Block-Oriented Modeling of Nonlinearities in Electro-Acoustic Transducers

Toningenieursprojekt

Kaspar Glattfelder

Supervisor: O.Univ.Prof. Mag.art. DI Dr.techn. Robert Höldrich

Graz, November 30, 2021

UNIVERSITÄT
FÜR MUSIK UND
DARSTELLENDE KUNST
GRAZ - AUSTRIA

institut für elektronische musik und akustik

Abstract

This project deals with the fundamentals of measuring, modelling, and simulating a nonlinear system using a cascade of Hammerstein models. The measurement is based on the exponential sine sweep which enables a separate extraction of the higher order impulse responses of a given system. A python script calculates the impulse response and extracts the kernels of the measurement, which are then exported as wave files. These can be used in combination with a very simple plugin to perform the simulation in any DAW that supports VST plugins. In addition, the patterns of the radiated sound from the simulation, compared to the real loudspeaker are analyzed and discussed. Furthermore, a listening trial is conducted to validate the algorithms performance on simulating the distortion.

Contents

1 Introduction

Electro-acoustic transducers, or simply speakers and microphones, are essentially omnipresent throughout everybody's life. The first speakers came up approximately 150 years ago and quickly gained popularity, especially now with devices such as smartphones, smart-speakers and Bluetooth-headsets taking over the market. These transducers can be analyzed and identified in their properties and behaviors to create models. The extracted models can be used to further refine the quality of the sound or to digitally simulate the identified speaker (e.g., the cabinet of a guitar amplifier with its distinctive sound). This identification processes have been done for decades and are very well studied areas of research.

One particular aspect of the transducer is its nonlinear behavior, that tends to be especially prominent when operating the speaker at high sound pressure levels (high displacement of the membrane). This distortion diminishes the sound quality and can create additional harmonic components that were not originally part of the signal. Although the total amount of the harmonic distortion can be quantified, it is not possible to further characterize the distortion with the conventional identification processes -they only are capable of capturing the linear behavior.

Brunet demonstrates existing methods of higher order system identification in his dissertation from 2014 [Bru14]. The first attempts took place in the 1980s based on the **"White Box Model"** and since then deliver predictions of mechanical behavior of the loudspeaker for low frequencies and small signals based on sine excitation of the system. The nonlinearities are expressed through Volterra kernels, which are limited to a low order ($K = 3$). Nowadays the white box approach is still used, but only delivers viable results at the lower end of the spectrum.

Next to the research on the "White Box Model" another approach evolved in the 1990 - the **"Black Box Model"** which leaves the mechanical behavior of the system aside and focuses more on a statistical model, again based on the Volterra model but this time in the frequency domain. Because of the very high complexity, also this model is limited to a very low order. (Again $K = 3$)

In 2000 Farina showed in [Far00], that using an exponentially swept sine as an excitation signal to identify a system, gives access to the linear responses but most importantly, to the individual harmonic nonlinear responses. Based on this research, Rebillat then showed in [RHCK10] how to obtain the model of the nonlinearities by using a cascade of Hammerstein models.

2 Inspection of Harmonic Distortion

The basic approach to identify the behavior of a loudspeaker i.e., its transfer function is to excite the loudspeaker, record the response of the input signal and then compare the signals with each other i.e., perform a deconvolution. There are many forms of excitation signals ranging from a single short impulse to periodic wide-band noise played over a longer time period. However, the most commonly chosen excitation methods are the MLS (Maximum-Length-Sequence) pseudo-random white noise and sine sweeps or chirps. Regardless of the chosen method, the desired output remains the transfer function, but the results differ in the case of a nonlinear system. When using an exponential sine sweep as an input signal, the system is excited with a sine tone moving from the lower to the upper end of the desired frequency range with the frequency increasing exponentially.

2.1 Measurement Using an Exponential Sweep

Equation 1 shows the formula to derive a sweep ranging from f_1 to f_2 with a duration of T. The starting and ending values of the function are defined as $f(0) = f_1$ and $f(T) = f_2$. The band of interest of the system under test is only defined in the frequency range $[f_1, f_2]$. This also means, that any further calculations and simulations are only valid in the measured frequency range.

$$x(t) = \sin\left[K\left(e^{\frac{t}{L}} - 1\right)\right] \tag{1}$$

where:

$$K = \frac{2\pi f_1 T}{\ln\left(\frac{f_2}{f_1}\right)} \qquad L = \frac{T}{\ln\left(\frac{f_2}{f_1}\right)} \tag{2}$$

f_1 = starting frequency
f_2 = ending frequency
T = sweep duration

Choosing the parameters of the sweep carefully is crucial to the result of the obtained impulse response. First and foremost, the phase of the signal depends on the starting and ending ending angular frequency ω_1, ω_2 and the duration T. Novak proposed a method in [nls15] called the sine-phase-matched sweep which will ensure, that each harmonic order impulse response will be phase matched with the linear impulse response. For the method to work, it is mandatory that the sine sweep is zero-phased not only at the beginning, but also at the end of each octave. This way, each harmonic-order impulse response will be phase matched with the linear IR. The generation of the sweep and the underlying theory is discussed thoroughly in Novak's work about Synchronized Swept Sine Theory from 2015 [nls15]. The ordinary sweep equation 1 is modified slightly to ensure a proper magnitude and phase estimation of the higher harmonic frequency responses. Also, the initial phase at time $t = 0$ will be zero degrees. The basic idea is to assign a desired sweep duration $\tilde{T}$ and then find the nearest rounded number of the rate of frequency L.

$$L = \frac{1}{f_1}\text{round}\left[\frac{\tilde{T} f_1}{\ln\left(\frac{f_2}{f_1}\right)}\right] \tag{3}$$

The duration T that ensures the synchronised sweep is then retrieved like the following:

$$T = L \ln\left(\frac{f_2}{f_1}\right) \tag{4}$$

Now the sweep can be generated using equation 5 without the -1 term and the updated duration T .

$$x(t) = \sin\left[K e^{\frac{t}{L}}\right] \tag{5}$$

Starting and ending frequencies can be picked arbitrarily. Using the parameters as shown, the sine sweep will start and, most importantly, end on a zero crossing which will prevent the spectrum from being polluted by an offset caused by the abrupt cutting off. Therefore, there is no need to apply a fade in or fade out. This further reduces the artificially induced signals in the spectrum since fades are essentially amplitude modulations which will affect the amplitude spectrum of the carrier signal.

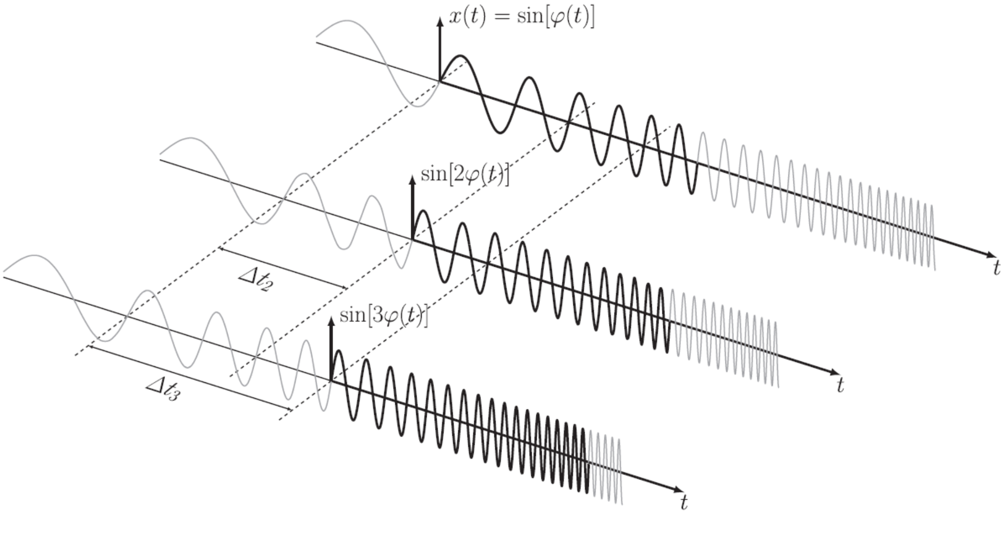


Figure 1 – Schematic representation of phase synced sines over three octaves [nls15]

2.2 Impulse Response

In order to obtain the impulse response, the system response signal $y(t)$ can either be divided with the excitation signal in the frequency domain or convolved with the inverse filtered input signal in the time domain which is mathematically the same operation. Both are viable options and relatively cheap to compute on modern processors. However, the inverse filter method implies a convolution of signals that can reach very high amounts of samples depending on the chosen duration and sample rate. In case of very long signals, the spectral division is computed faster.

2.2.1 Inverse Filter

In frequency varying signals, the energy at any frequency is proportional to the time spent on that specific frequency. Therefore, the spectrum of a linear sine sweep is flat. In the case of an exponentially increasing frequency of the input sweep, the amount of energy decays exponentially with rising frequency. This leads to an energy drop towards high frequencies in the spectrum of approximately 3dB per octave as seen in figure 2 according to equation 6. Although the inverse filter is time reversed, the spectrum looks the same regardless of the direction in the time domain since the time spent on each frequency is the same. To receive a flat transfer function and balance out the energy levels when convolving the input sweep with the inverse filter, a scaling term k needs to be applied to the inverse filter. [MSWH09]

$$E(j\omega) = \frac{L^2}{K}\frac{1}{L + j\omega} \tag{6}$$

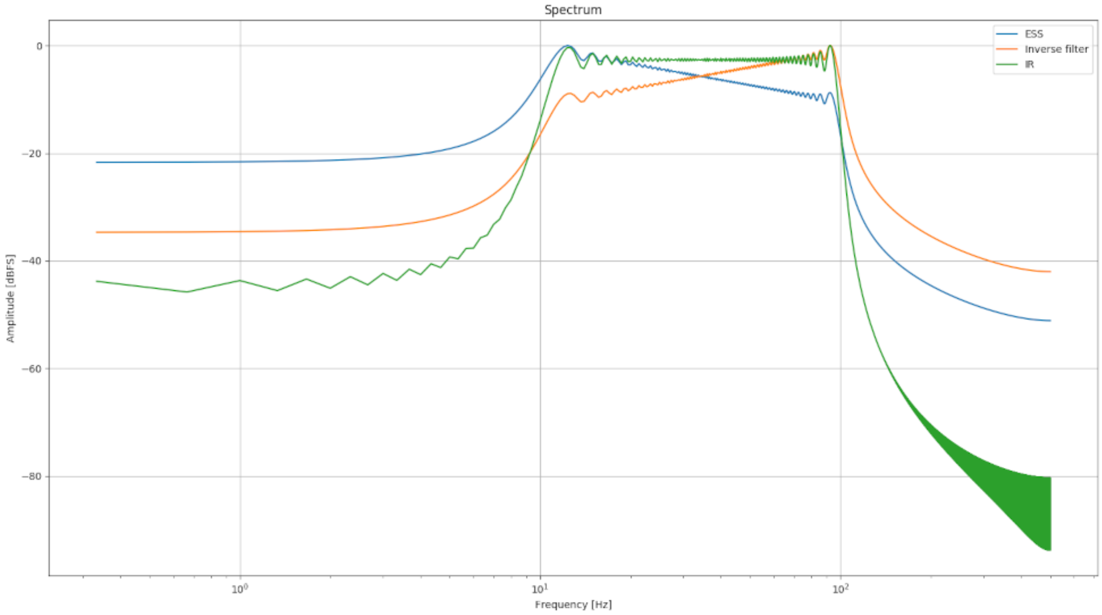


Figure 2 – Spectral comparison of exponential sweep, inverse filter and resulting impulse response [Joj]

