

# 多チャンネル音場制御のための 低遅延畳み込みアルゴリズムの実装とその性能評価\*

☆北川雄一（東京電機大学），伊勢史郎（東京電機大学/JST, CREST）

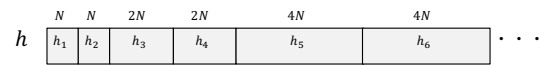
## 1 はじめに

近年，音場シミュレータなどリアルタイムで大規模な畳み込み演算を行う信号処理技術の需要が高まっている[1]。リアルタイム畳み込み演算はDSPなど信号処理専用の演算装置を用いて実現することができるが，室内の残響を含む場合など数秒に至るインパルス応答を畳み込む場合には大規模メモリへのアクセス速度がボトルネックとなり，DSPのもつ高速な演算性能を引き出すことができない。一方，近年の汎用CPUを搭載するPCは大規模メモリに対する高速アクセスを高度なキャッシュ構造により実現し，さらに複数コアを有することによりDSPの演算性能に迫りつつある。そこで本研究ではPCを用いて多チャンネル音場制御のためのリアルタイム畳み込みを低遅延で行う装置を実現するために，異なるFFTポイント数で周波数領域畳み込み演算を行うアルゴリズムを実装し，遅延量，畳み込み長，PC性能などの関係を調査する。

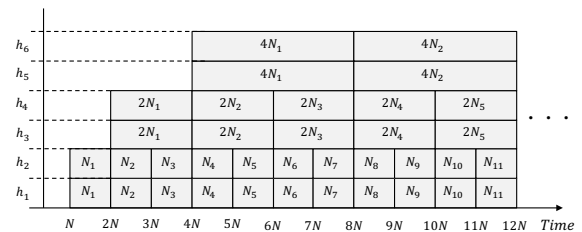
## 2 畳み込みアルゴリズムの実装

畳み込み演算を低遅延で実現するアルゴリズム[2]を用いる。すなわち Fig.1 のように畳み込みに使用するフィルタ係数の初期部分を小さなブロックに分割し，後期になるにつれて大きなブロックに分割する。分割したフィルタ係数を各ブロック毎に周波数領域で畳み込み，出力結果を加算する。Fig.1(a)に分割されたフィルタ係数の一例を示す。この場合，フィルタ係数 $h$ の長さは  $14N$  であり，各ブロックのフィルタ係数長はそれぞれ  $N, N, 2N, 2N, 4N, 4N$  と2つの同じ幅のブロックを並べ2倍に増やしていく。さらに大きなフィルタ係数長を分割する場合は， $8N, 8N, 16N, 16N$  と分割数を増やせばよい。Fig.1(b)に各ブロックの畳み込み演算を実行する工程表を示す。例として，Fig.1(b)における $h_1$ は，ブロック幅 $N$ 分の入力信号をバッファリングする必要があり，さらに時間  $2N$  までに計算を完了

する必要がある。その後，次のブロックの計算を開始する。このようにプロセッサは各ブロックにおいて工程表をもとに反復計算を行う。また， $h$ の偶数ブロックでは，初期のバッファ処理を均一に行うためブロック幅分を前方シフトし，計算を終えてからブロック幅分を後方シフトする。



(a) フィルタ係数の分割



(b) 畳み込みの工程表

Fig.1 低遅延畳み込み演算のブロック構成

低遅延畳み込みアルゴリズムを PC に実装するため， Visual C#および Intel C++によりアプリケーションソフトを開発した。フィルタ係数を分割する際の最小ブロック幅 $N$ はオーディオデバイスのバッファサイズと等しい値とした。畳み込み長はフィルタ係数の分割数  $FFTLevel$  を用いて式(1)により求められる。

$$Convolution_{length} = \{(2^{FFTLevel} - 1)\} \times N \quad (1)$$

同じフィルタ係数長をもつブロックの畳み込み演算を一つのスレッドに割り当て，マルチコアプログラミングにより複数のブロックを並列に演算し，スレッド同期は行わずに畳み込み演算が終わり次第，結果を加算した。

## 3 評価実験

### 3.1 実験条件

開発した畳み込みアルゴリズムを PC に実装し，性能評価を行った。畳み込み演算用 PC を

\*Implementation and Performance evaluation of Low-latency convolution algorithm for Multi-channel sound field control. by KITAGWA Yuichi (Tokyo Denki University), ISE Shiro (Tokyo Denki University/CREST, JST)

