



# Implementation of non-equal partitioned multi-channel convolver

Luca Battisti<sup>1</sup>

Department of Electronics, Information and Bioengineering, Polytechnic of Milan  
Milan, Italy

Angelo Farina<sup>2</sup>

Department of Industrial Engineering, University of Parma  
Area delle Scienze, Parma, 43100, Italy

Antonella Bevilacqua<sup>3</sup>

Department of Industrial Engineering, University of Parma  
Area delle Scienze, Parma, 43100, Italy

Lamberto Tronchin<sup>4</sup>

Department of Architecture, University of Bologna  
Cesena, Italy

## ABSTRACT

*Convolution has become a largely exploited signal operation due to its several applications in digital signal processing. In the realm of audio elaboration, convolution has been used to impose a spectral and/or a temporal structure onto a signal. This is possible by convolving the sound signal with the Room Impulse Response (RIR). The acoustic footprint of these sound signals can be completely transferred to another sound signal. With a multichannel approach, convolution assumes even wider application fields. One of the outcomes can be considered the Ambisonics recording made by a multi-channel convolver. A similar concept can be applied to the mixing phase of audio post-production, where direction-based audio objects are converted to Ambisonics to be reproduced in similar speaker setups. This paper deals with the analysis of an existing algorithm related to a multichannel convolver software and evaluates its efficiency to optimize the results for non-expert users. The results show that the management of the filter matrix outlines weaknesses in the process of assembling new matrices. A proposed solution regards the optimization of such algorithm.*

## 1. INTRODUCTION

Stereophony has been always considered a profitable business in broadcasting and music production that includes surround techniques that have been initially developed for movie theater applications,

---

<sup>1</sup> Luca.battisti@polimi.it

<sup>2</sup> Angelo.farina@unipr.it

<sup>3</sup> Antonella.bevilacqua@unipr.it

<sup>4</sup> Lamberto.tronchin@unibo.it



to best give an immersive audio perception to the audience. One of the first sound systems is considered the 6-channel surround format (Dolby Stereo) created in 1977. Systems such as Ambisonics, Binaural, Ambiophonics, and Wave Field Synthesis (WFS) adopt the soundfield (or wavefiled) approach instead of the speaker-directional paradigm utilized by the traditional systems [1]. The Ambisonics transmission channels contain a speaker-independent representation of sound field that must be decoded by the reproduction system. This approach allows a representation in terms of source directions rather than loudspeaker positions and offers a considerable degree of flexibility in the number of speakers used for playback [2].

In recording activity, the number of coincident capsules of an omnidirectional microphone has effects on the spatial resolution. It is noted that the recording method requires also an encoding phase that converts the recording tracks from the physical capsule signals (A-format) to the Ambisonics signals (B-format). Despite its solid technical foundation and many advantages, Ambisonics and other spatial audio techniques had not until recently been a commercial success, especially for its historical necessities of powerful digital signal processing machines [3].

Digital filters have become very important for their more versatile applications with respect to analog ones. The linear time-invariant (LTI) systems grant to compute or extract the impulse response (IR), while digital processing can be practiced with convolution. In the realm of audio processing, convolution enact a spectral and/or temporal structure onto a sound signal. These structures are given by the signal to which the convolution is applied [4].

IRs are used to creating sound spatial effects where the acoustic characteristics of the environment are recreated. This can be achieved by designing IRs artificially (e.g. simulated reverberation generators) or by acquiring them directly from real environments. Both techniques have been used to create a sort of virtual soundscape in the soundtrack. Furthermore, the number of IRs required for the 3D sound depends on the grade of the surround refinement to be achieved.

A multichannel convolver is used for both spatial audio processing and audio spatial effects. One application is the encoding/decoding to obtain Ambisonics signals from a multichannel microphone in a generic space distribution; others regard the extraction of speaker-independent sound field signals to be reproduced in an Ambiophonics system. Special IR matrices can be adopted to perform a super directional microphone or a super directional loudspeaker simulating musicians' movements on the stage.

This paper explores traditional and more efficient ways to perform convolution maintaining the real-time features as a constraint to keep the multichannel convolver suitable for general purposes. On this basis, an algorithm implementation of an open-source multichannel convolver has been developed as an audio plug-in software. To be suitable for non-experts, a handier software is proposed by analyzing its optimization compared to the previous ones. In particular, the contents of this paper are organized as follows:

- An overview of different convolution algorithms is introduced in Section 2 by comparing the less optimized time-domain approach to the non-uniform partitioned IR convolution.
- Section 3 illustrates a theoretical background to analyzing convolution algorithms.
- Section 4 outlines the implementation of the open-source software algorithm compared to the state of the art.
- Section 5 shows the different implementation schemes adopted in the original and the renovated version of the software.