Assuming an exponential input sweep according to equation 1 the inverse filter is calculated by scaling the amplitude of time reversed $x(t)$ by:

$$k(t) = e^{\frac{t}{T} \ln \frac{f_2}{f_1}}$$

Which will result in an exponentially decaying sweep:

$$\tilde{x}(t) = \frac{x(T-t)}{k(t)}$$

The impulse response $g(t)$ as shown in figure 3 can then be computed by convolving the inverse filter $\tilde{x}(t)$ with the system response $y(t)$ according to equation 7.

$$g(t) = y(t) \circledast \tilde{x}(t) \tag{7}$$

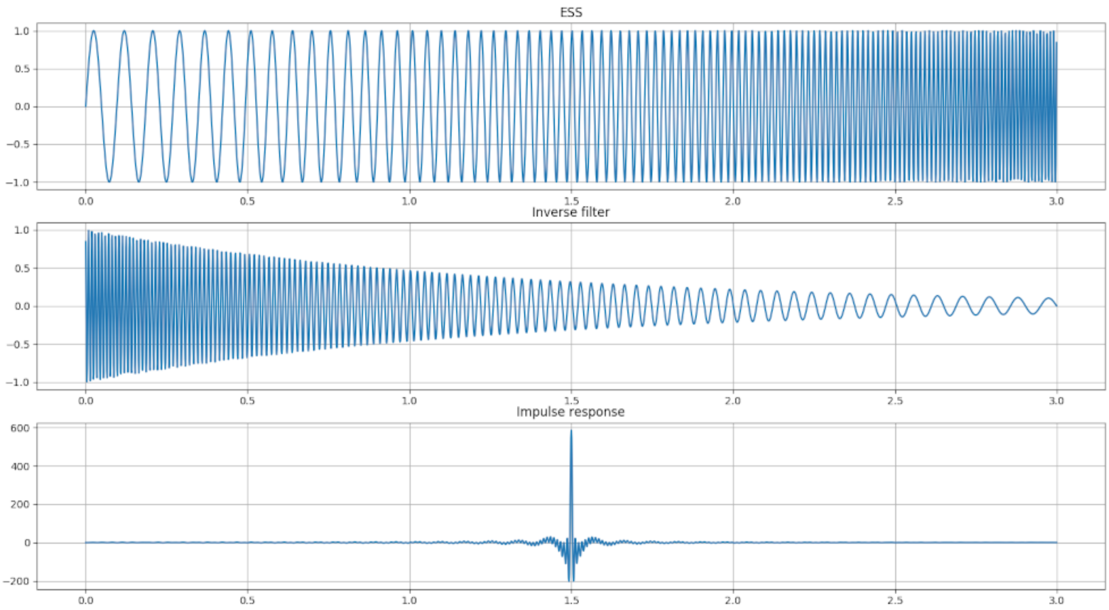


Figure 3 – Time signal comparison of exponential sweep, inverse filter and resulting impulse response [Joj]

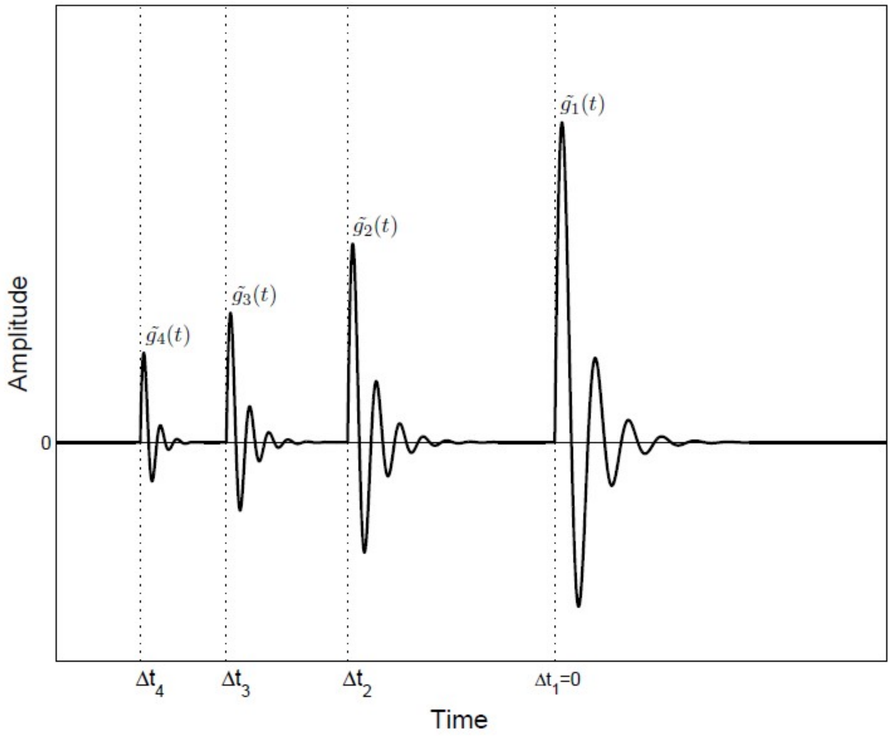


Figure 4 – Resulting impulse response of nonlinear system measured with exponential sweep method [RHCK10]

Another approach demonstrated by Novak in [nls15] performs the deconvolution with an analytical expression $\tilde{X}(f)$ for positive frequencies in the frequency domain.

$$g(t) = \mathcal{F}^{-1}\Big[\mathcal{F}\big[y(t)\big]\tilde{X}(f)\Big] \tag{8}$$

The analytical expression for the Fourier transform of the synchronised sine sweep is derived in great detail in [nls15] and results into:

$$X(f) = \frac{1}{2}\sqrt{\frac{L}{f}}\exp\left[j\omega L\Big(1 - \ln\Big(\frac{f}{f_1}\Big)\Big) - j\frac{\pi}{4}\right] \tag{9}$$

which results for $\tilde{X}(f)$ in:

$$\tilde{X}(f) = 2\sqrt{\frac{L}{f}}\exp\left[-j\omega L\Big(1 - \ln\Big(\frac{f}{f_1}\Big)\Big) - j\frac{\pi}{4}\right] \tag{10}$$

2.2.2 Higher Order Terms

The resulting impulse response as shown in figure 4 yields multiple peaks spread over the time axis when measuring a nonlinear system. The initial peak at $t = 0$ corresponds to the linear response of the system i.e., the ordinary transfer function of the measured system without distortion. Due to the nonlinear increase of the instantaneous frequency the distortion products are pushed to the left of the initial impulse response. This behavior makes the exponential sweep method prone to distortion but most importantly, provides a possibility to analyse the individual harmonic distortion products produced by the measured system. [Far00] In normal cases the distortion would be rejected by applying a window over the initial impulse response only and then transferring the resulting time signal to the frequency domain. The acquired transfer function describes the system completely and is enough to simulate a linear system. However, when dealing with nonlinear systems the distortion products on the left are time-shifted higher harmonic impulse responses that are crucial to the modelling and simulation process. When the deconvolution is performed on the distorted system, the resulting impulse responses are given by:

$$y(t) \circledast s(t) = \sum_{k=1}^{N} g_k(t + \Delta t_k) \tag{11}$$

The negative time shift of the k^{th} harmonic impulse responses can be computed according to equation 13. The acquired value also corresponds to the time it takes for the sweep to progress through k octaves. It can be seen that the time delay is directly proportional to the duration of the sweep T. The individual impulse responses must not overlap, therefore the sweep duration needs to be picked high enough for all impulse responses to decay in between the time delays Δt_k. This is especially important when measuring highly reverberant systems with long decay times.

$$k \cdot \frac{d}{dt}\left[\frac{\omega_1 T}{\ln\left(\frac{\omega_2}{\omega_1}\right)}\left(e^{\frac{t}{T}\ln(\frac{\omega_2}{\omega_1})} - 1\right)\right] = \frac{d}{dt}\left[\frac{\omega_1 T}{\ln\left(\frac{\omega_2}{\omega_1}\right)}\left(e^{\frac{t+\Delta t_k}{T}\ln(\frac{\omega_2}{\omega_1})} - 1\right)\right] \tag{12}$$

which is solved for Δt_k to:

$$\Delta t_k = \frac{T\ln(k)}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \qquad \forall k \in \mathbb{N} \tag{13}$$

The value of Δt_k for each order is constant and the impulse responses will occur on the very precise time lag before the linear response. However, since the delays are proportional to the logarithm of k, the time lags between get smaller with increasing order.

Note that Δt_k won't result in integers only and therefore, the advancements in time will include fractions of samples. Rounding the values to the next integer value of samples will result in severe deviations of the resulting simulation and must not be done. Rather, non-integer sample delays can be realized by exploiting the shift property of the Fourier transform. In practice, first the rounded time delay of Δt_k is applied and is then followed by a shift of the remaining non integer time delay t_r.

$$\mathcal{F}\big[g(t-t_r)\big] = G(f)e^{-j\omega t_r} \tag{14}$$

where

$t_r = \Delta t_k - \text{round}(\Delta t_k)$

3 Reproduction of Harmonic Distortion

The individual impulse responses of the system g_k are being separated by applying suitable windows of the same size. The time shift of the k^{th} order is computed by equation 13. The position of the windows should be slightly shifted to the left to avoid cutting off any part of the leading part of the response.

Figure 5 shows the complete schematic overview of extracting the kernels of a nonlinear system. The Device under Test (DUT) features multiple parallel branches consisting of a static non-linearity followed by a linear transfer function $h_k(t)$. This constellation is known as a cascade of Hammerstein models. The contained transfer functions $h_k(t)$ are the kernels which completely identify the model. The input signal is elevated to the k^{th} power and then convolved with the corresponding kernel. Afterwards the K parallel branches are summed up to simulate the system. Due to the elaborated choice of parameters demonstrated in chapter 2.1 the individual signals are summed up phase synced and therefore represent the actual waveform of the signal. If the phases are not synchronised the simulated waveform differs to a large degree leading to an inaccurate simulation.

