IMPLEMENTATION OF REAL-TIME PARTITIONED CONVOLUTION ON A DSP BOARD

E. Armelloni C. Giottoli A. Farina

University of Parma
Industrial Engineering Dept.
Via delle Scienze 181/A, 43100 PARMA, ITALY
enrico.armelloni@unipr.it
angelo.farina@unipr.it

ABSTRACT

Convolution using very long filters is required in order to achieve realistic artificial reverberation or spatial effects. Unfortunately, DSP (Digital Signal Processors) platforms have limited computational power (compared with a modern PC) and consequently it is not possible to design very long filters based on typical time-domain, direct-form algorithms, i.e. FIR or IIR structures. To perform this task, other algorithms are necessary.

In this paper the implementation of a real-time partitioned convolution algorithm on a DSP platform will be demonstrated. In this manner, efficient convolution with long Impulse Responses is attained, with the advantage of low input/output delay.

1. INTRODUCTION

Most multi-channel 3D reconstructions of room reverberation involve convolution between audio signals and multiple long Impulse Responses, an effect that is impossible to achieve using direct-form FIR filters on a DSP platform.

The goal of this paper is to describe the partitioned convolution employed by the authors on an Analog Devices DSP platform, and to discuss its performance in comparison with traditional filtering algorithms.

A DSP is a special digital processor, optimized for performing efficiently in a single clock cycle a number of simultaneous operations: typically a DSP can perform a complete multiply-and-accumulate (MAC) every clock tick.

In order to filter an audio signal correctly in real-time, the DSP must finish all computations during the sampling period, so it will be ready to process the next incoming data sample.

Considering typical audio sampling frequencies, DSP's clock and the number of cycles needed for every MAC, it is easy to understand why the max FIR order is usually low, around 2000 taps, while a satisfactory emulation of a typical concert hall requires at least 80,000 points at 48 kHz sampling rate.

In the following sections, other solutions will be analyzed, in particular frequency-domain convolution algorithms [1,2].

2. DESCRIPTION OF THE ALGORITHMS

A brief explanation of the well-known frequency-domain convolution algorithms is given here, both in unpartitioned and partitioned forms.

2.1. The unpartitioned Overlap-and-Save algorithm

Although this frequency-domain convolution algorithm is not the one employed here, it is useful to review it quickly, as it is fundamental for understanding partitioned convolution, which will be described in the next sub-section.

The convolution of a continuous input signal *x(t)* with a linear filter characterized by an impulse response *h(t)* yields an output signal *y(t)* by well-known convolution integral seen in equation (1).

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{\infty} x(t-\tau) \cdot h(\tau) \cdot d\tau \qquad (1)$$

When the input signal and the impulse response are digitally sampled ($t = i \cdot \Delta t$) and the impulse response has finite length N, such an integral reduces to a sum of products (equation 2).

$$y(i) = \sum_{j=0}^{N-1} x(i-j) \cdot h(j) \qquad (2)$$

The sum of N products must be carried out for each sample, resulting in an enormous number of multiplications and sums.

However, the convolution task can be significantly simplified performing FFTs and IFFTs, because the time-domain convolution reduces to simple multiplications in the frequency domain, between the complex Fourier spectra of the input signal and impulse response. As the FFT algorithm inherently assumes that the analysed segment of signal is periodic, a straightforward implementation of the frequency domain processing produces unsatisfactory results (circular convolution). The periodicity causes by FFTs must be removed from the output sequence, which can be done with the overlap-and-save algorithm [1]. The flow chart in Figure 1 explains the process.

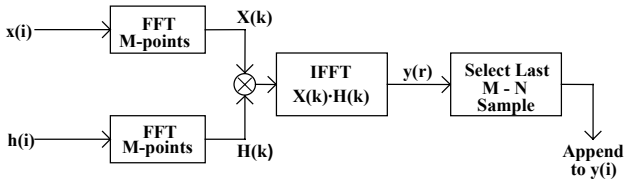

Figure 1: The Overlap-and-Save algorithm.

