計測自動制御学会東北支部 第178回研究集会 (1998.11.20)
資料番号 178-3

分散演算型ＬＭＳ適応フィルタの収束特性解析

Convergence Property Analysis of LMS Adaptive Filters Using Distributed Arithmetic

○高橋 強*，豊田真嗣**，恒川佳隆**，三浦守**

○Kyo Takahashi*, Shinji Toyoda**, Yoshitaka Tsunekawa**, Mamoru Miura**

*岩手県立産業技術短期大学校，**岩手大学

*Iwate Industrial Technology Junior College, **Iwate University

キーワード： 分散演算 (distributed arithmetic), LMSアルゴリズム (LMS algorithm), 適応関数空間 (adaptive function space), 収束速度 (convergence speed), オフセット (offset)

連絡先： 〒028-3615 岩手県紫波郡矢巾町大字南矢幅10-3-1 岩手県立産業技術短期大学校電子技術科
高橋 強，Tel.:(019)-697-9082,Fax.:(019)-697-9089, E-mail:kyo@iwate-it.ac.jp

1. はじめに

現在，適応フィルタはエコーキャンセラ，ノイズキャンセラ，自動等化器など応用範囲は広く，様々な分野で実現の必要性が高まっている．適応フィルタを実現する際，高速性，低消費電力，良好な収束特性，小さな滞在時間 (Latency) など様々な性能が要求されるが，これらを同時に満足することは非常に困難である．

これまで，低消費電力，小規模ハードウエアを実現するために乗算器を用いない，いわゆるマルチプライヤレスな構成法として分散演算 (Distributed Arithmetic) を用いた適応フィルタが提案されてきた [2) 3) 4) 6)]．しかし，計算機シミュレーションによる我々の検討により，従来法は入力信号の符号化に特殊な符号化を用いていることにより，収束速度が大幅に劣化することがわかった．そこで，我々は入力信号の符号化に一般的な２の補数形式を用いて分散演算アルゴリズムを展開し，収束速度を大幅に改善することを可能にした．さらに，このアルゴリズムに基づいた分散演算型LMS適応フィルタのVLSI構成法を提案してきた [5)]．我々の構成法では，次数に対して高速性と滞在時間をほぼ一定に保った上で，低消費電力，小規模ハードウエアを実現することができる．

本報告では，従来法の収束速度が大きく劣化する原因を解析的に明らかにする．従来法では，入力信号の符号化に特殊な符号化を用いてアルゴリズムを導出しており，符号化後の入力信号ベクトルには定常的なオフセットが加わることになる．これにより，入力信号ベクトルの自己相関行列の最大固有値は，入力信号の分散に依存せずほぼ一定値となる．このことは，ステップサイズパラメータに最適値を設定できないことを意味し，その結

果，収束速度が大きく劣化する．

これに対し，我々の提案法では入力信号の符号化に2の補数形式を用いているため，符号化後の信号にオフセットが加わることはない．このため，自己相関行列の最大固有値は入力信号の分散に対応した値となり，最適なステップサイズパラメータを選択することが可能である．これにより，提案法では良好な収束速度を実現することができる．

2. 分散演算型適応フィルタ

従来，分散演算は定係数の内積演算を効率的に行うための計算手法として用いられてきたが，係数が時変となる適応信号処理においても有効な演算手法となる．

本章では，まず2の補数形式に基づいた定係数の分散演算の原理について述べ，次にLMSアルゴリズムの更新式から我々の提案する2の補数形式に基づく分散演算を用いた適応フィルタ(以下，DA適応フィルタと呼ぶ)と従来法の更新式を示す．