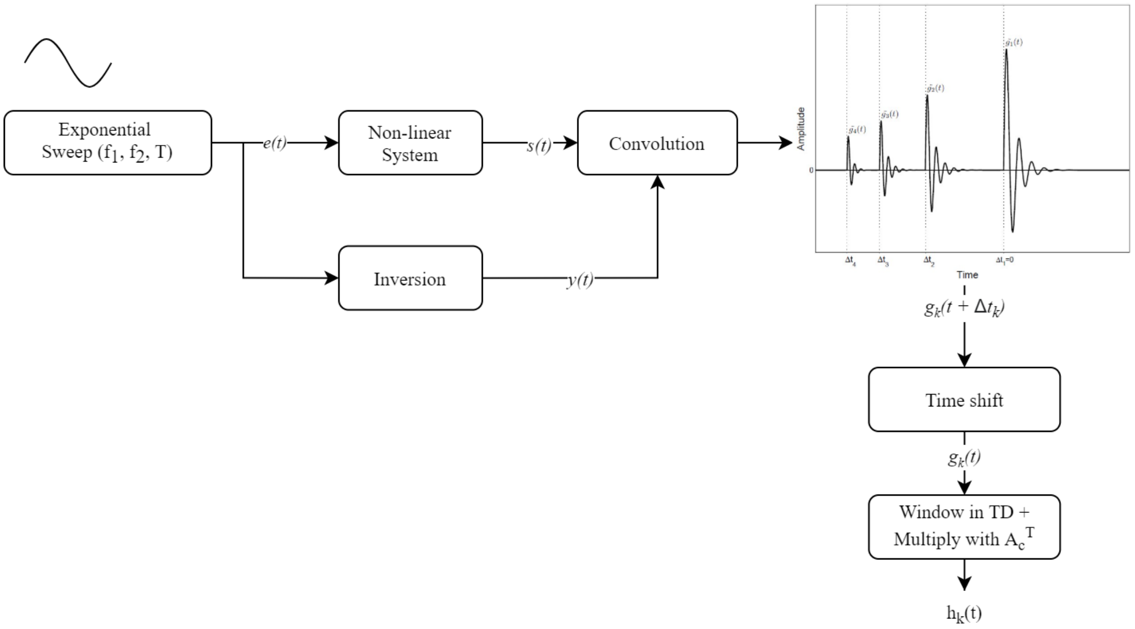


Figure 5 – Schematic overview of the identification process

$$s(t) = \sum_{k=1}^{K} h_k \circledast e^k(t) \tag{15}$$

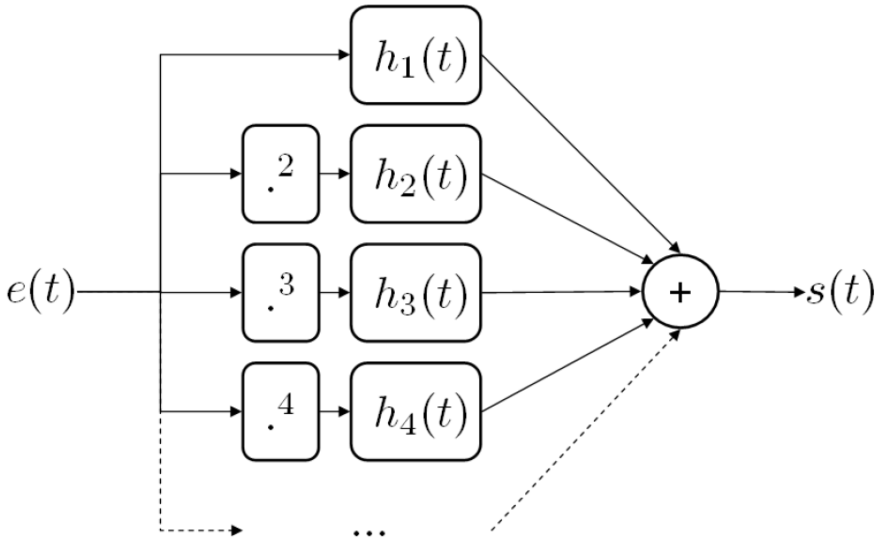


Figure 6 – Block diagram representation of a cascade of Hammerstein models [RHCK10]

3.1 Extracting the Kernels

The results of the measurement described in the previous section are not the actual impulse responses of the higher orders but instead, the diagonal terms of the Volterra kernels. In order to gain access to the impulse responses that are suitable to simulate using a Hammerstein model, the kernels have to be extracted first.

3.1.1 The Volterra Series

The Volterra series was first applied by Wiener when he attempted to analyse slightly nonlinear systems. Its idea is to represent the output, $y(t)$, of a nonlinear system as a functional series expansion of its input, $x(t)$.

$$y(t) = \sum_{k=0}^{K} y_k(t) \tag{16}$$

where

$$y_k(t) = \sum_{i_1=0}^{T} \sum_{i_2=0}^{T} \dots \sum_{i_k=0}^{T} h_k(i_1, \dots, i_k) x(t - i_1) \dots x(t - i_k) \tag{17}$$

Equation 16 shows the Volterra series where $y_k(t)$ is the Volterra operator which essentially performs nested convolutions up to the k^{th} integral. The k^{th}-order integral, $h_k(i_1, \dots, i_k)$ is called the Volterra kernel of the Volterra operator. [GB10]

The series makes use of memory, as the system output also contains products of previous sample values with different delays. This becomes more obvious when the series is written out, demonstrated by the following equation:

$$y(t) = \sum_{i_1=0}^{T} h_1(i_1)x(t-i_1) + \sum_{i_1=0}^{T}\sum_{i_2=0}^{T} h_2(i_1,i_2)x(t-i_1)x(t-i_2) + \dots \tag{18}$$

A method to quantify the practical applicability of a Volterra series model, is to look at the number of model parameters (independent kernel values) which is defined as:

$$N_{par} = \frac{(T+K+1)!}{(T+1)! \cdot K!} \tag{19}$$

The Volterra model is linear in its parameters and could be approximated by a least squares algorithm. However, it is only a valid method for either low nonlinearities (small K) or short memories (small T). A more universal approach is to model the system in a block-like structure. Thereby, the linear and nonlinear parts can be represented in many forms. To demonstrate the relationship to the Volterra series, the linear part is represented as a FIR filter and the nonlinear block is represented as a polynomial [GB10, p.386] . Written as a Wiener system the equation is expressed like the following:

$$y(t) = \sum_{k=1}^{K} c(k)\left(\sum_{i=0}^{T} h(i)x(t-i)\right)^{k} \tag{20}$$

where

$h(i)$ = elements of the FIR filter
$c(k)$ = polynomial coefficients of the non-linearity

Through comparing equations 20 and 17 the Volterra kernels of a Wiener system are given by:

$$h_k(i_1, i_2, \dots, i_n) = c(k)h(i_1)h(i_2)\dots h(i_k) \tag{21}$$

3.1.2 The Hammerstein Model

In most common systems, nonlinearities occur at the very beginning and are substantially memoryless. After the initial distortion happened, the signal will pass on through linear systems that are characterized by temporal effects which means that the system has memory. This is also the case when looking at the setup of a speaker in a room that yields nonlinear behavior. The distortions occur in the electro-mechanical transducer and are then radiated into the room where they propagate though the air and reflect off walls.

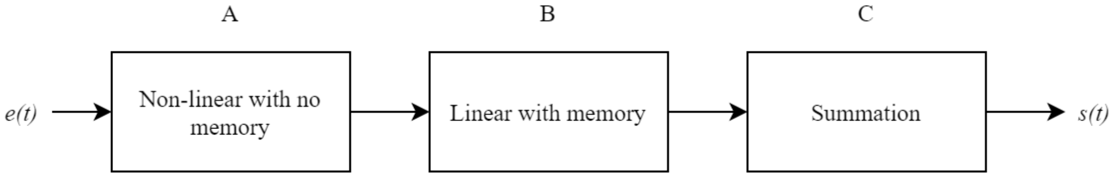


Figure 7 – Hammerstein model

By taking equation 20 and reversing the order of the linear and nonlinear block, a Hammerstein system is produced.

$$y(t) = \sum_{i=0}^{T} h(i) \sum_{k=1}^{K} c(k) x(t-i)^k \tag{22}$$

And the Volterra kernels are given by:

$$h_k(i_1, i_2, \ldots, i_k) = c(k) h(i_1) \delta(i_2 - i_1) \delta(i_3 - i_1) \ldots \delta(i_k - i_1) \tag{23}$$

where $\delta(i-j)$ is the Kronecker delta. The Volterra kernels of a Hammerstein system will therefore only be nonzero on the main diagonal.

When putting the Hammerstein system into the context of the measurement with an exponential sweep, sinusoidal terms with higher orders $\sin^k(\omega t)$ and different angular velocities $\sin(k\omega t)$ need to be related with each other. In practice, this means that the measured results g_k need to be related to the simplified Volterra kernels h_k.

$$s(t) = g_1 \circledast \sin(\omega t) + g_2 \circledast \sin(2\omega t) + g_3 \circledast \sin(3\omega t) + \ldots \tag{24}$$

$$= h_1 \circledast \sin(\omega t) + h_2 \circledast \sin^2(\omega t) + h_3 \circledast \sin^3(\omega t) + \ldots \tag{25}$$

Sinusoidal terms can be decomposed using trigonometry. The following equations demonstrate that sinusoidal signals raised to higher powers also contain terms of different angular frequencies. It is also notable that even orders contain continuous terms, which are noticeable as a DC offset, while odd orders only contain sinusoidal terms. The continuous analytic terms as expressed here could be subtracted from sinusoidal input signals after they are raised to the higher power to entirely remove the DC offset. However, in practice the input can be an arbitrary signal and the analytic solution is not valid anymore. A suitable approach to remove unwanted frequencies outside of the band of interest is to apply a band pass filter after the simulation.

$$\sin^2(\omega t) = \frac{1}{2} - \frac{1}{2}\cos(2\omega t) \tag{26}$$

$$\sin^3(\omega t) = \frac{3}{4}\sin(\omega t) - \frac{1}{4}\cos(3\omega t) \tag{27}$$

$$\sin^4(\omega t) = \ldots \tag{28}$$

3.1.3 Hammerstein Transfer Matrix

With the equations above, a linear equation system is obtained which can be solved easily in the frequency domain. The Fourier transforms of equations 25 are defined as the following:

$$\begin{aligned} S(f) &= G_1(f)X(f) + G_2(f)X(\tfrac{f}{2}) + G_3(f)X(\tfrac{f}{3}) + \dots \\ &= \big[H_1 + \tfrac{3}{4}H_3 + \tfrac{5}{8}H_5\big]X(f) + j\big[-\tfrac{1}{2}H_2 - \tfrac{1}{2}H_4\big]X(\tfrac{f}{2}) + \dots \end{aligned}$$

So, the relation of the higher harmonic frequency responses and the kernels of the Hammerstein model is derived.

$$\begin{aligned} G_1(f) &= H_1 + \tfrac{3}{4}H_3 + \tfrac{5}{8}H_5 \\ G_2(f) &= -j\frac{1}{2}\big[H_2 + H_4\big] \\ G_3(f) &= -\frac{1}{4}H_3 - \frac{5}{16}H_5 \\ &\vdots \end{aligned}$$

We can easily solve it, obtaining the required Volterra kernels as a function of the measured higher harmonic frequency responses:

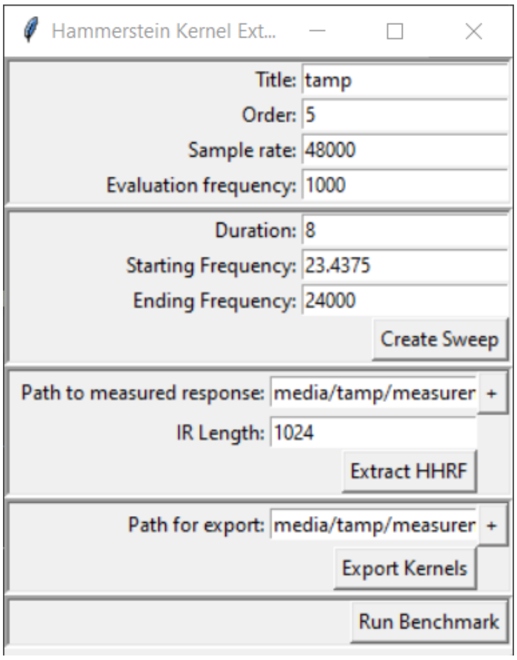
$$\begin{aligned} H_1(f) &= G_1 + 3G_3 + 5H_5 \\ H_2(f) &= j2H_2 + j8H_4 \\ H_3(f) &= -4H_3 - 20H_5 \\ &\vdots \end{aligned}$$

The higher harmonic frequency responses G_k can be multiplied with the equations above, written in matrix form, using the dot product to acquire the kernels of the system. The simulation is done by performing convolutions of the kernels with the input signal risen to the corresponding power and then summing the results as shown in equation 11.

$$\begin{pmatrix} H_1(f) \\ H_2(f) \\ \vdots \\ H_K(f) \end{pmatrix} = \mathbf{A} \begin{pmatrix} G_1(f) \\ G_2(f) \\ \vdots \\ G_K(f) \end{pmatrix} \tag{29}$$

4 Realisation as Python Script and Reaper Patch

The setup of the simulation consists of a python script that models the system, and a reaper patch that performs the real time simulation. Prior to the modeling process, the system needs to be measured which cannot be done with common measuring software like "Room Eq Wizard" or ARTA since they automatically strip away the distortion products on the left when exporting the impulse response. Therefore, the python script generates the sweep which can be used to measure the system i.e., play and record with any given DAW. This also ensures that the parameters of the sweep are picked accordingly.



