A linear phase IIR filterbank for the radial filters of ambisonic recordings

Christophe Langrenne, Eric Bavu, Alexandre Garcia

A LINEAR PHASE IIR FILTERBANK FOR THE RADIAL FILTERS OF AMBISONIC RECORDINGS

C. Langrenne[1], É. Bavu[1] and A. Garcia[1]
[1] Conservatoire National des Arts et Métiers (CNAM)
Laboratoire de Mécanique des Structures et des Systèmes Couplés (LMSSC)
christophe.langrenne@lecnam.net

ABSTRACT

Higher order ambisonics involve excessive bass boosts, especially for high orders and at low frequencies. In order to avoid unnecessary and excessive amplification for components that do not contribute significantly to the soundfield, a filterbank can be used in order to cut-off noise amplification. In the present paper, we propose a linear phase IIR filterbank implementation that allows to avoid the used of fast block convolutions or nonlinear IIR filters. Our approach is based on local overlap-add time reversed block convolutions, which allow the filterbank to exhibit a linear delay, which only depends on the sectioned block size. When combined with radial filters of a rigid spherical recording array, this approach allows to change the cutoff frequencies of the filterbank with more flexibility than precomputed FIR filterbanks.

1. INTRODUCTION

Higher order Ambisonics decomposition of natural soundfields is often performed using rigid spherical microphone arrays, mainly because of its simple implementation [1, 2]. All the electronic equipment can be conveniently placed inside the spherical measurement array, without affecting the scattered acoustic field. However, restitution systems for HOA sound field synthesis generally exhibit a much larger radius than measurement arrays. The well-known bass-boost effect is directly linked to the relatively small size of the measurement array: low frequencies have to be amplified, especially for higher order components of the Ambisonics decomposition. The dynamic range for filtering purposes is limited, mainly by the signal-to-noise ratio of the microphone array. In order to overcome this problem, we developed a microphone array prototypes using analog MEMS microphones, which have become a viable solution in a small packaging, with a reasonable price, thanks to the growing use of these sensors in domotics and in the mobile phone industry. MEMS microphone from the same production batch exhibit very similar characteristics and can be used for array signal processing without any level or phase calibration. The proposed prototype is made of groups of 4 MEMS microphones for the same sensor position, in order to improve the signal to noise ratio by 6 dB. This microphone array is a 5^{th} order Ambisonics system (50 sensors 200 MEMS on a Lebedev grid).

Nevertheless, this approach does not dispense from the need to filter higher order coefficients. A simple high-pass filtering on each order component is not sufficient, since this would not only cause losses in terms of amplitude and power but would also affect the loudness of the restitution. A filterbank is therefore needed in order to cut-off noise amplification at low frequencies, and to apply appropriate gains for loudness equalization. For this purpose, Baumgartner *et al.* [4] proposed a non-linear phase filterbank based on Linkwitz-Riley IIR filters. In order to avoid group delay distortions, Zotter proposed a linear phase filterbank based on FIR filters and the use of fast block convolutions [5]. This solution is although not very flexible, since the FIR strongly depend on the radius of the measurement array, and on the filterbank's cut-off frequencies. Any change in the measurement system require a new computation of each corresponding FIR filters.

In the present paper, a linear phase IIR filterbank is implemented. Thanks to the use of local overlap-add time reversal blocks [6], the filterbank exhibits a linear phase delay which only depends on the the time reversal blocksize. The proposed filterbank implementation allows to change in real-time the frequency bands and the loudness equalization (diffuse or free field equalization) using the Faust programming language [7].

2. AMBISONICS

In this paper, we use the following convention for spherical coordinates:

$$\begin{cases} x = r\cos(\theta)\cos(\delta) \\ y = r\sin(\theta)\cos(\delta) \\ z = r\sin(\delta) \end{cases} \quad (1)$$

with r, θ and δ denoting the radius, the azimuth and the elevation angle (see Fig. 1).

doi:10.25836/sasp.2019.03

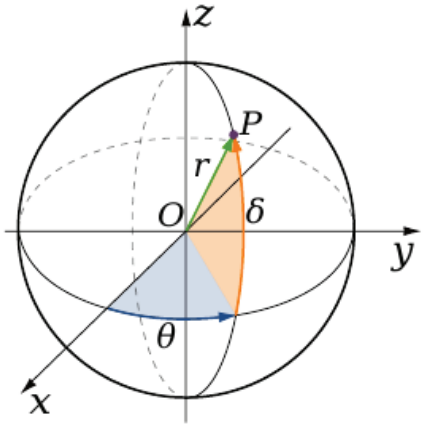

Figure 1. Spherical coordinate system. A point P (x, y, z) is described by the radius r, azimuth θ and elevation δ.