As the process outputs only $M - N$ convolved data points, the input window of M points must be shifted to the right over the input sequence of exactly $M - N$ points, before performing

the convolution of the subsequent segment. The trade-off is that FFTs of length $M > N$ are required. Typically, a factor of two $(M = 2 \cdot N)$ gives the best efficiency to the overlap-and-save algorithm.

2.2. The uniformly-partitioned Overlap-and-Save algorithm

This algorithm was first proposed on a general-purpose computer by Stockham in 1966 [1], and ported on a DSP chip for audio applications by Kulp in 1988 [2]; recently it was extended to multichannel operation on a PC platform by Torger and Farina [3].

The impulse response h is partitioned in a reasonable number P of equally-sized blocks, where each block is K points long, as shown in Figure 2.

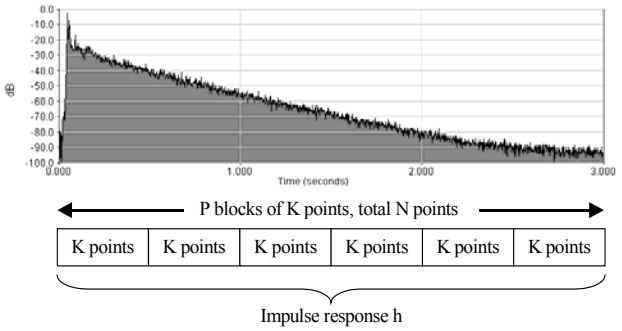

Figure 2: Impulse response partitioning.

Each of these blocks is treated as a separate impulse response, and convolved by a standard overlap-and-save process. Each block is zero-padded to the length L (typically the power of 2 closer to $2 \cdot K$), and transformed with FFT so that a collection of frequency-domain filters S is obtained.

Also the input data are processed in overlapped blocks of L points (each block begins L-K points after the previous).

The results of the multiplications of the P filters S with the FFTs of the latest P input blocks are summed in P frequency-domain accumulators, and at the end an IFFT is done on the content of first accumulator for producing a block of output data (obviously only the latest L-K points of the block have to be kept).

Each block of input data needs to be FFT transformed just once, and similarly a single IFFT is required after frequency-domain summation, so the total number of FFTs is minimized.

The main advantage compared to unpartitioned convolution is that the latency of the whole filtering processing is just L points instead of M. It means that the I/O delay is kept to a low value, provided that the impulse response is partitioned in a sensible number of chunks (8 – 32).

Regarding the computational load, in principle this increases only slightly over the Overlap-And-Save algorithm, as shown by Kulp [2]. In practice, instead, due to the faster memory access available when processing small data structures, the performance can be even greater than with the Overlap-And-Save, as shown in sect. 3.3.

Figure 3 outlines the whole process, for a simple case in which the number of partitions $P = 3$.

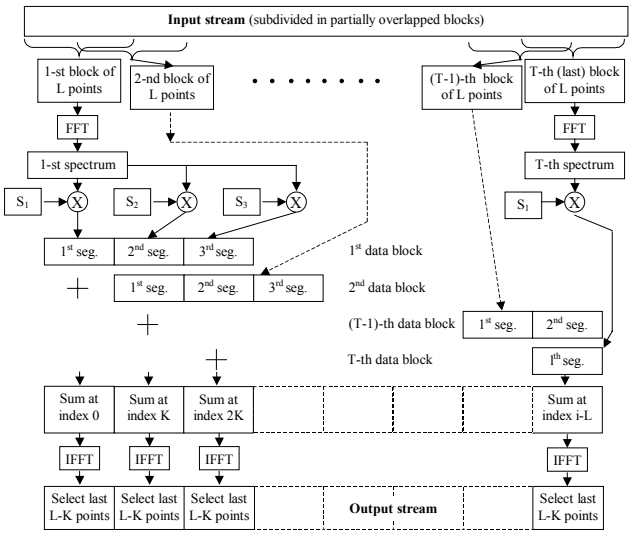

Figure 3: Partitioned convolution.