Title: tamp
Order: 5
Sample rate: 48000
Evaluation frequency: 1000
Duration: 8
Starting Frequency: 23.4375
Ending Frequency: 24000
Path to measured response: media/tamp/measurer +
IR Length: 1024
Path for export: media/tamp/measurer +

Figure 8 – User interface of the python application

4.1 Sweep Generation

An exponential sweep is generated using the chirp function from the signal package in python and afterwards exported as a wave file. To create a sine instead of a cosine function, a phase offset of $-90\,\mathrm{deg}$ needs to be applied to the chirp function parameters. The duration of the sweep T is determined according to equation 4. A sample rate of 192 kHz prevents higher order aliasing artefacts from interfering with the band of interest $[f_1, f_2]$ in the simulation. The exact parameters used in the sweep simulation are listed in table 1.

Sample rate	$f_s = 192\,\text{kHz}$
Starting frequency	$f_1 = 23.4375\,\text{Hz}$
Ending frequency	$f_2 = 24\,\text{kHz}$
Sweep duration	$T = 8\,\text{s}$

Table 1 – Parameters used for sweep generation in the python script

4.2 Separation of Impulse Responses

The higher order impulse responses are separated using a right-side cosine tapered window shifted to the right in order to avoid cutting of any preceding part of the IR. The individual positions of the IR's are calculated using equation 13. Figure 9 makes it visible, that the harmonics are essentially scaled down versions of the linear response.

The amount of distortion can be quantifies using the Total Harmonic Distortion (THD) which is the square root of the ratio of the power contained in the harmonics to the power containted in the fundamental. Similarly, the Harmonic Distortion (HD) is defined the same as the THD but only compared to one harmonic.

THD and HD for a sinusoidal input signal $x(t) = X\cos(2\pi f t)$ in a cascade of Hammerstein models identified at amplitude X_0 can be expressed as the following:

$$\text{HD}_k(X, f) = \left|\frac{\Gamma_k(X, kf)}{\Gamma_{\text{tot}}(X, f)}\right| \tag{30}$$

$$\text{THD}(X, f) = \sqrt{\sum_{k=2}^{K}\left[\frac{\Gamma_k(X, kf)}{\Gamma_{\text{tot}}(X, f)}\right]^2} \tag{31}$$

with

$$\Gamma_{\text{tot}}(X, f) = \sqrt{\sum_{k=1}^{K}\Gamma_k(X, kf)^2} \tag{32}$$

and

$$\Gamma_k(X, f) = \sum_{i=1}^{K}\left(\frac{X}{X_0}\right)^i C(i, k) H_i(f) \tag{33}$$

The direct computation of THD/HD as shown here is possible due to the knowledge of the kernels in the frequency range $[f1, f2]$. H refers to the kernels in the frequency domain and C to the Chebychev matrix.

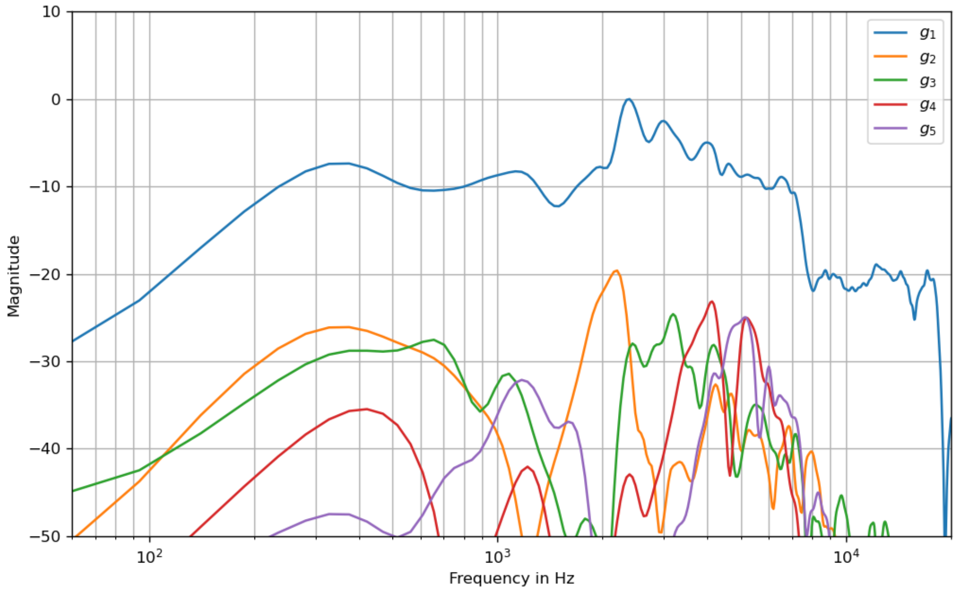


Figure 9 – The first 5 separated higher harmonic frequency responses g_k of the test system

4.3 Kernels

To receive the kernel of the cascade of Hammerstein models, the acquired impulse responses are multiplied with the complex valued inverse Chebychev matrix $\mathbf{A}_c$ using the dot product. The dimensions of $\mathbf{A}_c$ are not limited as the matrix is computed procedural within the run time of the script. The acquired matrix $\mathbf{H}$ holds the kernels in the columns and is then exported as a multi-channel wave file to be used in the MCFX 8 channel convolver plugin.

4.4 Simulation in Reaper

A real-time simulation of the modelled system including all its harmonic products can be computational demanding as it involves multiple convolutions at once. When running into problems, one might consider reducing the sample rate or similar measures to reduce the workload of the DSP. However, simulations with 8 convolutions running at 192 kHz are possible with modern computers and suitable audio interfaces. Table 2 gives an overview of the used specifications in the test setup.

4.4.1 Simulation Chain

Modelling the given system in form of a cascade of Hammerstein models opens the possibility of taking a similar approach in a simulation chain in the environment of a

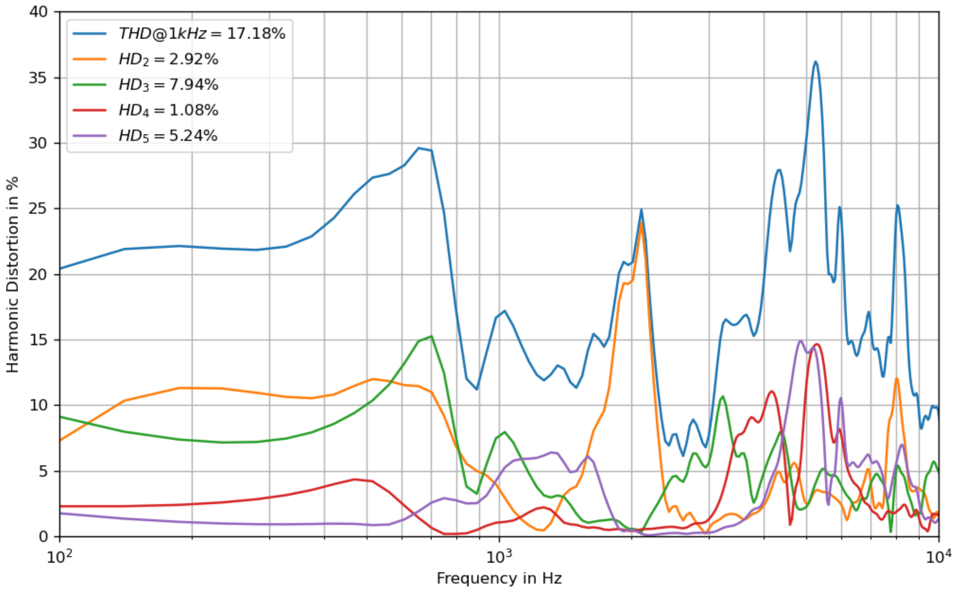


Figure 10 – Total and individual harmonic distortion of the test system

Digital Audio Workstation (DAW). Thereby, the input signal is placed in a track and then processed through two plugins which resemble the nonlinear part, followed by the linear part according to figure 7. A simple VST plugin "Harmonic Generator" has been created to produce the harmonics of the input signal. The first channel of the track serves as input and is processed according to equation 34. A knob steers the amplitude α of the harmonics which directly corresponds to the intensity of the distortion. α is set to 1 by default, which resembles the same distortion as measured.

$$y[n] = g \sum_{k=1}^{K} \big(\alpha x_1[n]\big)^k \quad \{\alpha \in \mathbb{R} : 0 < \alpha < 2\} \tag{34}$$

With higher orders, the resulting output signal can quickly get out of hand since the channels take the input to the power of up to 8. To compensate the exponential growth of the amplitude, the output of the combined channels is weighted with the coefficient g to avoid clipping.

$$g = \frac{K}{\sum_{k=1}^{K} \alpha^k} \tag{35}$$

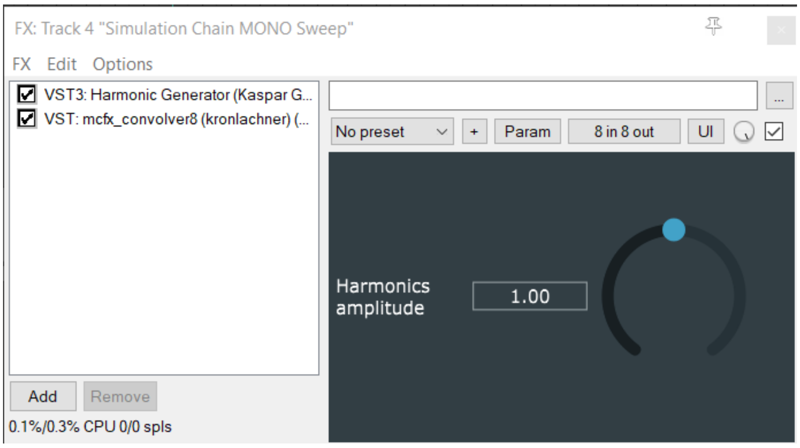
The gain g is calculated by setting the energy level of the measured distortion i.e., $\alpha = 1$ relative to the energy level of the desired harmonic amplification. The input $x_1[n]$ is assumed to be sinusoidal. To compensate the energy levels properly regardless of the signal form, the root mean square of the input clip would need to be calculated in advance. This can be a tricky task in DAWs since the VST plugin cannot analyze the

Operating System		Windows 10
Processor	Base Clock	2.3 GHz
	CPU Cores	4
	Threads	8
Memory	Installed	16 GB
	Base Clock	2400 MHz
	Type	DDR4
Audio Interface		RME BabyFace Pro

Table 2 – Technical specifications of test system

whole clip prior to playback. In practice the compensation like demonstrated in equation 35 delivers suitable results and is cheap to compute.

The individual harmonics k are then routed channel wise in the MCFX convolver. The configuration file is created in the modeling process of the python script and only needs to be imported into the VST Plugin. As soon as the exported kernels are loaded, each individual harmonic is convolved with the corresponding impulse response. In order to properly play back the resulting simulation, all channels of the track need to be routed into another track to sum everything up as mono. Otherwise only 2 of the N channels are sent to the master.



FX: Track 4 "Simulation Chain MONO Sweep"

☑ VST3: Harmonic Generator (Kaspar G...
☑ VST: mcfx_convolver8 (kronlachner) (...

Harmonics amplitude 1.00

0.1%/0.3% CPU 0/0 spls

Figure 11 – Interface of the Harmonic Generator Plugin

4.5 Results

The higher order terms are typically lower in amplitude than the linear term, which also resembles the perceived effect of the linear term being the base tone and the higher order terms the harmonics of the tone. In practice, most nonlinear systems tend to emphasize the odd terms over the even terms i.e., the 3^{rd} and 5^{th} order harmonics will be responsible for most of the distortion in an ordinary setup. This becomes even more visible in the spectrogram (Figure 12). The graph shows the progression of the sweep and its harmonics over the frequency band of interest in the specified duration. The y-axis is logarithmic, which is why the components are displayed straight. Due to the limited order, the script is only capable of simulating K orders. In most applications, this is enough to acquire an accurate simulation, although it will lead to slight deviations in the higher frequencies. The graph on the left shows a multitude of harmonics in the top left corner that wield very low energy levels but still contribute to the resulting spectrum when added up.