## 2. STATE OF THE ART

Methods to perform convolution trying to reduce the computational load and the amount of latency have been studied over the last decades. Many of these methods involve the Discrete Fourier Transform (DFT) and the partitioning of the IR, which makes the process possible in real-time. The DFT-based frequency domain implementation considerably reduces computation load, but at the same time it introduces delays in the pipeline process, which is directly proportional to the length of the IR [5]. Therefore, while the direct-form method immediately transmits the output sequence from the very first sample, this method must wait for an entire block of samples of the exact same length as that of the IR. This constraint has become a concern, especially for audio mixing applications, surround reproduction, and recording techniques involving multiple channel convolution.

### 2.1. Filtering in Time-Domain

Initially, digital filtering was used to modify the frequency domain characteristics (spectrum) of sound signals [6]. The convolution has developed for the corresponding operation in the time domain of the source signal to be elaborated with the IR. In other words, the convolution occurs by filtering a sound signal with another with the determination of a spectral or temporal structure onto the sound signal [7].

Digital filters are categorized and can be Infinite Impulse Response (IIR) or Finite Impulse Response (FIR) but in this paper only the FIR filter has been explored as a sum-of-products calculation. In the time domain, the computational is considered tedious for room responses of several seconds.

### 2.2. Filtering in Frequency-Domain

To overcome the computational load issue the convolution can be performed in a frequency domain, after having collected enough input samples to compute a Discrete Fourier Transform (DFT) [7]. Based on the Fourier transform theorem, the convolution in the time domain is equivalent to multiplication in the frequency domain. This means that after the use of the DFT to get the frequency representation of the signals (source and IR), these latest ones should be multiplied to obtain the frequency representation of the output and be transformed to the time domain again with the inverse DFT. The DFT conversion grows logarithmically with the IR length, while growing linearly with the IR in the direct form.

### 2.3. Impulse Response partitioning

In order to reduce the delay by the large-block convolution one of the options is to split the IR filter into blocks (subsequences) of equal or different sizes, compute the convolution of each block with the input signal, and sum the partial results to obtain the output signal. This means that the output sequence can be obtained without waiting for the whole elaboration which is the cause of the overall delay [8]. To achieve this objective, the subsequences shall be delayed based on the position of the block within the entire IR filter. This model results less efficient compared to the DFT method filtering a single large sequence but is still largely more efficient than the direct-form FIR convolution. However, computing many subsequences means more computation for the DFT conversions and more memory references. To optimize this process, it is possible to split the filter response into small blocks at the beginning of the IR and then increase the block size at the later time of the IR filter; the result is the best-optimization algorithm with no delay.

### 2.4. Multi-channel filtering

For the realization of more complex surround techniques such as Ambisonics [9], WFS [10] Ambisonics [11], or Spatial Multi-Input Multi-Output (MIMO) IR [12], a better efficiency combined with a low input-output delay is required.

By considering the Spatial MIMO IR, for example, the beamforming filter matrices should be taken for the analysis. As such, the multichannel convolution requires great computational capacities to process up to  $32 \times 32$  filter matrices realized with a specific IR partitioning [13].

### 3. THEORETICAL BACKGROUND

The convolution operation in the discrete time domain is described in equation (1) and (2).

$$y[n] = x[n] * h[n] \quad (1)$$

$$y[n] = \sum_{k=0}^{N-1} h[k] x[n - k] \quad (2)$$

DSP diagram model of this kind of processing is reported in Figure 1.

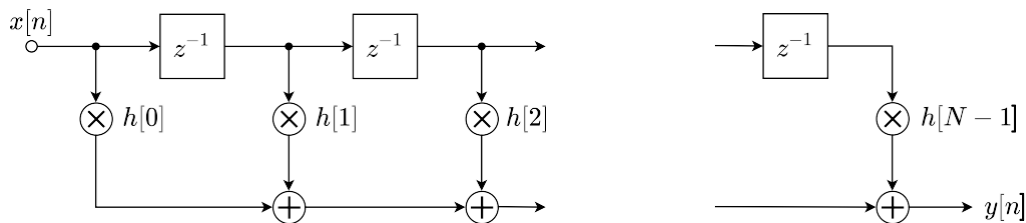


Figure 1: Direct-form implementation of convolution

Figure 1 shows that the computational load grows directly proportional to the length of the IR  $h[n]$ , composed of  $N$  samples [14]. Digital filtering in the time domain is a linear time-invariant system that implements the linear convolution of the input signal with the IR.