2.1 提案法の分散演算

分散演算は，内積演算をテーブルルックアップによって実現する計算手法である．

項数Nの定係数ベクトル$\boldsymbol{a}=(a_1,\ldots,a_N)$と変数ベクトル$\boldsymbol{v}=(v_1,\ldots,v_N)$との内積

$$y=\boldsymbol{a}\boldsymbol{v}=\sum_{i=1}^{N}a_iv_i \tag{1}$$

を考える．ただし，$-1\leq v_i<1$で，v_iはBビットの固定小数点形の2の補数表示である．つまり，

$$v_i=-v_i^0+\sum_{k=1}^{B-1}v_i^k2^{-k} \tag{2}$$

と表される．ここで，v_i^kはv_iのkビット目の値で0または1である．(2)式を(1)式に代入すれば，内積演算$\boldsymbol{a}\boldsymbol{v}$は次式で示される．

$$y=-\Phi(v_1^0,\ldots,v_N^0)+\sum_{k=1}^{B-1}\Phi(v_1^k,\ldots,v_N^k)2^{-k} \tag{3}$$

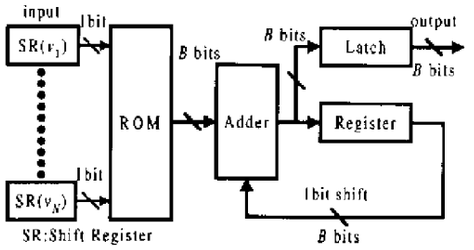

Fig. 1 Basic structure of distributed arithmetic

ただし，部分積を表す関数Φは

$$\Phi(v_1^k,\ldots,v_N^k)=\sum_{i=1}^{N}a_iv_i^k \tag{4}$$

である．(3)式より，分散演算を用いて内積を求めるためには，あらかじめ関数Φのテーブルを用意しておく必要がある．関数Φのテーブルには$(v_1^k,\ldots,v_N^k)$をアドレスとして，その各ビットパターンに対応した演算の結果を格納しておく．計算時は，アドレスのビットパターンによって関数Φのテーブルから演算結果を読み出し，右シフト加算を行う．この操作を語長B回行うことによって内積を求めることができる．Fig. 1に分散演算のアーキテクチャを示す．関数Φのテーブルは$(v_1^k,\ldots,v_N^k)$をアドレスとするROMで実現でき，右シフト加算は加算器とレジスタによって実現できる．

以上より，原理的には処理時間が項数Nに依存せず，語長Bのみに依存する構成が実現可能となる．また滞在時間も項数に依存せず，一定の値を保つことができる．ここでは滞在時間をあるサンプルがシステムに入力されてから，そのサンプルに対応した演算結果が出力されるまでの時間と定義する．さらに，乗算器を使用せずに内積演算の結果を求めることができるため，大幅に消費電力およびハードウェア量を削減することが可能となる．このようにハードウェアの効率性を考慮した場合，分散演算は非常に有効な手法となる．

2.2 DA適応フィルタの更新式の導出

LMSアルゴリズムの更新式に対して，2の補数形式を用いて式展開を行いDA適応フィルタの更新式を導出する．

タップ数Nの入力信号ベクトルを

$$\boldsymbol{S}(k) = [s(k), s(k-1), \ldots, s(k-N+1)]^T$$

タップ数Nの係数ベクトルを

$$\boldsymbol{W}(k) = [w_0(k), w_1(k), \ldots, w_{N-1}(k)]^T$$

とすると，フィルタ出力を求める式は次式で表される．

$$y(k) = \boldsymbol{S}^T(k)\,\boldsymbol{W}(k) = \boldsymbol{F}^T\,\boldsymbol{A}^T(k)\,\boldsymbol{W}(k) \quad (5)$$

$$\boldsymbol{S}(k) = \boldsymbol{A}(k)\,\boldsymbol{F} \quad (6)$$

上式において，アドレスマトリクス$\boldsymbol{A}(k)$は

$$\boldsymbol{A}(k) = \begin{bmatrix} b_0(k) & b_0(k-1) & \cdots & b_0(k-N+1) \\ b_1(k) & b_1(k-1) & \cdots & b_1(k-N+1) \\ \vdots & \vdots & \ddots & \vdots \\ b_{B-1}(k) & b_{B-1}(k-1) & \cdots & b_{B-1}(k-N+1) \end{bmatrix}^T$$