For any incident sound field, the pressure $p(kr, \theta, \delta)$ can be approximated as a truncated Fourier-Bessel series of order M:

$$p(kr, \theta, \delta) = \sum_{m=0}^{M} i^m j_m(kr) \sum_{n=-m}^{m} B_{mn} Y_{mn}(\theta, \delta) \quad (2)$$

with $i = \sqrt{-1}$, k the wave number, $j_m(kr)$ the spherical Bessel function, $Y_{mn}(\theta, \delta)$ the real spherical harmonic of order m and degree n and B_{mn} the wave spectrum.

In (2), the spherical harmonics are defined by:

$$Y_{mn}(\theta, \delta) = \sqrt{(2m+1)\epsilon_n \frac{(m-|n|)}{(m+|n|)}} P_m^{|n|}(\sin(\delta)) \times \begin{cases} \cos(|n|\theta) & \text{if } n \geq 0 \\ \sin(|n|\theta) & \text{if } n < 0 \end{cases} \quad (3)$$

with $\epsilon_n = 1$ for $n = 0$ and $\epsilon_n = 2$ for $|n| > 0$, $P_m^{|n|}$ are the associated Legendre polynomials of order m and degree $|n|$.

2.1 Encoding and decoding

In this section, we consider both spherical microphone and loudspeaker arrays, whose transducers are sampled on a spatial grid with L nodes calculated from a quadrature rule which is exact up to the order M. The recording, encoding and decoding operations are summarized as follows, in the frequency domain [9]:

$$s_{spk} = \underbrace{W Y F_{spk}^{-1}}_{decoding} \quad \underbrace{E_{mic} Y^T W s_{mic}}_{encoding} \quad (4)$$

where $s_{spk}^{L\times 1} = [s_1, s_2, \cdots, s_l, \cdots s_L]^T$ is the vector of input signals for the loudspeakers, $W^{L\times L}$ is the diagonal matrix of quadrature weights, $Y^{L\times(M+1)^2}$ is the matrix of spherical harmonics evaluated at the different nodes. $F_{spk}^{-1^{(M+1)^2\times(M+1)^2}}$ is the diagonal matrix of near field compensation filters with diagonal term:

$$F_{spk}(kr_{spk}) = i^{-(m+1)} k h_m^{(2)}(kr_{spk}) \quad (5)$$

where r_{spk} is the distance of the loudspeakers and $h_m^{(2)}$ the spherical Hankel function of second order. $E_{mic}^{(M+1)^2\times(M+1)^2}$ is the diagonal matrix of equalization filters for the rigid spherical microphone array with diagonal term given by:

$$E_{mic}(kr_{mic}) = i^{-(m-1)} (kr_{mic})^2 h_m^{'(2)}(kr_{mic}) \quad (6)$$

where r_{mic} is the radius of the rigid sphere supporting the microphones and $h_m^{'(2)}(kr)$ the derivative of the spherical Hankel function of second order with respect to the variable kr.

Finally $s_{mic}^{L\times 1}$ are the signals captured by the microphones. Transposition is denoted by the superscript T.

2.2 Radial Filters

The present paper specifically focuses on the radial filters, which correspond to the correction of the rigid spherical measurement array scattering and to the compensation of the restitution loudspeakers, which is obtained usingEqs. (5) and (6):

$$G_m(kr_{mic}, kr_{spk}) = -kr_{mic}^2 \frac{h_m^{'(2)}(kr_{mic})}{h_m^{(2)}(kr_{spk})} \quad (7)$$