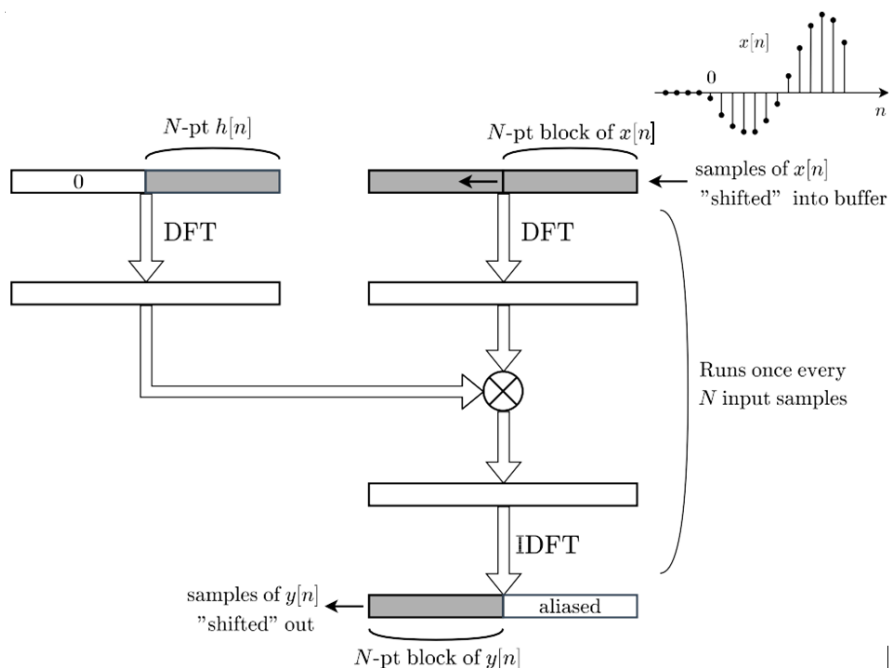


Figure 2: Large-block implementation of a convolution in frequency domain.



The sampling process in the frequency domain forces the signal to be periodic in time. This implies that the Inverse Discrete Fourier Transform (IDFT) of two  $N$ -length signals corresponds to the convolution of one sequence with the periodic version of the other [15]. This convolution is called circular convolution between two  $N$ -length signals, but it is not a correct representation of the same process in the time domain. A method to perform a linear convolution instead of a circular one is to add zero samples to both signals to distance the periodic replicas in the time domain and to reduce the output data to the right number of uncorrupted samples. After a zero-padding, DFT will be applied to both signals to be transformed back with the IDFT. Figure 2 summarizes the convolution process in the frequency domain.

The optimization of the DFT could be represented by the Fast Fourier Transform (FFT) algorithm that computes the IR just ones for all the successive block elaborations [16].

The IR  $h$  was initially partitioned in a different number  $P$  of equally sized blocks  $s_n$ , as shown in Figure 3.

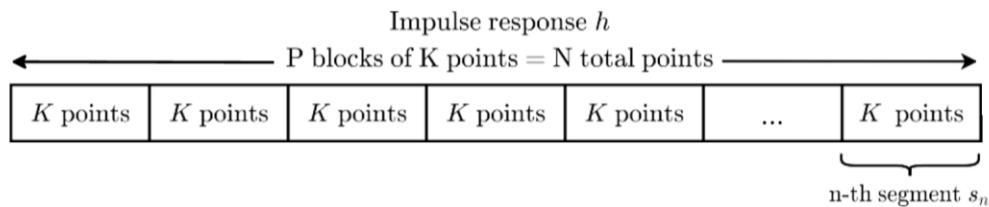


Figure 3: Partitioning scheme of the IR implementation.

Each of these blocks is treated as separate IRs, convolved by FFT windows of length  $L$ . A collection of frequency-domain filters  $S_n$  is obtained with the FFT. The results of the multiplications of these filters with the FFTs are summed and produce the same result as the unpartitioned convolution, by means of proper delays applied to the blocks of convolved data [17]. Figure 4 indicates the whole process.

