A-79

高速FIRフィルタリング算法を利用した射影法

Projection algorithm using fast FIR filtering techniques

田中 雅史　牧野 昭二　金田 豊
NTT ヒューマンインタフェース研究所
NTT Human Interface Laboratories

1. はじめに

近年提案されている高速射影法[1][2]の演算量は適応フィルタの次数を L、射影の次数を p とすると、約 $2L+20p$ であり、演算量 $2L$ の NLMS(学習同定法) とほぼ同程度の演算量の少ない手法といえる。しかし、音響エコーキャンセラのようにフィルタ長 L が数百、数千にもなる応用ではさらなる演算量の削減が要求される。

本報告では、高速 FIR フィルタリング算法[3]を射影法に導入することで、さらに演算量を削減する方法を示す。この提案法では、高速 FIR フィルタリング算法がブロック処理を行なうので推定誤差の出力が遅れるが、その他の収束特性は逐次処理に基づくオリジナルの射影法[4]の性能が保たれる。

2. 高速FIRフィルタリング算法を利用した射影法

高速 FIR フィルタリング算法は定数係数フィルタによる FIR フィルタリングを高速 (=低演算量) に行なう手法である。高速 FIR フィルタリングを行なう 1 つの方法として、 FFT を用いたものがよく知られている。この場合、計算効率を保つためには、1 度に使用する入力信号の区間 (ブロック長) を FIR フィルタの長さと同程度にする必要があり、そのために、大きな遅延が生じるという欠点がある。一方、本稿で検討対象とした文献[3]の高速 FIR フィルタリング算法は、短いブロック長に対しても演算量削減効果があるのが特徴である。

今回提案する射影法のアルゴリズムを Table に示す。アルゴリズムの基本は文献[1][2]の高速射影法であるが、射影法に高速 FIR フィルタリング算法を適用するために、ブロック長 $N(>1)$ 毎に 1 度、推定誤差の計算 (手順 1) とフィルタ係数の更新 (手順 8) を行う。また、それとは別に、逐次更新の場合と同じ推定誤差を得るために、高速 FIR フィルタリング算法で計算した推定値を補正する (手順 3, 4)。この手法は、ブロック処理のために、推定誤差の出力がブロック長だけ遅れるが、逐次更新するオリジナルの射影法と同じ推定誤差を得る。

文献[3]の高速 FIR フィルタリング算法の演算量はフィルタ長 L、ブロック長 N などに依存する。$L=NM$、$N=2^n$ となる場合には、Table の手順 1,8 に要する 1 時刻当たりの積和演算の回数は $2(2(3/2)^n-1)M$ となる[3]。また、手順 2 から 7 が要する積和演算は $2N+20p$ である。$L=1024$, $N=32$, $p=16$ の場合、積和演算は 1292 回であり、文献[1][2]の場合の約半分となる。

3. まとめ

高速 FIR フィルタリング算法を利用した射影法を示した。この手法により演算量が大幅に削減される。また、収束特性は小さな遅延を除いて逐次処理に基づくオリジナルの射影法に同じである。

参考文献

[1] M. Tanaka et al.: Proc. ICASSP-95, pp. 945-948, (1995-5)

[2] S. Gay and S. Tavathia: Proc. ICASSP-95, pp. 3023-3026, (1995-5)

[3] J. Benesty and P. Duhamel: IEEE Trans. on Signal Processing, vol. 40, no.12, pp. 2904-2920, (1992-12)

[4] 尾関, 梅田: 信学誌 (A), J67-A, pp. 126-132, (1984-2)