オーディオインターフェース(RME HDSPe MADIFX, Fs=48kHz)経由で測定用 PC に接続し、畳み込み演算用 PC のインパルス応答を計測し、その演算精度及び遅延時間を評価する。演算精度の評価式を式(2)に示す。

$$SNR = 10 \log_{10} \frac{\sum |p(n-m)|^2}{\sum |p(n-m) - p'(n)|^2} \quad (2)$$

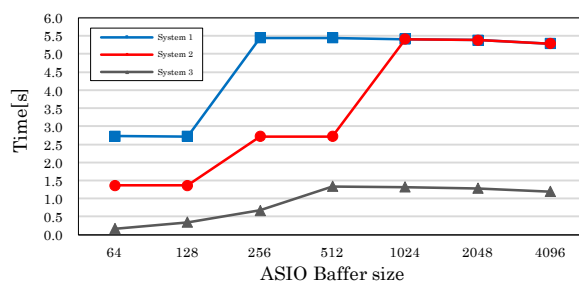
ここで、 $p(n-m)$ は元のフィルタ係数を $m$ サンプル遅延させた信号、 $p'(n)$ は測定した畳み込み演算 PC のインパルス応答である。またフィルタの構成は入力チャンネル数を 1, 出力チャンネル数を 96 とし、96 個のインパルス応答のチャンネルごとの SNR を算出した。フィルタへの入力信号はランダムノイズを用いた。これらの条件をもとに 3 台の異なる性能の PC (Table 1) を用いて評価を行った。

### 3.2 実験結果

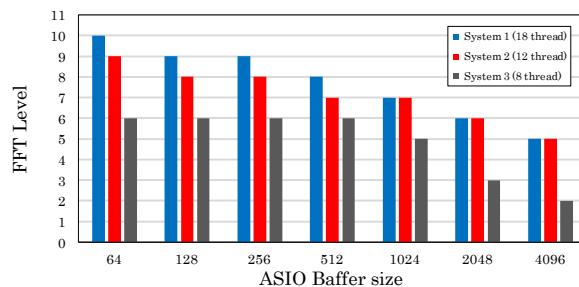
バッファサイズ、畳み込み長、CPU のスレッド数を変えながら畳み込み演算の可否、可能である場合その遅延時間を計測した。SNR が 100dB を超えた場合に畳み込み演算可能と判断する。バッファサイズと畳み込み長の関係を Fig.2(a)に示す。System1 ではバッファサイズが 256 以上、System2 ではバッファサイズが 1024 以上で約 5.5 秒の畳み込み演算が可能となり、System3 は約 1.2 秒の畳み込み演算が上限となる。System1 と System2 を比較するとバッファサイズが 64 から 512 にかけて約 2 倍の性能差が確認できる。これは System1 が System2 に比べて高速に信号を伝達できる Chipset を使用しており、それに伴い Memory も高速に信号を伝達できる DDR4 を使用しているためである。次にバッファサイズと FFT Level の関係を Fig.2(b)に示す。System1, 2 はバッファサイズが大きくなると計算可能な FFTLevel が小さくなる。System3 はバッファサイズが 512 以下では FFTLevel が 6 にとどまる。これは System3 のコア数が System1,2 に比較して小さくなり、畳み込みに使用できるスレッド数が限られるためである。以上から、より長いフィルタを畳み込むには CPU のコア数が重要となる。次にバッファサイズと遅延時間の関係を Fig.2(c)に示す。バッファサイズが大きくなると畳み込み演算に要する遅延時間が大きくなる。

Table 1 畳み込みシステム

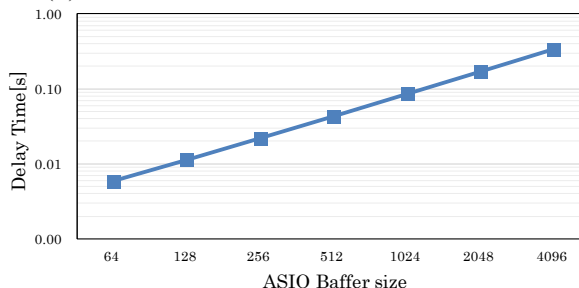
		System 1	System 2	System 3
CPU	型番	Intel core i7-5960X	Intel Core i7-4960X	Intel Core i7-860
	動作周波数	3.0GHz	3.6GHz	2.8GHz
	コア数 (スレッド数)	8 (16)	6 (12)	4 (8)
Chipset		Intel X99 Express	Intel X97 Express	Intel P55
Memory		64GB DDR4	64GB DDR3	8GB DDR3
OS		Windows 7 Professional	Windows 7 Professional	Windows 7 Professional



(a) バッファサイズと畳み込み長の関係



(b) バッファサイズとスレッド数の関係



(c) 遅延時間

Fig.2 実験結果

## 4 おわりに

フィルタ係数を異なる FFT 長で分割することにより低遅延化した畳み込みアルゴリズムを PC に実装し、PC の性能による畳み込みの演算精度、遅延時間を検証した。市販の PC を用いて 96 チャンネル、フィルタ係数長 5 秒以上の大規模な畳み込み演算をリアルタイムかつ低遅延で実現できることがわかった。

### 参考文献

- [1] 渡邊他, TVRSJ vol.20 No.1, 2015
- [2] J. Audio Eng. Soc., Vol 43, No. 3, 1995 March