For a digital implementation of these filters, one can use the following expressions [10]:

$$h_m^{(2)}(kr) = i^{m+1} \frac{e^{-ikr}}{kr} \sum_{k=0}^{m} a_{m,k} (ikr)^{(-k)} \quad (8)$$

with

$$a_{m,k} = \frac{(m+k)!}{(m-k)!k!2^k} \quad (9)$$

and

$$h_m^{'(2)}(kr) = \frac{m}{kr} h_m^{(2)}(kr) - h_{m+1}^{(2)}(kr) \quad (10)$$

Using eqs. (7), (8) and (10), the radial filters can be expressed as:

$$G_m(kr_{mic}, kr_{spk}) = ikr_{mic} r_{spk} e^{-ik(r_{mic}-r_{spk})} \frac{\sum_{k=0}^{m+1} a_{m,k} (ikr_{mic})^{(-k)} - m \sum_{k=0}^{m} a_{m,k} (ikr_{mic})^{(-k-1)}}{\sum_{k=0}^{m} a_{m,k} (ikr_{spk})^{(-k)}} \quad (11)$$

In this equation, the term $e^{-ik(r_{mic}-r_{spk})}$ is a delay corresponding to the propagation time between the radius of the loudspeakers and the radius of the rigid sphere and can be omitted. The term ik can be interpreted as a differentiator filter. The last term can be efficiently implemented as a discrete-time IIR filter, as proposed by Daniel for nearfield compensation [11]. In our practical implementation, we use the formation proposed by Adriaensen in [12].

doi:10.25836/sasp.2019.03

3. LINEAR PHASE IIR FILTERBANK DESIGN

At low frequencies and for high order components, the compensation gains exhibit very high values. As a consequence, there is a need to stabilize the filters for different Ambisonic orders m, using high-pass filters, whose slopes exceed $6m$ dB/oct. However, the use of this procedure induces a noticeable loudness loss below each cut-off frequency, due to the filtering of signals of higher orders than m. One way of circumventing this problem consists in designing a filterbank with cross-overs with compensated amplitudes as proposed by Baumgartner in [4], who proposed the use of a Butterworth filterbank with all-pass based phase compensation in order to have the same non-linear phase for each of the decomposition orders.

In order to avoid group delay distortions at low frequencies, we propose a realtime implementation of a linear phase IIR filterbank, whose implementation is based on the Powell and Chau technique [6]. This procedure is based on a conventional two-pass IIR filter. The time reversed section implementation of the noncausal function $H(z^{-1})$ is cascaded with the original causal IIR filter $H(z)$. The equivalent transfert function therefore becomes:

$$H_{eq}(z) = H(z^{-1})H(z)e^{-j\omega d(L)} \tag{12}$$

whose phase is linear and magnitude is equal to the square of the magnitude of the original IIR filter $H(z)$. The term $e^{-j\omega d(L)}$ corresponds to the overall delay induced by the time reversal sectioning procedure, which is $(4L-1)$ samples in our implementation.

Using a Butterworth filter $H(z)$, one obtains a linear Linkwitz-Riley filter, with crossover frequencies located at -6 dB attenuation. Using the well-known overlap-add method for sectioned convolution, the output can be obtained in realtime from a superposition of finite length responses from adjacent input sections of length L. The finite length of these input section correspond to a truncature of the IIR filter, which can cause a ripple in the pass-band section of the filter. Taking a sufficiently long value of L allows to ensure that the response magnitude is undistinguishable from the ideal magnitude response, with a nearly constant group delay.

4. REAL-TIME IMPLEMENTION WITH FAUST

Faust (Functional Audio Stream) is a functional programming language for sound synthesis and audio processing with a strong focus on the design of synthesizers, musical instruments, audio effects, etc. Faust targets high-performance signal processing applications and audio plug-ins for a variety of platforms and standards. It is used on stage for concerts and artistic productions, in education and research, in open source projects as well as in commercial applications.

4.1 Linear phase filter

Table (1) recalls the conventional overlap-add method for sectioned convolution. It is worth noticing that the original signal is split in two branchs in order to separate successive sections of L samples. Each branch is filtered using an IIR filter which is reset at the end of $2L$ samples, because it is a truncated sectioned convolution.

This overlap-add scheme can also be used to implement noncausal time reversed convolutions. The process proposed by Powell and Chau [6] consists in the following (see Table (2)):

1. Time reverse each input section using a LIFO (Last Input First Output)
2. Split the reversed sectioned signal and use the reseted IIR filter on each branch
3. Create output sections consisting of the trailing response from the current input section plus the leading response from the previous input section. To be implemented in real-time, this supposes to split again the branch to separate leading and trailing sections, and to delay the leading section of each branch by $2L$ samples
4. Time reverse the output sections using a LIFO.