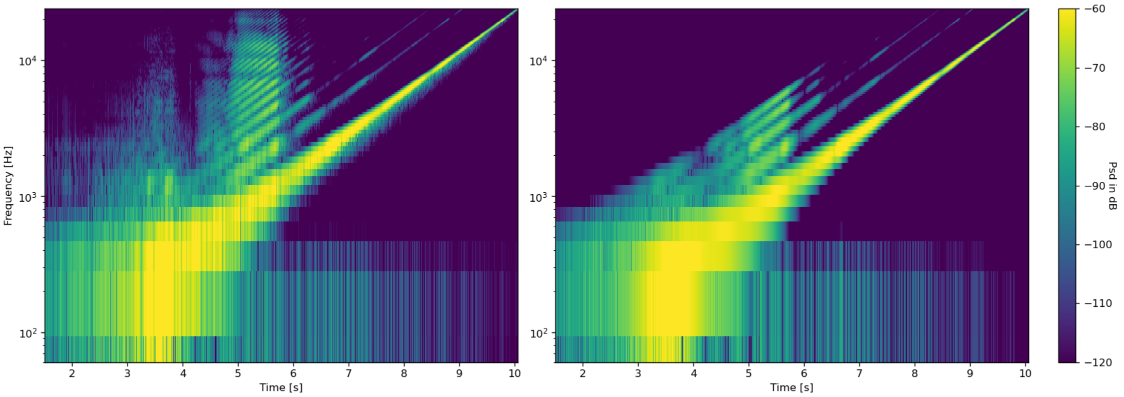


Figure 12 – Comparison of exponential sweep spectrograms with frequency displayed on logarithmic axis: left shows the recorded sweep, right the simulation with 5 harmonics

Figure 13 shows the waveform of each simulated signal, evaluated for all orders at 1 kHz. Again, the odd orders are most relevant to the resulting signal. Comparing the measured waveform with the simulated waveform in figure 15 shows a strong resemblance. The time delay is caused by the convolution operations but does not affect the result negatively as the individual components are all delayed the same.

Figure 14 shows the estimate power spectral density of the recorded and the simulated sweep in comparison. Both lines are derived using welch's method. The response at frequencies lower than 400 Hz cannot be considered accurate as the sweep is faded in up to 400 Hz to prevent the loudspeaker from permanent damage. Also, the frequency response of the simulation is approximately 2 dB lower than the target response in this frequency range which is due to the windowing applied on the extracted HHRFs for the simulation. However, the algorithm performs well with a mean squared error value of 1.84 summed over the entire band of interest.

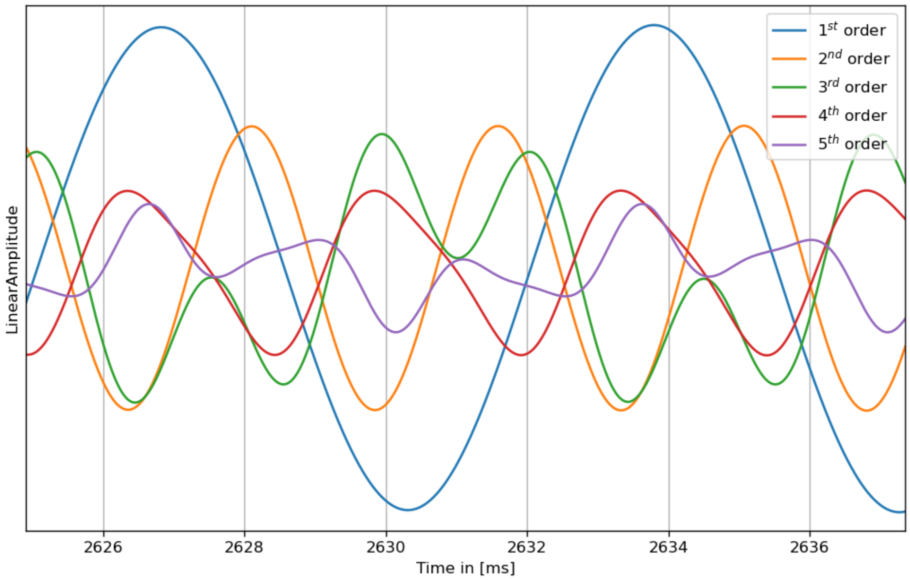


Figure 13 – Waveform of simulated signal, evaluated after each ascending order at 1 kHz

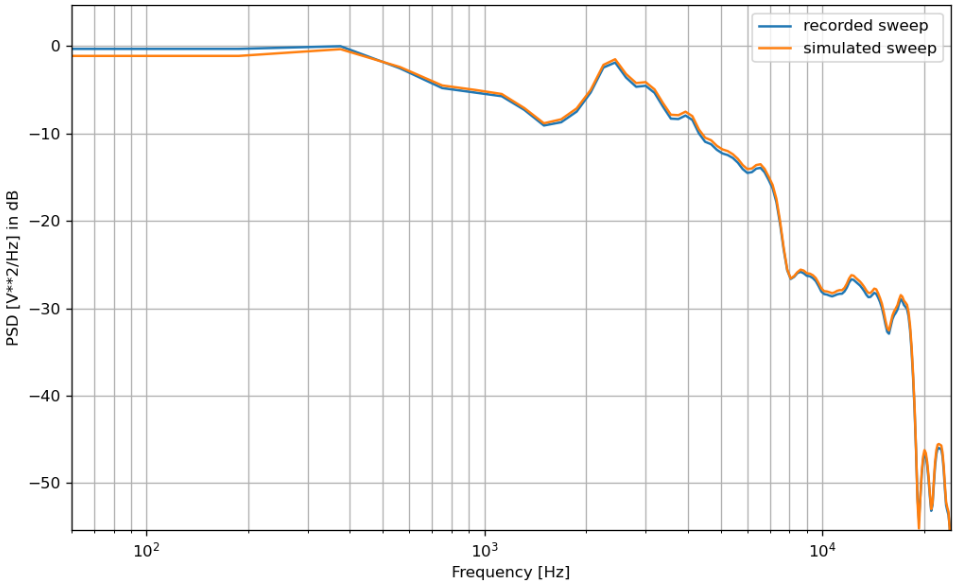


Figure 14 – Estimate power spectral density of recorded and simulated sweep in comparison

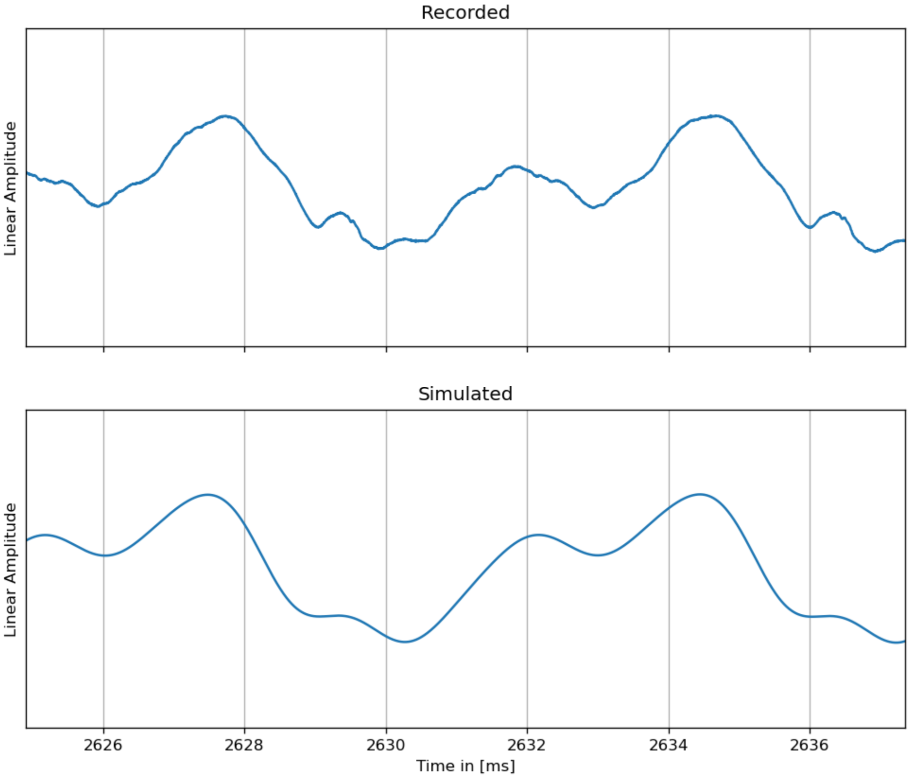


Figure 15 – Comparison of measured, and simulated waveform at 1 kHz

5 Comparison of Directivity Patterns

The directivity patterns of an ordinary 2-way HiFi speaker were measured and compared using the Double Circle Microphone Array (DCMA) at the Institute of Electronic Music and Acoustics. An anechoic chamber prevents reflections from polluting the measurement. The setup consists of 31 microphones distributed along the vertical circle at 1m distance and a remote-controlled turntable (Outline ET 250-3D) in the center of the circle. The speaker is placed on a adjustable loudspeaker stand on the turntable, and is then rotated along its axis for $350\,\mathrm{deg}$ with a step size of $10\,\mathrm{deg}$. The measurement process is controlled by a PureData patch which starts/stops the playback and recording tasks, as well as the position of the turntable. Figure 16 shows the schematic setup for measuring human voice directivity patterns. A detailed explanation of the construction and the signal processing of the Double Circle Microphone Array can be found in [bfr18].

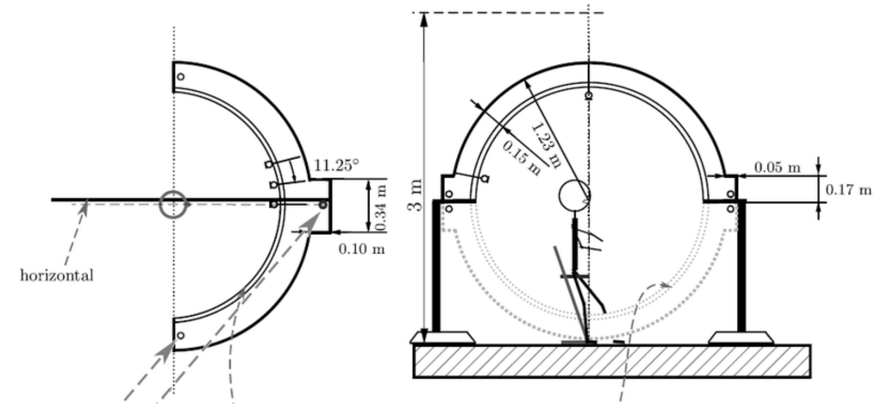


Figure 16 – Double Circle Microphone Array. Schematic of the measurement setup with two wooden circular rings (thickness: 21 mm) [bfr18]

5.1 Measurement

The idea of the measurement was to compare naturally radiated higher harmonics (distortion) with the simulated ones produced by the Hammerstein model. Therefore, the speaker was excited with very high amplitude first, in order to create sufficient distortion. This was realized by a "t.amp S-150 MK II" amplifier with a power of 150 W. The gain level was increased successively to avoid terminal damage of the speaker. However, ultimately the gain was set to the maximum of the amplifier. Very low frequencies $<$ 50 Hz were displacing the membrane extensively harder then higher ones. Therefore, a fade in up to 100 Hz was applied to the input sweep to ensure a more homogeneous distortion over the frequency spectrum.