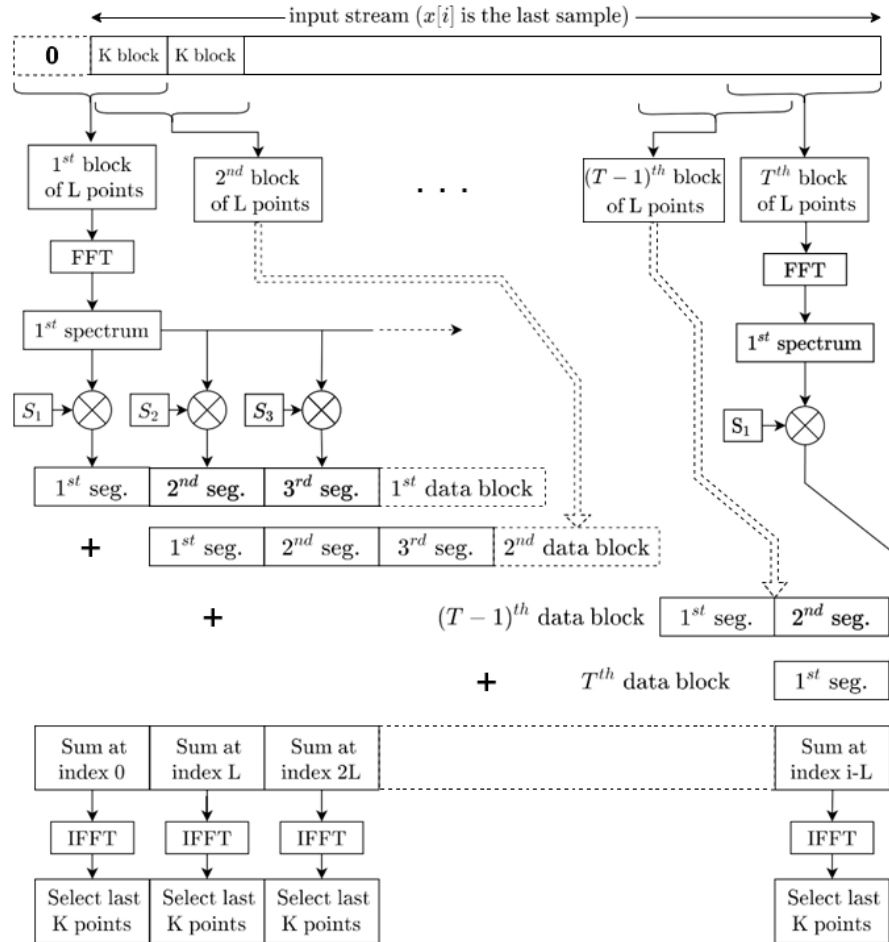


Figure 4: Scheme of the uniform partition of the IR process.

The process described in Figure 4 is less efficient than the unpartitioned overlap-and-save algorithm since it requires a higher number of arithmetic operations and more memory references [18]. Therefore, any new convolution process needs a specific design to find out the right balance between computational cost and I/O delay [17].

### 3.1. Zero-delay convolution algorithm

The computation of the result ( $y_0$ ) via direct-form filtering leaves just enough time to accumulate a block of  $N$  input samples to calculate  $y_1 = x * h_1$  using an  $N$ -point block transform. while the calculation of  $y_0$  proceeds with a second  $N$ -point, the second block output of  $y_1$  can be processed at the same time, which needs a  $2N$ -point block of samples, as shown in Figure 5.

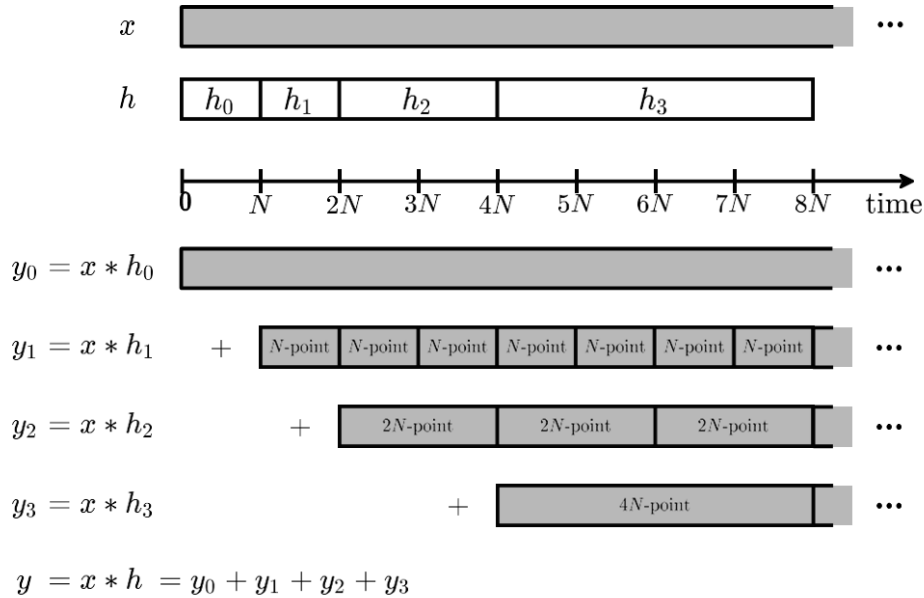


Figure 5: Scheme of minimum-cost partitioning of IR filter with no input-output delay.