スケーリングベクトル$\boldsymbol{F}$は

$$\boldsymbol{F} = [-2^0, 2^{-1}, \ldots, 2^{-(B-1)}]^T$$

で表される．また，

$$\boldsymbol{A}_{vi}(k) = [b_i(k), b_i(k-1), \cdots, b_i(k-N+1)]^T \qquad i = 0, 1, \cdots, B-1$$

をアドレスベクトルと呼ぶことにする．LMSによる更新式

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + 2\mu e(k)\,\boldsymbol{S}(k) \quad (7)$$

の両辺に左から$\boldsymbol{A}^T(k)$を掛け，

$$\boldsymbol{A}^T(k)\,\boldsymbol{W}(k+1) = \boldsymbol{A}^T(k)\{\boldsymbol{W}(k) + 2\mu e(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}\} \quad (8)$$

となる．ここで，誤差信号$e(k)$は

$$e(k) = d(k) - y(k) \quad (9)$$

であり，$d(k)$は未知システムの所望信号である．次に，$\boldsymbol{A}^T(k)\,\boldsymbol{W}(k)$と$\boldsymbol{A}^T(k)\,\boldsymbol{W}(k+1)$を，

$$\boldsymbol{P}(k) = \boldsymbol{A}^T(k)\,\boldsymbol{W}(k) \quad (10)$$

$$\boldsymbol{P}(k+1) = \boldsymbol{A}^T(k)\,\boldsymbol{W}(k+1) \quad (11)$$

と定義することにより，(8)式は

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + 2\mu e(k)\,\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F} \quad (12)$$

となる．ここで，関数$\boldsymbol{P}(k)$と$\boldsymbol{P}(k+1)$は

$$\begin{aligned}\boldsymbol{P}(k) &= [p_0(k), p_1(k), \ldots, p_{B-1}(k)]^T \\ &= [\boldsymbol{A}^T_{v\,0}(k)\,\boldsymbol{W}(k),\ \boldsymbol{A}^T_{v\,1}(k)\,\boldsymbol{W}(k), \\ &\qquad \cdots,\ \boldsymbol{A}^T_{v\,B-1}(k)\,\boldsymbol{W}(k)]^T\end{aligned}$$

$$\begin{aligned}\boldsymbol{P}(k+1) &= [p_0(k+1), p_1(k+1), \ldots, p_{B-1}(k+1)]^T \\ &= [\boldsymbol{A}^T_{v\,0}(k)\,\boldsymbol{W}(k+1),\ \boldsymbol{A}^T_{v\,1}(k)\,\boldsymbol{W}(k+1), \\ &\qquad \cdots,\ \boldsymbol{A}^T_{v\,B-1}(k)\,\boldsymbol{W}(k+1)]^T\end{aligned}$$

である．(10)式を用いて，(5)式のフィルタ出力は，

$$y(k) = \boldsymbol{F}^T\,\boldsymbol{P}(k) \quad (13)$$

と表される．

以上より，LMS適応フィルタでは(7)式のように係数$\boldsymbol{W}(k)$を直接更新するのに対し，DA適応フィルタでは(12)式のように，アドレスマトリクス$\boldsymbol{A}(k)$の列ベクトルである$\boldsymbol{A}_{vi}(k)$をアドレスとする空間(以下，適応関数空間と呼ぶ)の値を更新する．

さらに，入力信号の白色性を仮定して，(12)式における$\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}$の平均値は，

$$E[\,\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}] = 0.25N\,\boldsymbol{F} \quad (14)$$

となる．更新式は，(12)式中の$\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}$を(14)式で置き換えて

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + 0.5\mu N e(k)\,\boldsymbol{F} \quad (15)$$