3. IMPLEMENTATION ON A DSP BOARD

3.1. Features of the DSP platform

An Analog Devices ADDS 21161N Ez-Kit Lite board was chosen for implementation of the partitioned Overlap-and-Save algorithm. Most important features of this board are listed below:

- 100 MHz (10 ns) SIMD SHARC DSP core
- 600 MFLOPS (32-bit floating-point data), 600 MOPS (32-bit fixed-point data)
- Single-cycle instruction execution, including SIMD operations in 2 parallel computational units
- One Mbit on-chip dual-ported SRAM
- Integrated support for SDRAM and SBSRAM external memories

This board provides 4 channels IN and 8 channels OUT through a couple of audio converters (AD1836 and AD1852). On this platform, using a sampling frequency of 48 kHz, a 2000-taps direct-form FIR can be implemented. The corresponding maximum length of the impulse response is just around 40 ms. Using partitioned Overlap and Save, it was possible to convolve impulse responses much longer, as it is demonstrated in following sections.

It must be noted, however, that due to the SIMD architecture of this processor (dual ALU), the 2000-taps direct-form FIR filtering is possible simultaneously on two independent data flows (stereo processing).

3.2. Details of Partitioned Overlap-and-Save code

At the code initialization, the first step is impulse response treatment. The coefficients are downloaded on DSP, and

partitioned into P blocks, where each block is K points length; in this case $K = 4096$. Afterwards, each block is zero-padded to a length of L points (8192), and then transformed with the standard FFT procedure supplied by Analog Devices and stored in the external memory.

A ping-pong I/O buffer was used in the implementation of the partitioned Overlap-and-Save algorithm over DSP platform.

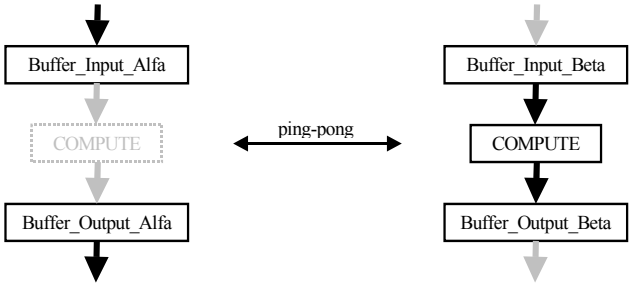

Figure 4: Input and output Ping-Pong buffers.

Figure 4 shows how the input data are going from the A/D converter to Buffer_Input_Alfa, while the processed samples are going from Buffer_Output_Alfa to the D/A. In the meantime, the filtering of the Buffer_Input_Beta is occurring, and the results are being written in Buffer_Output_Beta.

When the input buffer is full, "Ping-Pong" occurs, and the Alfa and Beta buffers are switched. Now Buffer_Input_Beta will be filled with the incoming data, and the previous block of data, stored in Buffer_Input_Alfa, will be processed by the block labelled "COMPUTE". Processed samples, contained in Buffer_Output_Beta, will be sent to the output.

In order to filter correctly, the processing task, performed by block "COMPUTE", must be finished before the "Ping-Pong" (buffer switch) occurs.

Each block is long 4096 points. But the FFT has a length $L = 8192$ points, and consequently a larger data chunk, constituted by the last and the previous input blocks, is processed.

The first step of the filtering procedure is clearly shown in Figure 5, where FFT[0] is the FFT transform of the processing stream and Filter[i] are P blocks ($P = 4$) containing FFT transforms of the impulse response blocks. FFT[0] block is multiplied for each Filter[i] blocks. Results (A_1, A_2, A_3, A_4) are stored in an apposite circular "Computation Buffer", the leftmost block is antitransformed and the last $L - K$ (that is 4096) points of it are sent to Buffer_Output_Beta.

In the second step, as shown in Figure 6, resulting blocks B_i (i.e. B_1, B_2, B_3, B_4) are added to the "Computation Buffer". After shifting the pointer to the accumulation blocks, the first P - 1 of them are added to the contents already saved in the blocks of the circular buffer, and the last one overwrites the block previously sent to Buffer_Output_Beta (in practice, the content of this accumulation block is zeroed before summing B_4). The content of the accumulation block where B_1 was summed is subject to IFFT and the last $L - K$ points of it are sent to Buffer_Output_Alfa.