Although the chamber is anechoic, low level reflections are still received at the microphones. The measurement should be free of reflections occurring inside of the measuring chamber to ensure a proper visualisation of the directivity patterns. The time delay of the reflections can be calculated quite easily when looking at the geometric properties of the setup and the chamber. Reflections arrive anywhere in the range from 30 to 300 samples of time delay after the direct sound arrives. Therefore, a right side cosine tapered window with a length of 140 samples is applied to neglect most of the reflections while still maintaining a suitable IR length (Figure 18).

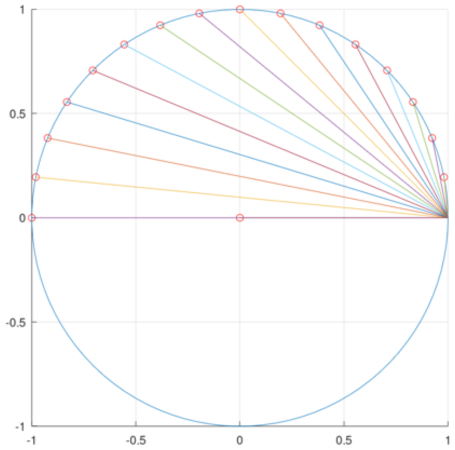


Figure 17 – Possible propagation paths of reflections on the DCMA scaffolding. The speaker emits sound at position (0, 0), the microphones are located along the ring. The colored lines represent the possible paths from the horizontal microphone (1, 0) to all others.

After measuring the distorted response of the speaker operating at very high-power levels, the response was estimated using the Hammerstein model. The signal of the microphone located directly in front of the speaker was used as the response (i.e., $0\,\mathrm{deg}$ horizontally and vertically), and the same sweep used in the measurement before served as the input signal for the simulation. The simulations were performed multiple times with different amounts of orders $K = [5, 10, 20]$. Afterwards, the gain of the amplifier was reduced to a suitable level to operate free of distortion and the simulated sweeps were measured in the Double Circle Microphone Array.

5.1.1 Speaker

The Device under Test is a basic Philips Consumer HiFi 2-way, Bass Reflex Speaker System. The crossover frequency of the speaker is determined at approximately 4 kHz. Further technical details are provided in table 3.

5.2 Results

The results are visualized in form of a balloon plot. Hereby the sound pressure levels are multiplied by an inverse matrix holding spherical harmonics coefficients for each measurement point (see [bfr18]). As the microphone ring is located at a distance of

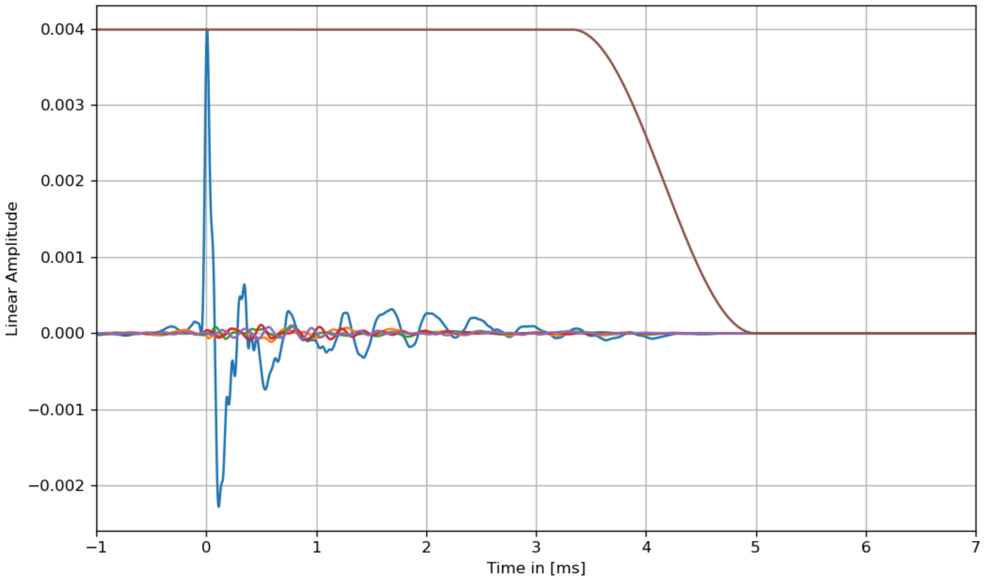


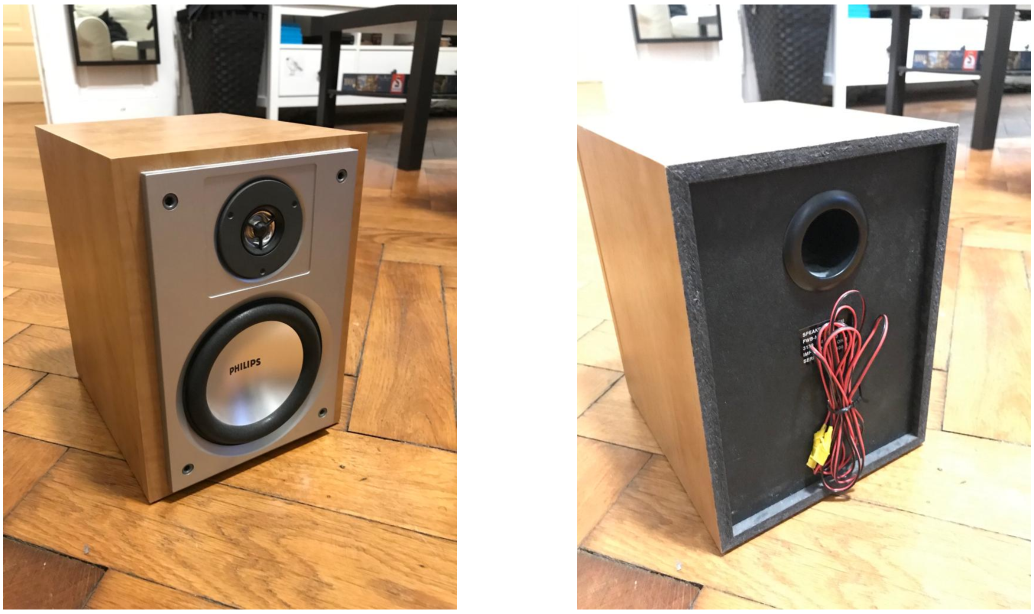
Figure 18 – Windowed impulse responses of the test system in time domain

System	2 way, Bass Reflex Speaker System
Impedanz	6Ω
Woofer	133,35mm
Tweeter	25,4mm
Dimensions (W x H x D)	180 x 244 x 215 (mm)

Table 3 – Technical specifications of the speaker under test

1m and the speaker itself reflects and bends propagating sound waves, the directivity pattern appears unsteady with lots of side lobes and layers in the plot. No smoothing is applied in order to make the differences between the real and simulated plots visible. Figure 20 shows the 3^{rd} harmonic that is emitted by the speaker. The left side shows the harmonic that is produced by the speaker itself when operating outside of its linear range. The right side shows the directivity pattern of the simulated distortion using the estimated nonlinearities from the Hammerstein kernels. The comparison shows a clear deviation between the patterns as the real harmonic is focused more on the lower part of the sphere whereas the simulation is directed straight in front at $0\,\deg$. This behavior is explained by the function of the 2-way speaker system. The real distortion occurs in the membrane of the woofer as it is displaced severely and therefore creating higher order harmonics. However, in the simulation, the simulated distortion is already contained in the input signal of the speaker and therefore routed through the crossover to the tweeter. The original directivity patterns of the higher harmonics can therefore not be recreated.

To visualize the discrepancy in the directivity patterns over the frequency band of interest, the energy on the cone surface on axis with opening angle of $20\,\deg$ is shown relative



(a) Front (b) Rear

Figure 19 – Test speaker

to the entire sphere surface. Thereby the spherical harmonic components are summed over the surface and multiplied with a map that resembles a cone with opening angle of $20\,\mathrm{deg}$. In the two-dimensional matrix Y_{sh} with dimensions N_x and N_y the map is defined as the following:

$$M(x,y) = \begin{cases} 1, & \text{if } (N_x/2 - x)^2 + (N_y/2 - y)^2 < r^2 \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

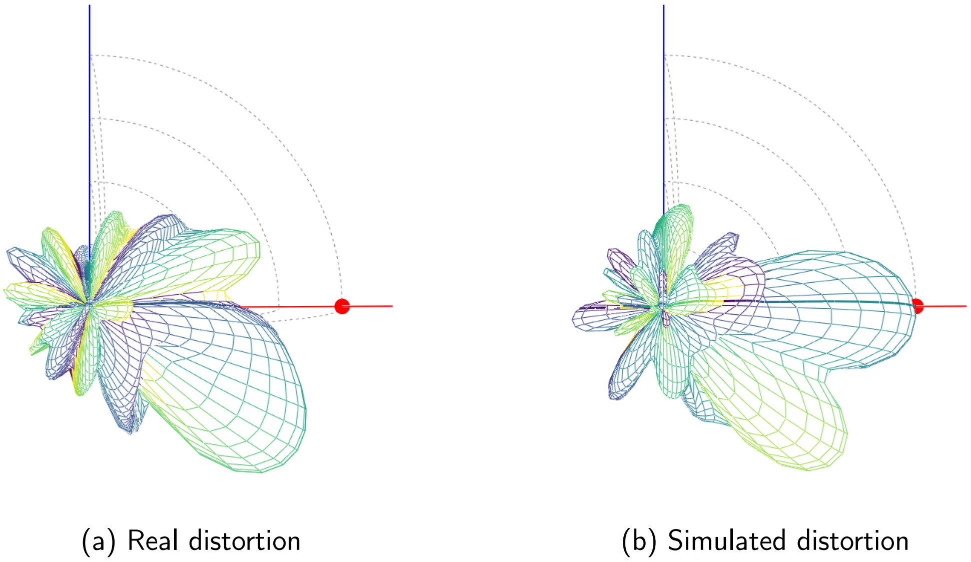
with:

$$r = \frac{N_x}{360} \cdot 20\,\mathrm{deg} \tag{37}$$

The ratio is then displayed in dB

$$\gamma_i[n] = 10 \cdot \log \frac{Y_{cone}}{Y_{sphere}} = 10 \cdot \log \frac{\sum_{x=0}^{N_x} \sum_{y=0}^{N_y} Y_{sh} \cdot M}{\sum_{x=0}^{N_x} \sum_{y=0}^{N_y} Y_{sh}} \tag{38}$$

Figures 21 to 23 show the contained energy summed up over the surface of the cone on axis compared to the whole sphere. One can see that the real and simulated linear responses i.e., 1^{st} harmonic align relatively good which means that the directivity patterns remain similar over the band of interest. This behavior is expected, as the linear response should remain the same independent of the distortion products in the signal. However, the 2^{nd} and 3^{rd} order harmonic align in the lower frequencies but start to deviate especially in the vicinity of frequencies in the range from 3 kHz to 5 kHz which meets the expectations.