4.2 Faust implementation

We use the following Faust implementation of a LIFO:

```
LIFO(L) = @(phase(L)*2) with {
phase(n) = (1) : (+ : %(n)) ~ _ ;
};
```

Each LIFO therefore induces a delay of $(L-1)$ samples before performing the time reversed sections. As a consequence, the clocks used to obtain the different branches begin with a first section of $(L-1)$ samples, and the following clock sections are of length L.

The overall FAUST implementation of a linear phase filter is shown on Fig. 2. A one-sample delay before the second LIFO is used in order to compensate the missing sample of the first LIFO. This therefore allows to use the same clocks, both for forward and backward filterings. Since each LIFO induces a delay of $(L-1)$ samples and the reversed time section induces a delay of $2L$ samples, this linear phase IIR filtering process exhibits an overall delay of $(4L-1)$ samples.

5. LINEAR PHASE FILTERBANK

The proposed filterbank implementation only makes use of passband filters. We made this choice in order to avoid soliciting the loudspeakers at very low frequencies. At high frequencies, a low-pass filter is added in order to combine it with the differentiator filter given in (11). The zero and the pole of these two filters compensate in order to avoid problems at high frequencies, when only the differentiator filter is used. This filter is not present in the first subsection, but only implemented in the second subsection.

doi:10.25836/sasp.2019.03

	section k (L)	section k+1 (L)	section k+2 (L)
x(n) clk_1 clk_2	$x_k(1),\cdots,x_k(L)$ $1\cdots1$ $0\cdots0$	$x_{k+1}(1),\cdots,x_{k+1}(L)$ $0\cdots0$ $1\cdots1$	$x_{k+2}(1),\cdots,x_{k+2}(L)$ $1\cdots1$ $0\cdots0$
x(n) $\times clk_1$ x(n) $\times clk_2$	$x_k(1)\cdots,x_k(L)$ $0\cdots0$	$0\cdots0$ $x_{k+1}(1)\cdots,x_{k+1}(L)$	$x_{k+2}(1),\cdots,x_{k+2}(L)$ $0\cdots0$
IIR(x(n) $\times clk_1$) IIR(x(n) $\times clk_2$)	Leading k Trailing $k-1$	Trailing k Leading $k+1$	Leading $k+2$ Trailing $k+1$
Sum=y(n)	$y_k(n)=L_k+T_{k-1}$	$y_{k+1}(n)=L_{k+1}+T_k$	$y_{k+2}(n)=L_{k+2}+T_{k+1}$

Table 1. Overlap-add method for sectioned convolution

	section k (L)	section k+1 (L)	section k+2 (L)
x(n) clk_1 clk_2	$x_k(1),\cdots,x_k(L)$ $1\cdots1$ $0\cdots0$	$x_{k+1}(1),\cdots,x_{k+1}(L)$ $0\cdots0$ $1\cdots1$	$x_{k+2}(1),\cdots,x_{k+2}(L)$ $1\cdots1$ $0\cdots0$
LIFO(L)=x(-n)	$x_k(L),\cdots,x_k(1)$	$x_{k+1}(L),\cdots,x_{k+1}(1)$	$x_{k+2}(L),\cdots,x_{k+2}(1)$
x(-n) $\times clk_1$ x(-n) $\times clk_2$	$x_k(L)\cdots,x_k(1)$ $0\cdots0$	$0\cdots0$ $x_{k+1}(L)\cdots,x_{k+1}(1)$	$x_{k+2}(L),\cdots,x_{k+2}(1)$ $0\cdots0$
IIR(x(-n) $\times clk_1$) IIR(x(-n) $\times clk_2$)	Leading k Trailing k-1	Trailing k Leading k+1	Leading k+2 Trailing k+1
(IIR(x(-n) $\times clk_1$) $\times clk_1$)@(2L) IIR(x(-n) $\times clk_1$) $\times clk_2$ (IIR(x(-n) $\times clk_2$) $\times clk_2$)@(2L) IIR(x(-n) $\times clk_2$) $\times clk_1$	Leading k-2 $0\cdots0$ $0\cdots0$ Trailing k-1	$0\cdots0$ Trailing k Leading k-1 $0\cdots0$	Leading k $0\cdots0$ $0\cdots0$ Trailing k+1
Sum=y(-n)	$y_{k-2}(-n)=L_{k-2}+T_{k-1}$	$y_{k-1}(-n)=L_{k-1}+T_k$	$y_k(-n)=L_k+T_{k+1}$
LIFO(L)=y(n)	$y_{k-2}(n)$	$y_{k-1}(n)$	$y_k(n)$

Table 2. Overlap-add method for time reversed convolution. @(2L) means to apply a delay of 2L samples.