Table: 高速 FIR フィルタリング算法を利用した射影法

記号の定義 (下付き文字はベクトルの次元を表す。)

k: 時刻, $x(k)$: 入力信号, $y(k)$: 目的信号, $e(k)$: 推定誤差信号

L: 適応フィルタ長, p: 射影の次数, N: ブロック長

$\hat{\mathbf{h}}_L(k)$: 適応フィルタ係数からなるベクトル

$\mathbf{x}_L(k)$: $\mathbf{x}_L(k) = [x(k), \ldots, x(k-L+1)]^T$

$\mathbf{z}_L(k)$: 次式で定義され、$\hat{\mathbf{h}}_L(k)$ の代わりに更新する中間変数。

(推定誤差 $e(k)$ は近似なく求まる。)

$$\mathbf{z}_L(k) = \hat{\mathbf{h}}_L(k) - \sum_{i=1}^{p-1} \mathbf{s}_p(k-1)_i \mathbf{x}_L(k-i)$$

$\mathbf{s}_p$ は手順 7 参照。$\mathbf{s}_p(k-1)_i$ は $\mathbf{s}_p(k-1)$ の i 番めの要素。

$\mathbf{e}_p(k)$: $\mathbf{e}_p(k) = [e(k), \ldots, (1-\mu)^{p-1} e(k-p+1)]^T$

μ: ステップサイズ $(0 \le \mu \le 2)$

$\mathbf{g}_p(k)$: $\mathbf{R}_p(k)\mathbf{g}_p(k) = \mathbf{e}_p(k)$ の解。

$\mathbf{R}_p(k)$: p 次の共分散行列 $\mathbf{R}_p(k) = \sum_{i=0}^{L-1} \mathbf{x}_p(k-i)\mathbf{x}_p(k-i)^T$

$\mathbf{r}_{p+N-1}(k)$: 1 次から $p+N-1$ 次の $x(k)$ の自己相関

0. 初期化

$$\mathbf{r}_{p+N-1}(N-1)_i = \mathbf{x}_L(N-1)^T \mathbf{x}_L(N-1-i), \quad (i = 1, \ldots, p+N-2)$$

$$\mathbf{s}_p(N-1) = \mathbf{e}_p(N-1) = \mathbf{g}_p(N-1) = 0$$

$k=N$ から始める

1. 高速 FIR フィルタリング算法を用いて次式を計算する

$$\hat{y}'(k+i) = \mathbf{x}_L(k+i)^T \mathbf{z}_L(k) \quad (i = 0, \ldots, N-1)$$

$i = 0, \ldots, N-1$ として、2 から 7 を繰り返す

2. $$\mathbf{r}_{p+N-2}(k+i) = \mathbf{r}_{p+N-2}(k+i-1) + x(k+i)\mathbf{x}_{p+N-2}(k+i-1) - x(k+i-L)\mathbf{x}_{p+N-2}(k+i-L-1)$$

3. $$\hat{y}'' = \mathbf{r}_{p+i-1}(k+i)^T \mathbf{s}_{p+i-1}(k+i-1)$$

4. $$e(k+i) = y(k+i) - \hat{y}'(k+i) - \hat{y}''$$

5. $$\begin{bmatrix} \mathbf{e}_p(k+i) \\ * \end{bmatrix} = \begin{bmatrix} e(k+i) \\ (1-\mu)\mathbf{e}_p(k+i-1) \end{bmatrix} \quad *\text{: don't care}$$

6. 文献[1][2]または、逆行列の定理等により、$\mathbf{g}_p(k+i)$ を計算。

7. $$\mathbf{s}_{p+i}(k+i) = \begin{bmatrix} 0 \\ \mathbf{s}_{p+i-1}(k+i-1) \end{bmatrix} + \begin{bmatrix} \mu \mathbf{g}_p(k+i) \\ 0_i \end{bmatrix}$$

8. 高速 FIR フィルタリング算法を用いて次式を計算する

$$\mathbf{z}_L(k+N) = \mathbf{z}_L(k) + [\mathbf{x}_L(k+N-p), \ldots, \mathbf{x}_L(k-p+1)]\, \mathbf{s}_{N+p-1}(k+N-1)_{p-N+p-1}$$

(s の p から $N+p-1$ の要素)