Audio Engineering Society

Convention e-Brief 88

Presented at the 134th Convention
2013 May 4–7 Rome, Italy

Workload estimation for low-delay segmented convolution

Malte Spiegelberg

HAW Hamburg, Hamburg, 22081, Germany
Malte.Spiegelberg@haw-hamburg.de

ABSTRACT

Zero-delay convolution usually follows a hybrid approach with convolution processing steps in both, the time and the frequency domain (Gardner, J. AES 43 (1995) [1]). Implementations are likely to ask for dynamic coding, and related workload estimations are focused on efficiency and are limited to the hybrid approach. This paper considers simpler implementations of segmented convolution that work in the frequency domain only and that achieve acceptable low delay for real-time applications when processing several seconds of impulse-response in FIR mode. Workload and memory demand are estimated for this approach in the context of likely application parameters.

1. INTRODUCTION

This paper considers the task of convolving an input signal with another signal, say a room impulse response of considerable length. To solve this computational task, here, the input and the impulse response are divided into segments and are zero-padded to prepare for convolution. The segment size is chosen in context of application.

2. NOTATION

The following variables are assigned:

$x[n]$	Input signal
L_x	Length of $x[n]$
N	Length of a segment $x_k[n]$
$h[n]$	Impulse response
L_h	Length of $h[n]$
P	Length of a segment $h_i[n]$
$y[n]$	Output signal
L_y	Length of $y[n]$

Capital letters are used for signals in the frequency domain, so $X[f]$ is the Fourier transformed input signal $x[n]$.

3. SEGMENTED CONVOLUTION

In signal processing tasks impulse responses of considerable length are occasionally convolved with other signals. The related computational effort can be lowered when processing large signal segments in the frequency domain. Real-time applications, however,

may dictate use of small segments. Therefore, input signal and impulse response are divided into segments. In this way, the output can start before the whole input signal is known.

When a segment $x_k[n]$ is loaded, it is transformed into the frequency domain by FFT. The transformed segment $X_k[f]$ is then multiplied with every segment $H_i[f]$ of the transformed impulse response. $h[n]$ can be segmented and transformed during system boot so that transformed segments $H_i[f]$ are present in memory. The result of the multiplication $Y_{ki}[f]$ is transformed back into time domain by the inverse FFT, IFFT. Finally, an overlap-add of all segments $y_{ki}[n]$ results in the output signal $y[n]$ [Fig. 1][2].

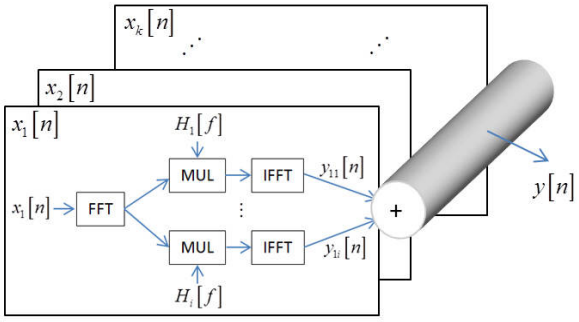
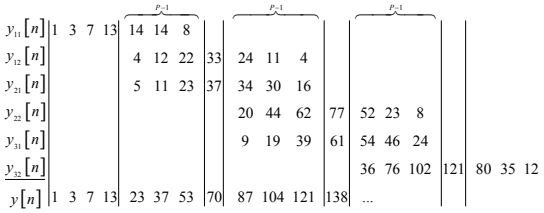


Figure 1 The process of segmented convolution

The size P of the segments $h_i[n]$ is an important parameter in the calculation. When P is smaller than N, the number of terms in the final overlap-add calculation increases unfortunately, computational effort will increase. When P is larger than N, a segment $x_k[n]$ and its resulting segments will stay in the process longer than necessary, memory demand will increase. Hence, the most efficient method is using $N = P$.
A tutorial example with very short segment length illustrates input, segmentation and overlap-add calculation of convolved segments (1).

$$x[n] = [1, 2, 3, ..., 31, 32]$$
$$h[n] = [1, 1, 2, 2, 4, 4, 2, 1]$$

$$N = P = 4 \qquad (1)$$
$$x_k[n] = \begin{cases} x[n] & if\ 4(k-1) \le n \le 3k \\ 0 & else \end{cases}$$
$$h_i[n-(i-1)4] = \begin{cases} h[n], & if\ 4(i-1) \le n \le 4i-1 \\ 0 & else \end{cases}$$