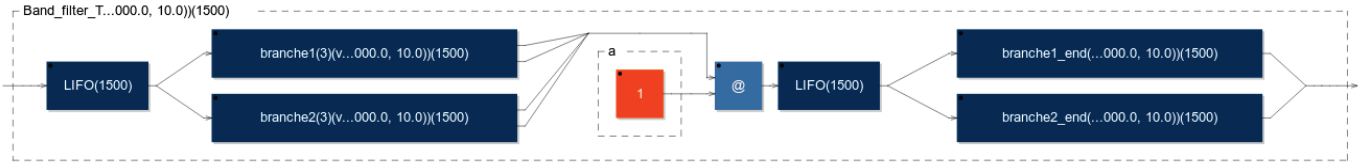

Figure 2. Block diagram of the Faust implemention.

5.1 Without radial filters of ambisonic recording

Fig. 3 shows the filterbank response for a sectioned convolution of length $L = 100$ samples. This figure shows that the choice of $L = 100$ is not sufficient for the lower passband filter, which has a lower bound of 20 Hz. At low frequencies, the impulse response is long and the truncated IIR length must be larger. On the other hand, one can notice that for high passband filters, the dynamic range are > 100 dB with only 100 sample sections.

Fig. 4 shows that the use of larger L values allow to get closer to the ideal filter response and very low frequencies. With $L = 900$ samples, the filterbank response is shown on Fig. 5. Its total response is flat by design, since it corresponds to a Linkwitz-Riley filterbank with crossover frequencies located at -6 dB attenuation.

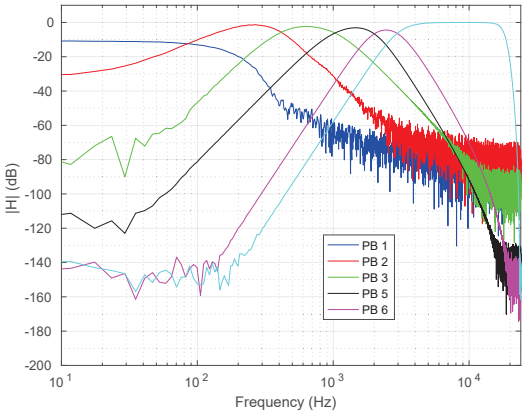

Figure 3. Filterbank response with $L = 100$

doi:10.25836/sasp.2019.03

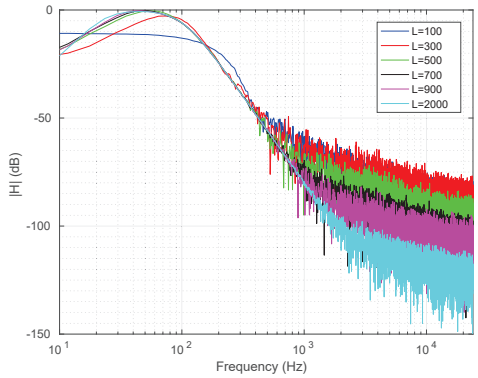

Figure 4. Low passband filter with different lengths of sectioned convolutions.

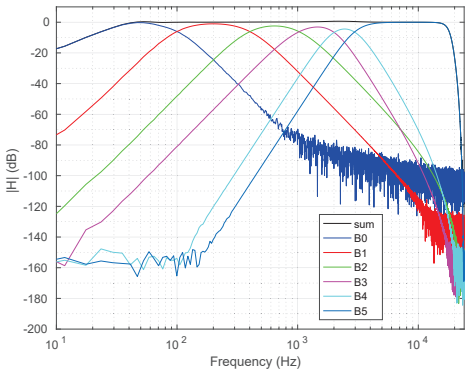

Figure 5. Filterbank response with $L = 900$.

