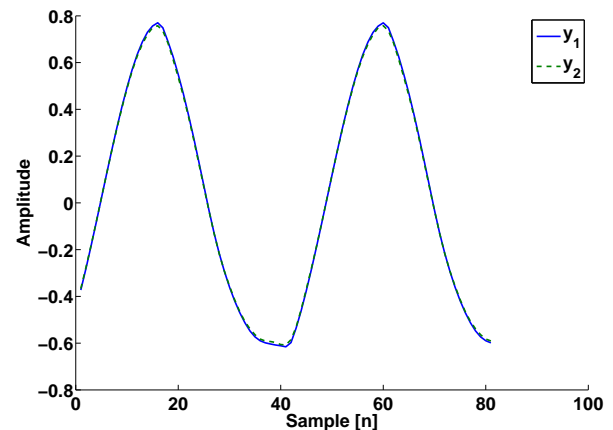


Figure 5: Comparison of the *ESS* through the *Tubescreamer*  $y_1$  and through the emulator  $y_2$  (zoom over the frequency  $f = 1000\text{Hz}$ )

## 5 Conclusions

An identification method of the cascade Hammerstein model by sine sweep has been elaborated in 2001 by Farina [6]. Since that time, the technique has been improved by several research teams. In [12], the computation of the Hammerstein kernels has been improved to arrive to an Optimized Hammerstein Kernels Identification by Sine Sweep (*OHKISS*) method. Based on this method, the toolbox presented in this paper has been created to ease the identification and the emulation of nonlinear systems. We have demonstrated its efficiency on a guitar distortion effect, the *Tubescreamer*. The resulting signal is close to the original, with less than 0.7 percent of mean error between the output *Tubescreamer* signal and its emulation.

## References

- [1] T. Schmitz, J.J. Embrechts, "Nonlinear Guitar Loudspeaker Simulation," presented at the *134th Convention of the Audio Engineering Society* (2013 May), convention e-Brief 96.

- 
- [2] M. Schetzen, *The Volterra & Wiener Theory of Nonlinear Systems* (John Wiley & Sons, 1980).
- [3] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches* Springer, (2007).
- [4] F. Kadlec, P. Lotton, A. Novak, & L. Simon, "A New Method for Identification of Nonlinear Systems Using Miso Model with Swept-Sine Technique: Application to Loudspeaker Analysis," presented at the *124th Convention of the Audio Engineering Society* (2008 May), convention paper 7441.
- [5] M. Rébillat, R. Hennequin, E. Corteel, & B. Katz, "Identification of Cascade of Hammerstein Models for the Description of Nonlinearities in Vibrating Devices," *Journal of Sound and Vibration*, Vol. 330, Issue 5, pp. 1018–1038 (2011 February).
- [6] A. Farina, A. Bellini, & E. Armelloni, "Nonlinear Convolution: A New Approach for the Auralization of Distorting Systems," presented at the *110th Convention of the Audio Engineering Society* (2001 May), convention paper 5359.
- [7] T. Schmitz, J.J. Embrechts, "Improvement in Nonlinear Guitar Loudspeaker Sound Reproduction," presented at the *21th International Conference on Systems, Signals and Image Processing* (2014 May), international conference on IEEE.
- [8] L. Tronchin, "The Emulation of Nonlinear Time-Invariant Audio Systems with Memory by Means of Volterra Series," *J. Audio Eng. Soc.*, Vol. 60, No. 12, (2012 December)
- [9] A. Novak, L. Simon, & P. Lotton, "Analysis, Synthesis, and Classification of Nonlinear Systems Using Synchronized Swept-Sine Method for Audio Effects," *Eurasip Journal on Advances in Signal Processing*, Vol. 2010:793816, (2010 July). <http://asp.eurasipjournals.com/content/2010/1/793816>
- [10] A. Novak, L. Simon, "Nonlinear System Identification Using Exponential Swept-Sine Signal," presented at the *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No 8, pp. 2220–2229, (2009 August).
- [11] T. Schmitz, Download *OHKISS toolbox*: <https://github.com/TSchmitzULG/OHKISS>
- [12] T. Schmitz, J.J. Embrechts, "Optimized Hammerstein Kernels Identification by Means of a Sine Sweep Technique Applied to Nonlinear Audio Devices Emulation" submitted for publication.
- [13] G. Palm, "On Representation and Approximation of Nonlinear Systems," *Journal Biological Cybernetics*, Vol. 34, Issue 1, pp. 49–52 (1979 September).
- [14] A. Farina, "Advancements in Impulse Response Measurements by Sine Sweeps," presented at the *122th Convention of Audio Engineering Society* (2007 May), convention paper 7121.
- [15] Scip Solver: <http://scip.zib.de/>
- [16] Tubescreamer Ts808: <http://www.analogman.com/tshist.htm>
-