Considering  $M$  as a generic length of each block transform convolver, the scheme provided in Figure 5 indicates that each output starts with a delay of  $M$  samples into the IR filter  $h$ . Despite this strategy representing a minimum-cost solution, this algorithm is completely impractical because all the block computations must be done within a sampling period and similarly if the filter partitions number and size increase, the number of simultaneously and intense computations are directly proportional [19]. An alternative solution to this concern consists of enlarging the direct-form convolution partition  $h_0$  enough to give more time to do the computation of the following block convolution  $h_1$ . Each block of size  $M$  samples should start at least  $2M$  samples into the IR filter. This ensures that the processor is working on a particular computation, as shown in Figure 6.

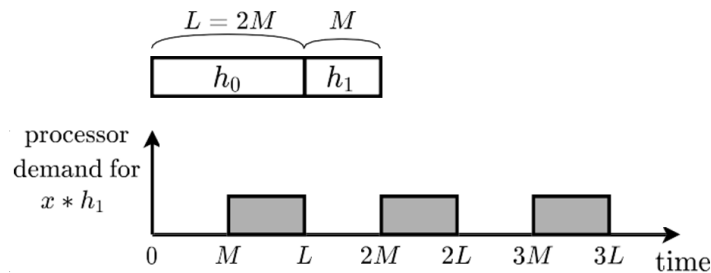
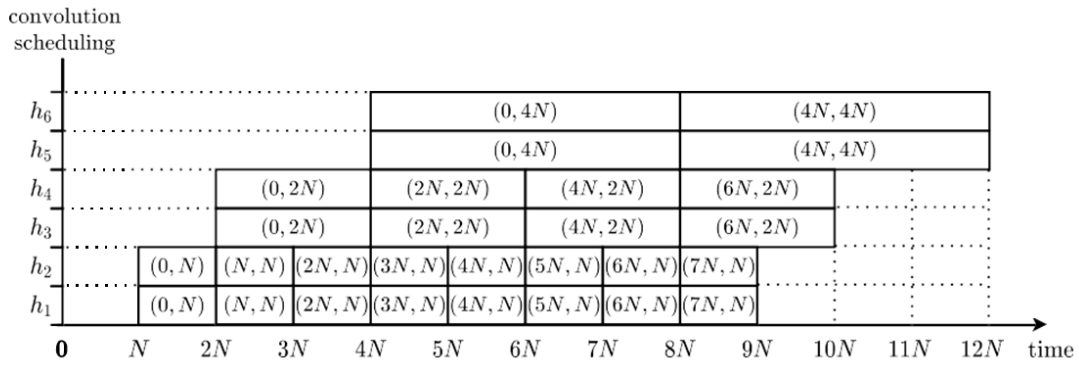
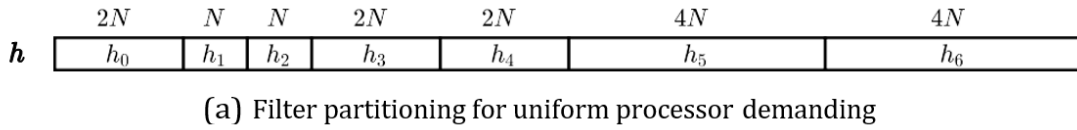


Figure 6: Uniform processing by using  $M$ -point transform block.

On this basis, to keep the processor demand constant over time, the overall filter partition scheme must be redesigned as illustrated in Figure 7, where the first block  $h_0$  is computed by using the direct-form method, while the following blocks are computed by using the block transform method. Based on this scheme, the best optimization is obtained based on the smallest block size where the block transform convolution is more efficient than the direct-form filter.





(b) Block convolution scheduling

Figure 7: General filter partitioning.

The scheme drafted in Figure 7 shows that the processor is uniformly loaded over time, except at the very start of the operation where a transient take-off is necessary.

By doing a comparison between algorithms, there are several ways that the Fourier Transform stage can be optimized. Some of them are listed as follows:

- Pre-calculate the spectra of all filter response blocks;
- Optimize the convolution operation by using real input FFTs, exploring symmetry in the spectral product;
- Reuse input block spectra whenever possible, especially in partitioned algorithms;
- Calculate large input block spectra using the results of smaller input block calculations.