As shown on Fig. 7, the group delay is almost constant around the ideal value of $(4L-1)/F_s$, with $F_s = 48$ kHz. Fig. 7 allows to show that the group delay only deviates from this ideal value by less that 0.5 msec from 20 Hz to 20 000 Hz.

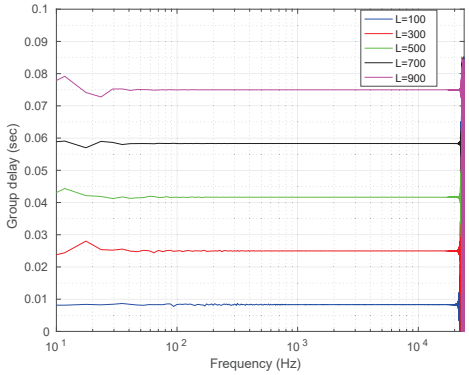

Figure 6. Group delays for different lengths of sectioned convolutions.

5.2 With radial filters of ambisonic recording

Our 5$^{\text{th}}$ order ambisonic recording prototype is a rigid sphere of radius 0.07 m. The 50 MEMS microphones are spatially sampled on the sphere using a Lebedev rule (see Fig. 8). Because of the large radius, this is not necessary to filter the first ambisonic order. In return, the aliasing frequency is 6500 Hz. Our spherical restitution prototype is composed of 50 loudspeakers, on the same Lebedev grid, at a radius of 1.07 m. For the low frequencies, six larger

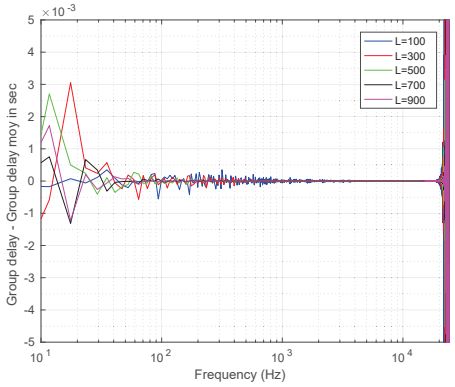

Figure 7. Deviations from the ideal group delays, for different lengths of sectioned convolutions.

loudspeakers are added and driven with ambisonic signals up to order 1 ((see Fig. 9).

Figure 8. Cnam prototype of 5th order ambisonic recording sphere.

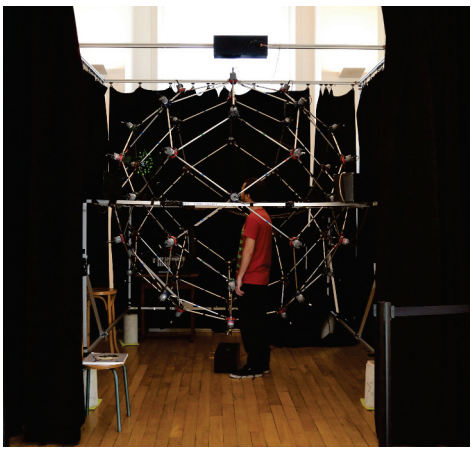

Figure 9. Cnam prototype of 5th order ambisonic restitution sphere.

Fig. 10 and Fig. 11 show the magnitude and the phase responses of the radial filter obtained after the proposed filterbank. The delay term of (11) is not included, and the ideal delay of (4L-1) samples induced by the proposed filterbank implementation is also compensated. When compared with the theoretical filters of (11), small differences

can be noticed above 10 kHz, due to the differentiator filter, which is above the aliasing frequency. The filterbank cutoff frequencies have been chosen in order to prevent any gain values exceeding 40 dB. These values can be changed in real time with a slider in the Faust panel. An amplitude compensation gain has also been added in the final version, as suggested by Baumgartner in [4].

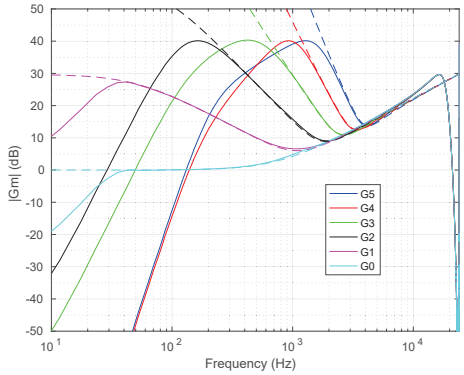

Figure 10. Magnitude response of G_m. Solid line : measured after the filterbank , Dashed line: ideal eq.(11).