(a) Real distortion

(b) Simulated distortion

Figure 20 – Directivity balloon plot of the 3^{rd} harmonic emitted by the speaker at 4.6 kHz

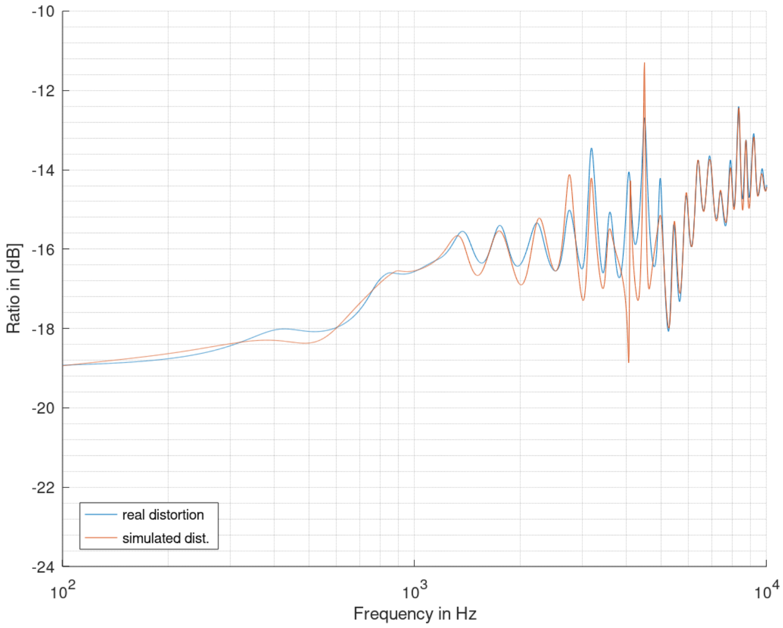


Figure 21 – Comparison of energy contained in cone surface on axis with opening angle of $20\,\mathrm{deg}$ to entire sphere surface of 1^{st} harmonic.

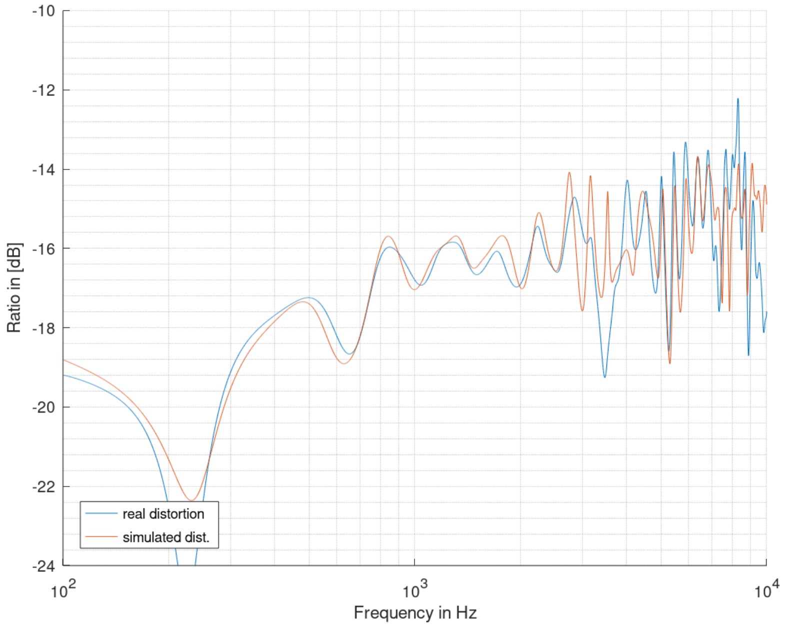


Figure 22 – Comparison of energy contained in cone surface on axis with opening angle of $20\,\mathrm{deg}$ to entire sphere surface of 2^{nd} harmonic.

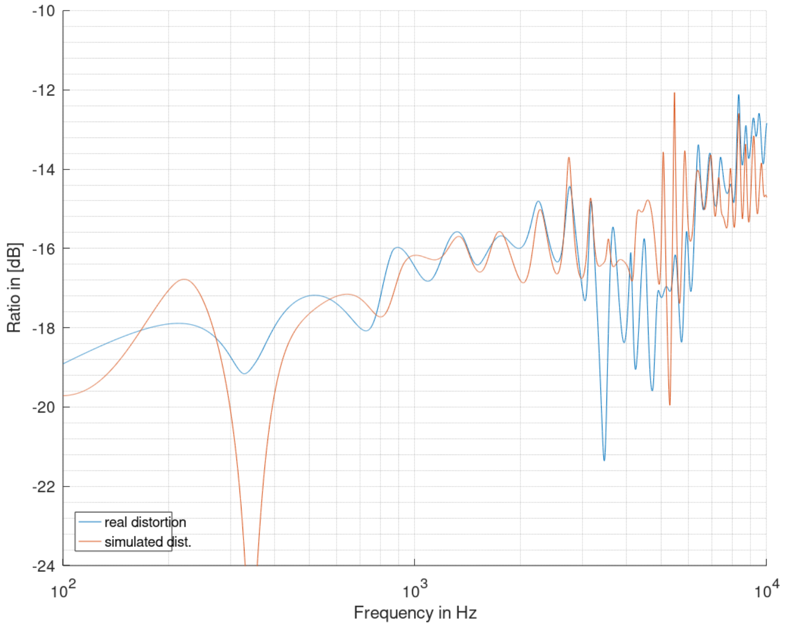


Figure 23 – Comparison of energy contained in cone surface on axis with opening angle of $20\,\mathrm{deg}$ to entire sphere surface of 3^{rd} harmonic.

6 Listening Test

In the course of the project a listening test has been conducted to validate the simulation i.e., see how well the simulation resembles the original distorted signal. Due to restrictions caused by the pandemic the test was designed to be performed at home, using the provided data. The digital audio workstation Reaper was used to conduct all measurements.

6.1 Creating the Test

6.1.1 Generating the Distortion

The test contained generated nonlinearities of mechanical nature as well as synthetically created ones. Distorting speakers will lead to subtle distortion mostly prominent in the odd orders. Synthetic distortion generated by VST Plugins is much more versatile and produces a much more distinct distortion. Both sources of distortion were used in the test to be able to compare the identification of synthetic and real distortion. The nonlinearities for the test were generated by the following:

— Distorting speaker
— Melda Production MSaturator - VST Plugin

Distorting speaker For recording the distorted test samples, a recording setup consisting of

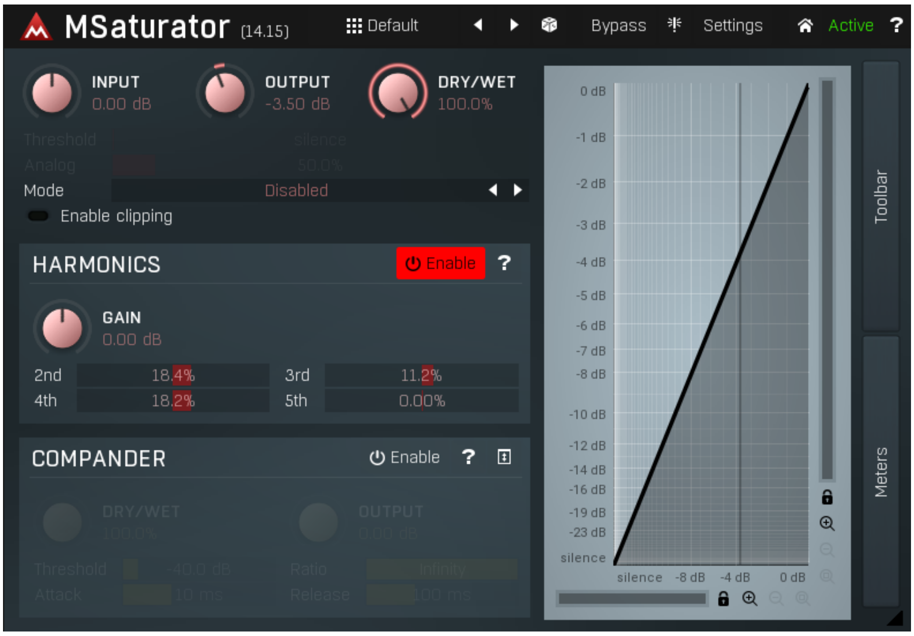
— 1" 1-way satellite speaker
— 1/2" measuring microphone
— Neumann artificial head
— 150W t.Amp Amplifier

was planted in the CUBE at the IEM in Graz. First an exponential sweep was played back by the speaker at high amplitude and recorded with the measuring microphone. The impulse response was derived as described in section 2.2 and windowed with a tapered cosine function to remove reflections and influences of the room as good as possible. Then the mono mix of the test samples were played back over the speaker and recorded with the artificial head to retrieve the distorted test samples used in the test as the reference signals.

The test samples with simulated distortion were created by sending the same test samples in mono through a effect chain consisting of the harmonic generator, followed by the MCFX convolver plugin in Reaper. Hereby, a variation in the order K i.e., the number of parallel branches used for the convolution as seen in figure 5 was introduced to retrieve more simulation results that could be used in the test. The resulting simulated samples consist only of one channel which is why they were treated with a room simulation and reverb to make the samples sound more like the recorded references.

Synthetically generated distortion To validate the performance of the application in a more confined environment, half of the presented test samples were generated synthetically. This way, any possible influences from the room or the speaker itself are eliminated and the amount of distortion is controllable in a much more precise way.

The plugin used to distort the samples was the Melda Production MSaturator Plugin. The signal wasn't saturated using a compressing transfer function (shown in the right side of the plugin interface in figure 24) as the generated harmonics would not have been easily determinable. Instead, the second, third and fourth harmonics were induced in an additive way defined over a value in percent of the total energy contained in the signal i.e., Harmonic Distortion at 1 kHz. This was later confirmed by analysing the signal with the application. The idea was to know exactly how many harmonics the signal contains to be able to simulate it accordingly i.e., not to over- or underfit.



INPUT 0.00 dB
OUTPUT -3.50 dB
DRY/WET 100.0%
Threshold silence
Analog 50.0%
Mode Disabled
Enable clipping
GAIN 0.00 dB
2nd 18.4%
3rd 11.2%
4th 18.2%
5th 0.00%
DRY/WET 100.0%
OUTPUT 0.00 dB
Threshold -40.0 dB
Ratio Infinity
Attack 10 ms
Release 100 ms

Figure 24 – Melda Production MSaturator Plugin

6.1.2 Test Samples

The test was performed using 12 test samples consisting of

- male and female voice recordings
- various music snippets

This ensured more generalized results then just relying on an exponential sweep as an input signal, although the sweep enables a very clear identification of differences over the band of interest, music and voice samples are suited better to identify transient differences which are noticeable as crackling or very bright noises. The music snippets were varying in terms of genre, amplitude, reverberation, etc. Half of the test samples were distorted using the saturator plugin, the other half consisted of recordings from the speaker.

6.1.3 Environment

Participants completed the test at home with headphones using the MUSHRA application. The samples were played back in stereo as there is no spacial information to be tested. Even though the final simulation will be used mostly in a binaural environment, it is beneficial to get rid of all unnecessary error sources during the test.

To avoid heavily varying results from the participants, a lower and upper bound was introduced. Hereby the nonlinearities were simulated with varying order (under/over-modeling of the nonlinear system). The lower bounds are realized as simulations that are deviating from the optimum on purpose to give the participant a bad reference. The upper bound was again resembled by the reference signal but with a first order lowpass filter applied at 12 kHz. This should minimize the subjective valuation of the participants in the test.

Participants entered their answers using sliders that cover semantic values ranging from "identical", "small difference" and "large difference". Doing so, the results carry more information in contrast to just a binary option of "same" and "different".

6.2 User Interface

A reference signal was displayed on the screen and was compared against 3 other samples. The test signals were associated with a slider that is initiated at a value of 50%. The test signals consisted of

— The optimum simulation
— 1 lower bound (underfitted simulation i.e., not enough harmonics simulated)
— 1 upper bound (reference with first order low pass filter)

The participants then repeatedly evaluated how similar each test signal is to the reference using music, voice and sweeps as test signals. The test signals were randomly picked from a pool and displayed in random order.

At the beginning, participants were confronted with the following text to explain the nature of the test:

This experiment compares (mechanical or synthetic) harmonic distortion with simulations of the distortion. The goal is to evaluate how well the simulation approximates the original distortion. Use Headphones! Specify how similar the distortion of the stimuli sounds in comparison to the reference on the left. The differences in timbre or reverb should be ignored. There will be 12 samples to compare, then the experiment is finished.