The process is repeated again for the next blocks, at each step the pointer to the "current" accumulation block is increased, and, being a circular buffer, after 4 steps it points again to the first block.

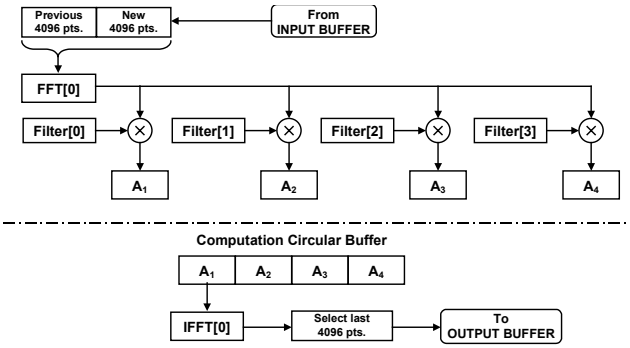

Figure 5: Step 1.

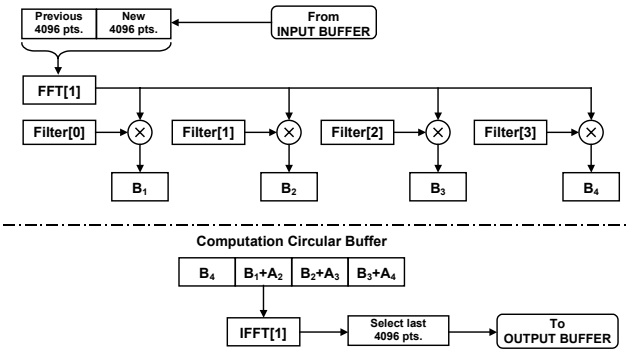

Figure 6: Step 2.

3.3. Performance

The Partitioned Overlap-and-Save algorithm was implemented for a DSP board configured under several sets of conditions. The performances that were reached are listed in Table 1.

Ch IN	Ch OUT	Number of blocks P	IR length
1	1	27	110,592
2	2	11	45,056
2	4	5	20,480
4	8	2	8,192

Table 1: Partitioned Overlap-and-Save algorithm performance on Analog Devices 21161N DSP board.

The impulse responses treated with this algorithm are much longer than 2000 points, the number of taps attainable with a traditional FIR filter. An impulse response of 110,592 points, at a sampling rate of 48 kHz, is 2.3 seconds long. For many cases, also a 45,056 points impulse response is long enough, making it possible to employ a single DSP board for processing a stereo signal. The performance in 2x2 mode (4 filters) is far in excess than the requirements for good cross-talk canceling filters, which are typically long around 4096 taps. These filters had to be truncated to 512 taps each when implemented with the direct-form time-domain FIR structure, causing sub optimal performances.

Other considerations include the efficiency and the input/output delay (latency). Tests performed employing the

DSP board demonstrated that the maximum efficiency is reached when the overlap between two input streams is around half of the FFT length, L (Figure 7).

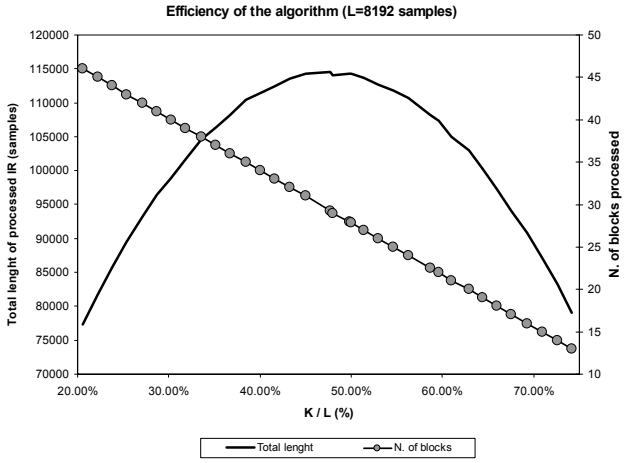

Figure 7: Efficiency as function of K / L

About the I/O delay, using L = 8192 points, and a sampling frequency F_s = 48 kHz, a latency of 170 ms is achieved.