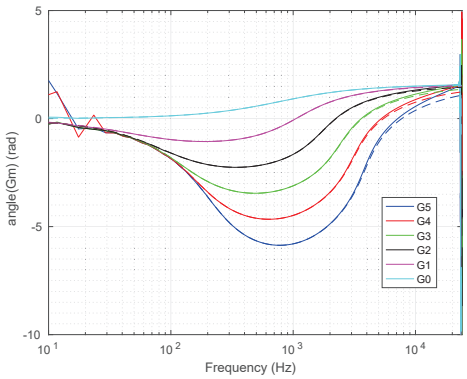

Figure 11. Phase response of G_m. Solid line : measured after the filterbank , Dashed line: ideal eq.(11).

6. CONCLUSIONS

A linear phase IIR filterbank is presented in this paper. When combined with the radial filters of an ambisonic recording rigid sphere, it allows to change the cutoff frequencies in real-time in order to adjust them more precisely, while achieving a linear phase filterbank. In order to meet the requirements for the lower passband filter, a truncated sectioned IIR filter with 900 samples is needed for $F_s = 48$ kHz, , which induces an overall delay of 75 msec.

The final code can be compiled up to the 5th ambisonic order. However, some buffer overflows appear from time to time at the 5th order. The code runs on only one CPU core, and the solution could come from the use of multiple cores, which is a challenge for audio stream synchronization.

7. ACKNOWLEDGMENTS

The authors would like to gratefuly thank Yann Orlarey and Stéphane Letz from the Grame institut. Yann Orlarey helped us with the LIFO Faust code and Stéphane Letz took his time to help us to improve the Faust code compilation. We also express our gratitude to Philippe Chenevez, head of Cinela society, for the realization of the so-called "MemsBedev" ambisonic recording sphere.

8. REFERENCES

[1] S. Bertet, J. Daniel, E. Parizet, L. Gros, and O. Warusfel. Investigation of the perceived spatial resolution of higher order ambisonics sound fields: a subjective evaluation involving virtual and real 3D microphones. In *AES 30th International Conference*, Saariselk, Finland, 2007.

[2] J. Meyer and G. Elko. A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield, *Proceedings of ICASSP02*, vol 2, Orlando, FL, USA, 2002.

[3] S. Favrot, M. Marschall , J. Ksbach , J. Buchholz, T. Weller. Mixed-order Ambisonics recording and playback for improving horizontal directionality, presented at the *AES 131st convention*, New York, USA, 2011.

[4] R. Baumgartner, H. Pomberger, and M. Frank. Practical Implementation of Radial Filters for Ambisonic Recordings, in *Proc. first International Conference on Spatial Audio*, Detmold, Germany, 2011.

[5] F. Zotter. A Linear-Phase Filter-Bank Approach to Process Rigid Spherical Microphone Array Recordings, in *Proceedings the 5th International Conference on Electrical, Electronic and Computing Engineering, IcETRAN 2018*, Pali, Serbia, 2018.

[6] S.R Powell and P.M Chau. A technique for realizing linear phase IIR filter, in *IEEE transactions and signal processing*, vol 29(11), pages 2425–2435, 1991.

[7] for Faust programming language, see https://faust.grame.fr/.

[8] P. Lecomte and P.-A. Gauhtier Real-time 3D ambisonics using Faust, processing, Pure Data, and OSC, in *Proc. of the 18 Int. DAFx Conference*, Trondheim, Norway, Nov 30 - Dec 3, 2015.

[9] P. Lecomte, P.-A. Gauthier, C. Langrenne, A. Berry, and A. Garcia A Fifty-Node Lebedev Grid And Its Applications To Ambisonics, Journal of the Audio Engineering Society, 64(11), (2016).

[10] M. Abramowitz and I. A. Stegun Handbook of mathematical functions, 10th Edition, Dover Publications, 1972.

[11] J. Daniel Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new Ambisonic format, in . *23rd AES conference*, Copenhagen, Denmark, 2003.

[12] Fons Adriaensen Near field filters for higher order Ambisonics, http://kokkinizita.linuxaudio.org/papers/hoafilt.pdf, 2006, Accessed: 2019-06-25.

doi:10.25836/sasp.2019.03