#### 4. PRACTICAL IMPLEMENTATION (MCFX CONVOLVER)

A practical implementation of the algorithms previously shown is the X-mcfx convolver developed from the mcfx plugin for open-source software, in relation to a multichannel approach. The original plug-in has been developed to simplify the creation and playback of surround sound productions by providing user-friendly access to Ambisonic techniques [8]. Generally, the most recent toolkits for spatial audio have been developed in graphical programming environments in a proper Digital Audio Workstation (DAW) containing the Graphical User Interface (GUI). The *mcfx\_convolver* plug-in is a multichannel convolution matrix that uses an efficient non-uniform partitioned FFT convolution algorithm. The GUI allows the selection of the partition size.

The whole algorithm of convolution processing is managed by a custom library with two main classes: a class master (MtxConvMaster) and a class slave (MtxConvSlave). MtxConvMaster main duty is to configure the partitioning of the filters and manage the blocking process. The block processing duty is to act as a direct intermediary between the input and output streams sent from and to the plug-in main code and the slave classes. The MtxConvSlave class contains the real core algorithm that achieves this complex partitioning structure. Each MtxConvSlave elaborates a pre-defined number of equalized partitions of the IR. Class “Master” elaborates as many Slaves as are necessary to fill up the entire IR length. Each slave-related thread provides then a partial convolution result for all the output channels, as indicated in Figure 8.



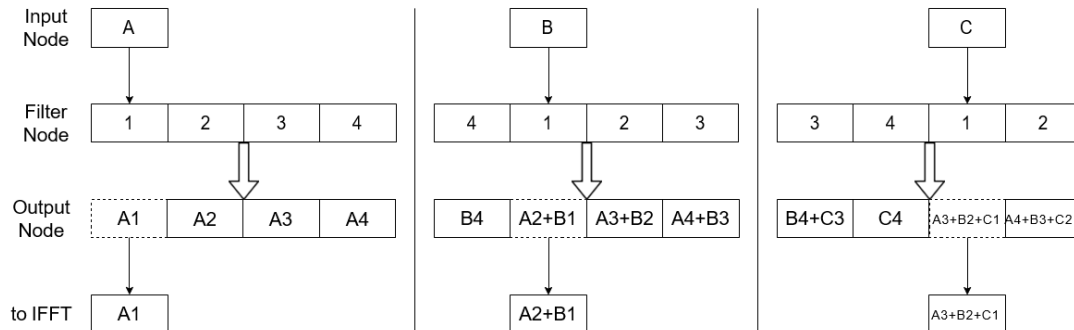


Figure 8: Nodes elaboration with the input blocks,

The thread processes the block with each assigned partition of the IR and adds the elaborated blocks to the output node with the right partition. In the end it transforms back the current block and adds it to the output buffer with the right amount of delay.

The matrix of filters load in `mcfx_convolver` can be done in two ways: with IRs stored in separate files or through packed matrix files. Nonetheless, each IR is referenced to a matrix cell through the information contained in an additional file having a conf format. The IRs in the packed matrix version are arranged in one single wave file with a particular structure. The column of the matrix is summed together, and the resulting array is arranged in the wave file channels. The software will then split every IR assigning the right number of samples.

#### 4.1. X-MCFX convolver

The new X-MCFX convolver includes improvements like the update of the JUCE framework [20] which simplifies the multi-platform binaries building. Furthermore, it develops the recent VST3 plugin format and renovates the GUI. The X-MCFX convolver is also provided with an editable library path, and with a master gain. The partitioning parameters are re-arranged from the original version of the MCFX convolver [21]. All the functional information about the multichannel convolver as the matrix dimensions or the length of the filters is kept permanently on the GUI.

The whole debug window of the MCFX convolver has been replaced by a single row status bar where the loading phases and potential errors are highlighted.

The main improvements are the reintroduction of the original Gardner's scheme which set the number of equal-sized partitions to 2. With this scheme, better efficiency is achieved. The matrices can be loaded just in the packed version and the information needed to load them is requested through the GUI. The required value can also be stored in the metadata chunk of the wave file, so that it elaborates with the matrix file [22]. Figure 9 shows the configuration of the X-MCFX convolver.