4. APPLICATION TO SURROUND SOUND

The ability to manage long impulse responses is an important step in order to implement an Ambiophonics surround system [3,4,5] over DSP boards. This system, pioneered by Ralph Glasgal, is based on a frontal stereo-dipole, some surround speakers and an optional rear stereo-dipole.

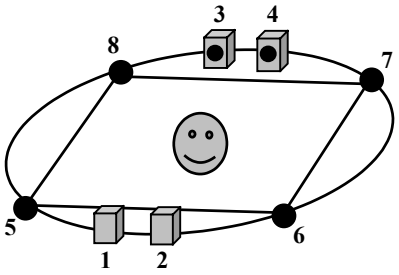

Figure 8: Ambiophonics' 2D geometry: 2 stereo dipoles, 4 surround speakers.

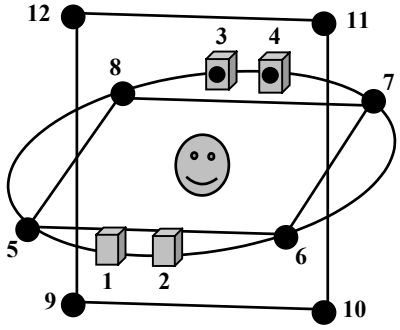

Figure 9: Ambiophonics' 3D geometry: 2 stereo dipoles, 8 surround speakers.

A simple scheme is shown in Figure 8, where loudspeakers 1,2 represent the frontal Stereo-dipole, loudspeakers 3,4 represent the rear Stereo-dipole and loudspeakers 5,6,7,8 are the "surround".

This geometry can be implemented using three DSP boards, one for the two stereo-dipoles (running two cross-talk canceling filters) and two for driving reverberation loudspeakers (by convolution with long room impulse responses).

A further step, to improve reproduction performances, is to add 4 more "surround" speakers to the geometry, as seen in Figure 9, but this requires the use of other two DSP boards.

5. CONCLUSIONS

This paper shows how to implement the uniformly-partitioned Overlap-and-Save algorithm over a low-cost DSP platform. In the currently available implementation, it provides higher throughput than the traditional unpartitioned overlap-and-save and than the non-uniform partitioned convolution algorithms [6], the theoretically cheaper alternatives.

In [3] it was already shown how the same algorithm outperforms the others also on a standard PC, thanks to the advanced architecture of modern CISC processors.

At this point, the choice between DSP and PC is due only to evaluation of the cost of the hardware (the DSP board employed here is around 360 USD, while a PC equipped with an 8-channels sound card of equivalent quality costs at least 1500 USD), and of the computing power required (a PC can process something as 48 convolutions of 110,592 taps each, the DSP just one). These figures show that nowadays the use of a DSP is sensible only when it is required to implement a light, compact system and with little number of channels, otherwise a PC provides a significantly better price/performance ratio.

6. ACKNOWLEDGEMENTS

This work was supported economically and technically by the Ambiophonics Institute, founded by Ralph Glasgal.

7. REFERENCES

[1] T. G. Stockham Jr., "High-speed convolution and correlation", *Proc. 1966 Spring Joint Computer Conf.*, AFIPS, Vol 28, 1966, pp. 229-233.

[2] Barry D. Kulp – "Digital equalization using Fourier Transform techniques" – *Pre-Prints of the 85th AES Convention*, Los Angeles, 3-6 November 1988.

[3] A. Torger, A. Farina, "Real-Time Partitioned Convolution for Ambiophonics Surround Sound", *2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, 21-24 October 2001.

[4] R. Glasgal, K. Yates, "Ambiophonics – Beyond Surround Sound to Virtual Sonic Reality", *Ambiophonics Institute*, 1995.

[5] A. Farina, R. Glasgal, E. Armelloni, A. Torger, "Ambiophonic Principles for the Recording and Reproduction of Surround Sound for Music", *19th AES Conference*, Scholss Elmau, Germany, 21-24 June 2001.

[6] W.G. Gardner, "Efficient convolution without input-output delay", *Journ. AES* vol. 43, n. 3, 1995 March, pp. 127-136.