4. WORKLOAD ESTIMATION

This section models computation effort to finally suggest optimal $N = P$ in the context of application demand. The process of segmented convolution takes four steps of processing:

- FFT of an input segment $x_k[n]$
- Multiplication with the segments $H_i[f]$ of the transformed impulse response
- IFFT of the resulting segments $Y_{ki}[f]$
- Overlap-add of the segments $y_{ki}[n]$

4.1. FFT of input segments $x_k[n]$

The length of the result of an input segment $x_k[n]$ of length N convoluted with the impulse response segment $h_i[n]$ of length P is $N + P - 1$. Therefore input segment and impulse response segment have to be zero-padded to length $N + P - 1$ before FFT calculation. For this workload estimation, the Radix-2 Algorithm will be used. This will request a total input length of $M = 2^n$. Therefore N should match P and zero-padding targets the length $M = 2N$. There will be $M / 2$ complex multiplications and M complex additions [3, p. 23]. These complex operations will cost as $5M$ flops considering that a complex addition will need two real additions and a complex multiplication will need three real multiplications and three real additions [3, p. 15]. The effort for one segment $x_k[n]$ will be

$$W_1 = 5M \cdot \log_2(M) \qquad (2)$$

flops.

4.2. Multiplication with the segments $H_i[f]$

The impulse response of length L_h is split into L_h / P segments with the same amount of segments after transformation. The result of every of the FFTs

calculated before is multiplied with every segment $H_i[f]$. This will need $12 * L_h$ flops for one segment $X_k[f]$ to obtain the result $Y_{ki}[f]$.

4.3. IFFT of the resulting segments $Y_{ki}[f]$

All segments $Y_{ki}[f]$ have to be transformed back into time domain with the IFFT operation. IFFT and FFT have the same effort. So for every segment $Y_{ki}[f]$ there will be

$$W_2 = \frac{L_h}{P} \cdot \left(5M \cdot \log_2(M)\right) \quad (3)$$

flops.

The calculations shown have to be done for each segment of $x[n]$. When $x[n]$ is split into segments of length N, there are L_x / N segments. Adding this factor and taking the estimations before, the workload estimation before the overlap-add can be calculated. It will be

$$W_T = \frac{L_x}{N} \cdot \left(12 \cdot L_h + \left(\frac{L_h}{P} + 1\right) \cdot 5M \cdot \log_2(M)\right) \quad (4)$$

flops.

4.4. Overlap-add of the segments $y_{ki}[n]$

An overlap-add calculation only requires real additions and the amount of terms depends on the number of segments in $x[n]$ and $h[n]$. This number grows from the beginning of the operation to a maximum of $L_x + L_h - 1 - 2P$ terms for each segment $y[n]$, see examples. The workload of the overlap-add calculation can be described for the whole calculation. For the estimation, the whole calculation is considered to be at maximum overlap. The total amount of additions for the overlap therefore is

$$W_{over} = \left(\frac{L_x}{N} + \frac{L_h}{P} - 2\right) \cdot \left(\left(2 \cdot \frac{L_h}{P} - 1\right) \cdot (P-1) + \frac{L_h}{P} - 1\right) \quad (5)$$

flops.

4.5. Estimation Results

The following table provides data for different sample rates and impulse response lengths see Table 1. L_x is set to one second according to the sample rate. Therefore, the numbers represent operations per second. The line with $N = P = 1$ represents discrete convolution, the last two lines represent the convolution in the spectral domain with Fourier transformations across the entire sound.·

	L_h = 2 Sec. @ 44.1 kHz	L_h = 0.5 Sec. @ 48 kHz
$N = P$	$W_T + W_{over}$	$W_T + W_{over}$
1	$1.56 \cdot 10^{10}$	$1.15 \cdot 10^{9}$
8	$2.53 \cdot 10^{10}$	$7.49 \cdot 10^{9}$
16	$1.51 \cdot 10^{10}$	$4.47 \cdot 10^{9}$
32	$8.75 \cdot 10^{9}$	$2.59 \cdot 10^{9}$
64	$4.99 \cdot 10^{9}$	$1.48 \cdot 10^{9}$
128	$2.80 \cdot 10^{9}$	$8.32 \cdot 10^{8}$
256	$1.55 \cdot 10^{9}$	$4.63 \cdot 10^{8}$
44100	$2.28 \cdot 10^{7}$	
48000		$1.22 \cdot 10^{7}$