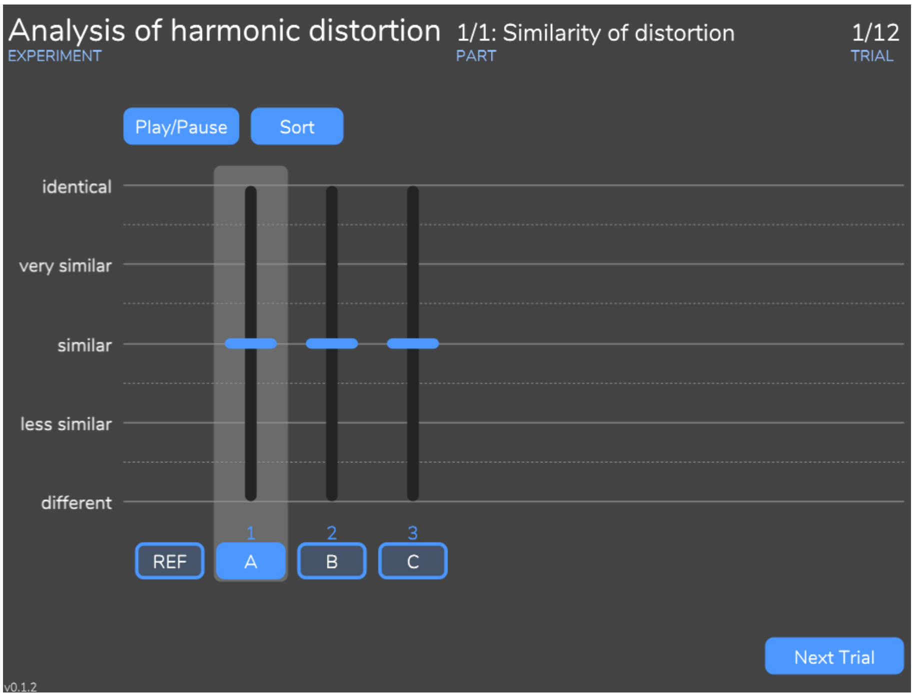


Figure 25 – The user interface of the presented listening test

6.3 Results

Although it was specified that the goal of the test is to evaluate the distortion of the sound only, timbre and room influence of the test samples very likely had an impact on the listener. Therefore, it is clear that test participants judged the samples differently depending on their ability to hear the distortion isolated from the rest. Participants that are experienced with hearing trials might be able to judge more objectively than less experienced participants. The conducted trial consists of 11 participants that come mostly from the audio engineering field with the exception of 2 participants coming from other branches.

The results of the listening test with synthetically generated distortion are shown in figure 26. The box on the left represents the answers given by the participants regarding the performance of the simulation executed with optimum modelling conditions. As seen in figure 24 it is known that the reference sample was distorted using a saturator and adding distortion up to the fourth harmonic. Therefore, the optimal simulation parameter was chosen to be $K = 4$. Acting as a lower boundary, column C shows the results of the simulation using only 2 harmonics i.e., being inefficient on purpose to provide a reference that differs more from the original. It is obvious to see that C performs significantly worse than A. A has its median in between "identical" and "very similar" and the lower quartile at "very similar" and is therefore performing very good. It is interesting to note that even the upper bound i.e., the original with a first order lowpass filter applied at 12 kHz performs worse than the simulation in the trial. This could be an indication for modifying the upper bound too much as the differences in the stimuli were very subtle.

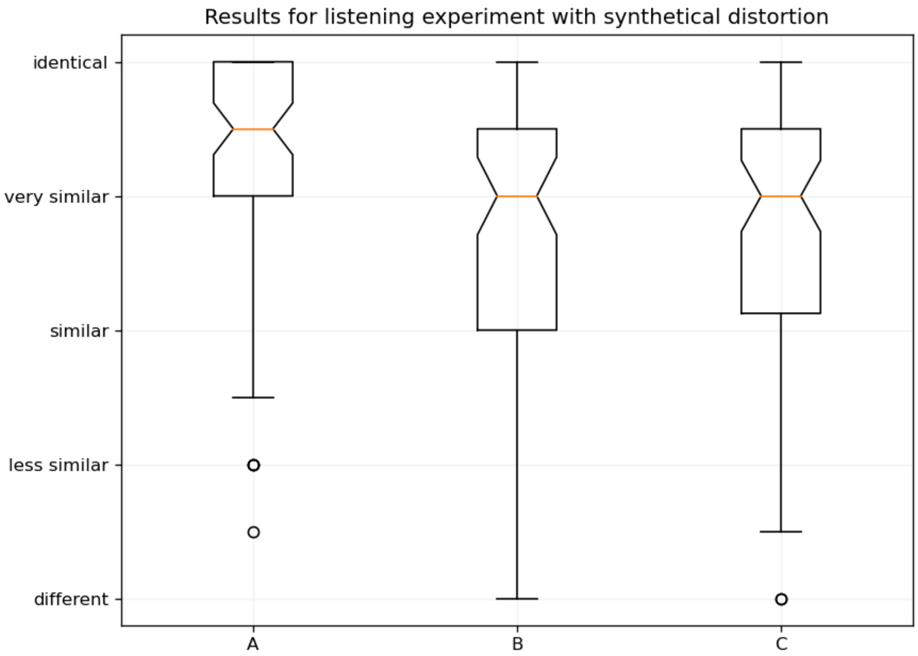


Figure 26 – **A**: Simulated distortions with optimum modelling conditions, **B**: Reference signal with first order low pass filter applied, **C**: Simulated distortions with insufficient number of harmonics

Figure 27 yields the results of the listening test for the mechanical distorted samples. Columns A to C are chosen the same as in the previous figure. It is striking that the median of the optimal solution lies in the vicinity of "less similar" which is a very strong difference compared to the synthetic distortion in figure 26. The upper bound B is evaluated to be between "identical" and very similar by approximately 75 percent of the participants and the lower boundary C ranges between "less similar" and "different". The divergent results of the second part of the listening test are traced to two factors. One being the loudspeaker that was used in the recording setup for the listening trial. By powering the loudspeaker beyond its capacity not only harmonic distortion appeared, but also non-harmonic distortion in form of rattling and buzzing of the enclosure. The current application is not capable of simulation this kind of distortion and is therefore failing to reproduce the distortion properly. By recording the test samples in the CUBE using the loudspeaker under very high load, the reproducibility of recordings also is lower than creating them solely in the DAW. While recording, the speaker would sound different over the course of a longer time, possibly because of the membrane or the coil slowly wearing out. Another possible cause for the worse performance of the simulation is the influence of room and timbre on the recording. Due to the artificial room simulation that was applied to the simulated test samples in order to adapt them to the original recordings, the room impression and the timbre differentiate from the original.

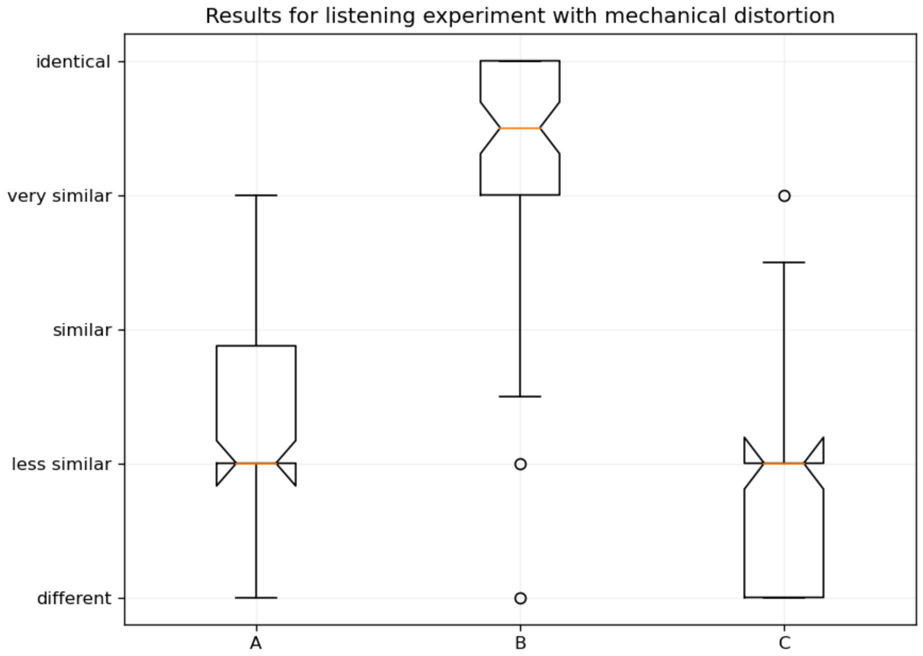


Figure 27 – **A**: Simulated distortions with optimum modelling conditions, **B**: Reference signal with first order low pass filter applied, **C**: Simulated distortions with insufficient number of harmonics

7 Conclusion

The listening test brought up the mismatch between synthetical and mechanical distortion. While it is possible to simulate synthetically generated distortion in a confined environment (i.e., the signal never leaves the DAW) nearly identically, the simulation performs far worse in a recording setup with a distorting loudspeaker. Mostly this is caused by the loudspeaker also producing other byproducts apart from harmonic distortion that contribute to the sound, such as rattling and buzzing of the enclosure. Further, the harmonic distortion produced by a loudspeaker is usually very low around less than 10 percent of total harmonic distortion. This leads to a small SNR for the higher orders and therefore to large amounts of noise in the extracted kernels, further reducing the quality of the simulation. It is beneficial to establish a high SNR while recording e.g., shield off noisy amplifiers from the microphone or, if necessary, reduce the distance to microphone.

The comparison of the directivity patterns of the loudspeaker under full load and with simulated distortions measured in the Double Circle Microphone Array revealed differences. In the vicinity of the crossover frequency the balloon plots show variation in the directivity for the second harmonic as seen in figure 20. This was according to the expectations as the harmonic would be played over the woofer for the reference whereas in the case of the simulation it would be played over the tweeter. The energy comparison in figures 21 to 23 further reinforce the hypothesis as the graphs deviate at the corresponding frequencies of the harmonics. However, the figures should be interpreted with caution since this is an experimental way of looking at the presented data.

The python app provides an algorithm that is not limited in terms of simulated harmonics with the option to export the kernels to a DAW using the MCFX Convolver Plugin. The foundation of the code strongly bases on the work of Rebillat in [RHCK10] combined with the work of Novak in [nls15] which makes the synchronized swept sine theory the subject of discussion. Given an up-to-date DSP and computer setup, simulations with an order of 16 at a sampling rate of 192 kHz can be performed in real time in a DAW.

References

[bfr18] m. brandner, m. frank, and d. rudrich, "dirpat—database and viewer of 2d/3d directivity patterns of sound sources and receivers," *journal of the audio engineering society*, may 2018.

[Bru14] P. Brunet, "Nonlinear system modeling and identification of loudspeakers," 2014.

[Far00] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," 11 2000.

[GB10] F. Giri and E.-W. Bai, *Block-oriented nonlinear system identification*. Springer, 2010, vol. 1.

[Joj] Jojek. Sweep inverse filter spectrum. [Online]. Available: https://dsp.stackexchange.com/questions/41696/calculating-the-inverse-filter-for-the-exponential-sine-sweep-method

[MSWH09] Q. Meng, D. Sen, S. Wang, and L. Hayes, "Impulse response measurement with sine sweeps and amplitude modulation schemes," 01 2009, pp. 1 – 5.

[nls15] a. novak, p. lotton, and l. simon, "synchronized swept-sine: theory, application, and implementation," *journal of the audio engineering society*, vol. 63, no. 10, pp. 786–798, october 2015.

[RHCK10] M. Rébillat, R. Hennequin, E. Corteel, and B. F. Katz, "Identification of cascade of Hammerstein models for the description of nonlinearities in vibrating devices," *Journal of Sound and Vibration*, vol. 330, no. 5, pp. 1018–1038, Sep. 2010. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00619301