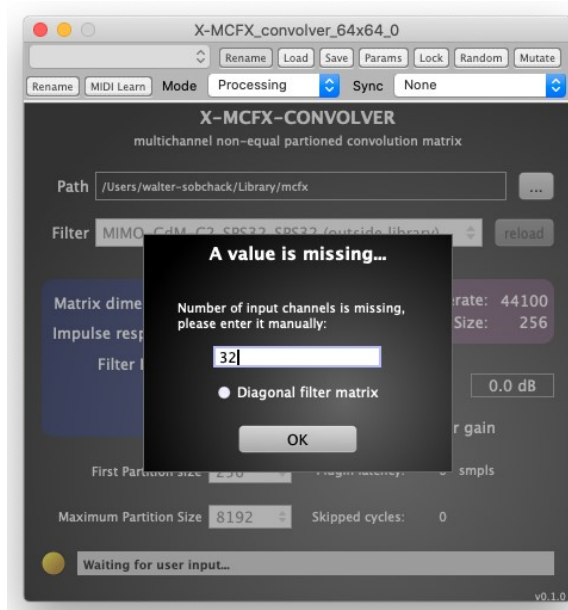


Figure 9: X-MCFX-convolver configuration with the input channels request.

## 5. TESTS AND SIMULATIONS

A specific test has been run to evaluate the performance difference between the new partitioning scheme set by the modified plug-in and the existing MCFX convolver. In particular, a  $32 \times 32$  matrix with 1024 IRs has been loaded, where each filter is composed of 96000 samples at 48 kHz. Both plug-ins have been fed by 32 white noise tracks and different configurations of FPS and MPS have been analysed for both the schemes.

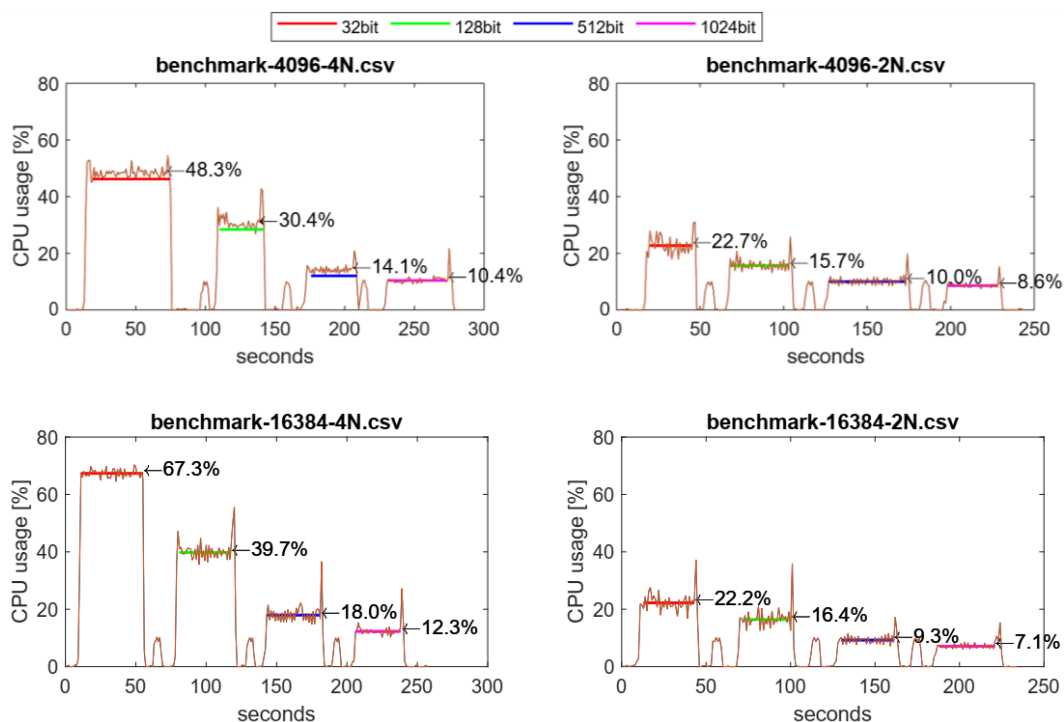


Figure 10: Benchmarks comparison between X-MCFX (Left) and MCFX (Right) convolver.



The results shown in Figure 10 highlight 4 equal-sized partition schemes with the new plug-in. The lowest MPS values brought the best performance in the 4N scheme, although the 2N scheme yields a better overall efficiency with the same configuration.

By considering an MPS value equal to 16384, the efficiency of Gardner’s scheme is more evident between the 4N and the 2N scheme; in fact, a similar result is shown on the 2N graphs while a noticeable difference is highlighted on the 4N graphs at different MPS values, as summarised in Table 1.

Table 1: Partitioning parameters.