Table 1 Workload estimation for two commonly used sample rates, $N = P = 1$ represents discrete convolution in the time domain

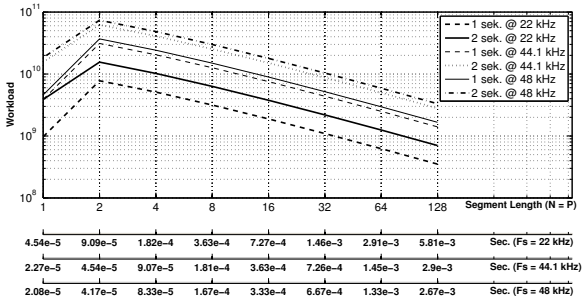


Figure 2 Workload estimation versus segmentation length $N = P$, workload represents the total flops $W_T + W_{over}$, amended scales represent processing delay in relation to segmentation length and sample rate.

5. MEMORY ESTIMATION

Implementation of segmented convolution will require memory in a predictable way. Memory estimation takes into account that memory can be overwritten as soon as a segment $x_k[n]$ is no longer part of the calculation. The transformed segments $H_i[f]$ of the impulse response will be needed for the whole calculation so these cannot be overwritten.

It is possible to estimate the memory needed for the whole calculation. As maximum memory should be estimated, the sectors with maximum overlap are considered. At this point of the calculation $2 \cdot L_h / P$

segments are active in the calculation. Another L_h / P segments have to be prepared for the next sector, so memory is also required for these segments. The total amount of samples to be stored in memory is:

$$\underbrace{\left(\frac{2\cdot(2P-1)\cdot L_h}{P}\right)}_{H_i[n]} + \left[\underbrace{2\cdot(2N-1)}_{\text{segment } X_k[n]}\cdot \underbrace{\frac{L_h}{P}}_{\substack{\text{loading next}\\ \text{segments}}}\right] + \left[\underbrace{2\cdot(2N-1)}_{\text{Segment } Y_j[n]}\cdot \underbrace{\frac{2\cdot L_h}{P}}_{\text{maximum overlap segments}}\right]\cdot S\,Bit \quad (6)$$

And for N = P:

$$(2P-1)\cdot\frac{8L_h}{P}\cdot S\,Bit \quad (7)$$

For instance, with the examples from section 4 [Tab. 1, Fig. 2], a second of impulse response at 44,1 kHz and $N = P = 64$ needs $7 \cdot 10^5$ samples in memory, and an extreme case, two seconds at 48 kHz and $N = P = 2$ needs $1{,}15 \cdot 10^6$ samples.

6. OPTIMIZATION

There are further options to reduce processing power or memory demand by proposed algorithms, even before processor-specific optimization.

For instance, the calculation can optionally be optimized by a factor of two when using an interleaving of the input signal segments. Two segments $x_1[n]$ and $x_2[n]$ form a complex value $z_1[n]$ that is transformed and multiplied with all transformed segments of $h[n]$. The result $ZE_1[f]$ is then transformed back into the time domain. The segments $y_{ki}[n]$ are the real and imaginary part of the results $ze_j[n]$ [4].

Or alternatively, the process can be improved by using other FFT algorithms. Radix 4 and Split-Radix are two examples of common FFT-Algorithms that can reduce workload. An optimized FFT method for low latency convolution is presented in [5].

Concerning data format, working with floating point is recommended as coding will be simpler compared to fixed point usage.

7. SUMMARY

Workload and memory demand are estimated for the segmented convolution in the spectral domain. The approach translates parameter size such as sample size and sample rate to suggest window size and related delays for individual applications. Feasibility can be evaluated before implementation on specific processors.

8. REFERENCES

[1] William G. Gardner, "Efficient Convolution without Input-Output Delay" in *Journal of the Audio Engineering Society*, Vol. 43 (3), 127-136 (1995)

[2] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997

[3] Eleanor Chu and Alan George, *Inside the FFT Black Box: Serial and Parallel Fast Fourier Transform Algorithms*, CRC Press, 2010

[4] Udo Zölzer, *Digitale Audiosignalverarbeitung*, Vieweg+Teubner Verlag, 2005

[5] Jeffrey Hurchalla, "A time distributed FFT for efficient low latency convolution," *Audio Engineering Society Convention 129*, 2010