Scheme	Maximum Partition Size	Total Partitions	Wasted Samples
4N	4096	$28 + 18 = 46$	1784
	16384	$36 + 2 = 38$	9976
2N	4096	$14 + 20 = 34$	1848
	16384	$18 + 4 = 22$	10040

## 6. CONCLUSIONS

This paper deals with a methodology for standardizing the loading method of IR matrices into multi-channel convolvers, developed in particular for spatial audio applications. The analysis of the existing plug-in has demonstrated an overall efficiency compared to the original setup. The revised version of the MCFX convolver is intended to be used especially for large matrices stored in single files. The main advantages brought by the X-MCFX convolver are the faster configuration of new IR matrices and the better performance of the elaboration. This contributes to creating larger libraries that may involve spatial audio encoding/decoding elaborations for different microphones or speaker setups, spatial audio format conversion, and MIMO characterization. Nonetheless, optimizations other than the partitioning can be implemented for further performance improvements, where future works are addressed.

## 7. REFERENCES

1. Proverbio, A. Bernardini, A. Sarti, A. Toward the wave digital real-time emulation of audio circuits with multiple nonlinearities. *European Signal Processing Conference (EUSIPCO)*, 24, 151-155, August 24-28, 2020.
2. Ge, Z. Li, L. Qu, T. Partially matching projection decoding method evaluation under different playback conditions. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 29, 1411-1423 (2021).
3. Favrot, A. Faller, C. Adaptive non-coincidence correction for A to B-format conversion. *144<sup>th</sup> Audio Engineering Society Convention*, Milan, Italy, May 23-26, 2018.
4. Liang, Y. Ling, B.W.K. Infinite Impulse Response Filter Bank Based Graphic Equalizer Design via Functional Inequality Constrained Optimization and Genetic Algorithm. *IEEE Access*, 9, 65116-65126 (2021).



5. Leglaive, S. Badeau, R. Richard, G. Separating time-frequency sources from time-domain convolutive mixtures using non-negative matrix factorization. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 264-268 (2017).
6. Cariolaro, G. *Classical signal theory*, in *Unified Signal Theory*. London, Springer, 2011, 17–79.
7. Kulp, B.D. Digital equalization using Fourier transform techniques. *Journal of the Audio Engineering Society*, 1988.
8. Gardner, W.G. Efficient convolution without input-output delay. *Journal of the Audio Engineering Society*, **43(3)**, 127–136, (1995).
9. Gerzon, M.A. Ambisonics in multichannel broadcasting and video. *Journal of the Audio Engineering Society*, **33(11)**, 859–871 (1985).
10. De Vries, D. Baan, J. Auralization of sound fields by wave field synthesis. *Journal of the Audio Engineering Society*, (1999).
11. Farina, A. Glasgal, R. Armelloni, E. Torger, A. Ambiophonic principles for the recording and reproduction of surround sound for music. *Journal of the Audio Engineering Society* (2001).
12. Farina, A. Chiesi, L. Measuring spatial MIMO impulse responses in rooms employing spherical transducer arrays. *Journal of the Audio Engineering Society* (2016).
13. Bertet, S. Daniel, J. Moreau, S. 3D sound field recording with higher order Ambisonics - objective measurements and validation of spherical microphone. *Journal of the Audio Engineering Society* (2006).
14. Farina, A. Amendola, A. Chiesi, L. Capra, A. Campanini, S. Spatial PCM sampling: A new method for sound recording and playback. *Journal of the Audio Engineering Society* (2013).
15. Diniz, P.S.R. da Silva, E.A.B. Netto, S.L. *Digital Signal Processing: System Analysis and Design*, 2nd Edition, Cambridge University Press, 2010.
16. Torger, A. Farina, A. Real-time partitioned convolution for Ambiophonics surround sound. *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics* (Cat. No.01TH8575), 195–198, 2001.
17. Armelloni, E. Giottoli, C. Farina, A. Implementation of real-time partitioned convolution on a DSP board. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (Cat. No.03TH8684), 71–74, 2003.
18. Stockham Jr, T.G. High-speed convolution and correlation. *Proceedings of the April 26-28, 1966, Spring Joint Computer Conference*, ser. AFIPS '66 (Spring), Boston, Massachusetts: Association for Computing Machinery, 229–233, 1966.
19. Soo J.S. Pang, K.K. Multidelay block frequency domain adaptive filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(2), 373–376 (1990).
20. Storer, J. Juce application framework. Available online at [www.juce.com](http://www.juce.com) (accessed on 10/12/2020).
21. Kronlachner, M. Ambisonics plug-in suite for production and performance usage. Austria: Linux Audio Conference, Graz, 2013.
22. Farina, A. X-volver vst plugin. Available online at <http://pcfarina.eng.unipr.it/X-volver.htm>, 2010-2020.