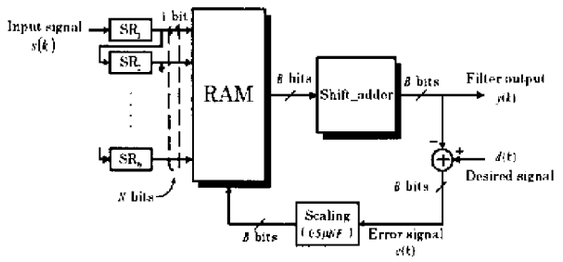

Fig. 2　Block diagram of DA adaptive filter

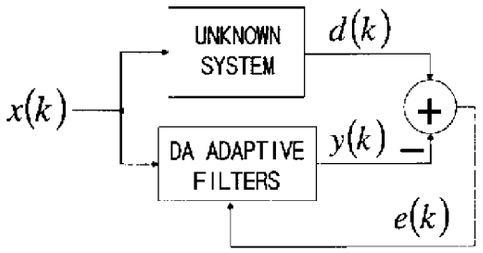

Fig. 3　Computer simulation model

と簡略化される．この更新式を用いた場合にも，多くの計算器シミュレーションにより収束することが確認されており，また実際にディジタル電話回線の適応エコーキャンセラ等に用いられている[1]．(15)式の更新値は，$0.5\mu N$を2のべき乗で近似することにより，誤差$e(k)$に対するシフト操作のみで求めることができる．したがって，乗算器を使用しない，いわゆるマルチプライヤレスのハードウェア実現が可能となる．DA適応フィルタのブロック図をFig. 2に示す．適応関数空間の実現は，固定係数ではROMを使用したが，DA適応フィルタではRAMを用いている．

2.3　従来法の更新式の導出

従来法における入力信号$s(k)$の符号化は，まずオフセットバイナリ形式を用いて符号化し，次にスケーリングベクトルの変更と，ビットの値を"0","1"に対して"-1","1"を対応させて入力信号を表現している．このとき，入力信号ベクトルは以下の式で表される．

$$\boldsymbol{S}(k) = \boldsymbol{A}'(k)\,\boldsymbol{F}' \tag{16}$$

上式において，アドレスマトリクス$\boldsymbol{A}'(k)$は

$$\boldsymbol{A}'(k) = \begin{bmatrix} b'_0(k) & b'_0(k-1) & \cdots & b'_0(k-N+1) \\ b'_1(k) & b'_1(k-1) & \cdots & b'_1(k-N+1) \\ \vdots & \vdots & \ddots & \vdots \\ b'_{B-1}(k) & b'_{B-1}(k-1) & \cdots & b'_{B-1}(k-N+1) \end{bmatrix}^T$$

スケーリングベクトル$\boldsymbol{F}'$は

$$\boldsymbol{F}' = [2^{-1}, 2^{-2}, \ldots, 2^{-B}]^T \tag{17}$$

である．また，アドレスマトリクスの各列

$$\boldsymbol{A}'_{vi}(k) = [b_i(k), b_i(k-1), \cdots, b_i(k-N+1)]^T \qquad i = 0, 1, \cdots, B-1$$

をアドレスベクトルと呼ぶことにする．

更新式は，提案法と同様にLMSの更新式の両辺に左から$\boldsymbol{A'}^T(k)$を掛けることにより，

$$\boldsymbol{P}'(k+1) = \boldsymbol{P}'(k) + 2\mu e'(k)\,\boldsymbol{A'}^T(k)\,\boldsymbol{A}'(k)\,\boldsymbol{F}' \tag{18}$$

となる．ここで，入力信号に白色性を仮定して，(18)式の$\boldsymbol{A'}^T(k)\,\boldsymbol{A}'(k)$の平均値は，

$$E[\boldsymbol{A'}^T(k)\,\boldsymbol{A}'(k)] = N\boldsymbol{I} \tag{19}$$

となる．ここで，$\boldsymbol{I}$は$B \times B$の単位行列である．更新式は，(18)式中の$\boldsymbol{A'}^T(k)\,\boldsymbol{A}'(k)$を(19)式で置き換えて

$$\boldsymbol{P}'(k+1) = \boldsymbol{P}'(k) + 2\mu e'(k) N\,\boldsymbol{F}' \tag{20}$$

と簡略化される．

3.　DA適応フィルタの収束特性

前章で導出したDA適応フィルタの更新式を用いて計算機シミュレーションを行い，その結果に基づいて収束速度を検討する．

シミュレーションモデルをFig. 3に示す．未知系は，タップ数16の低域通過FIRフィルタである．

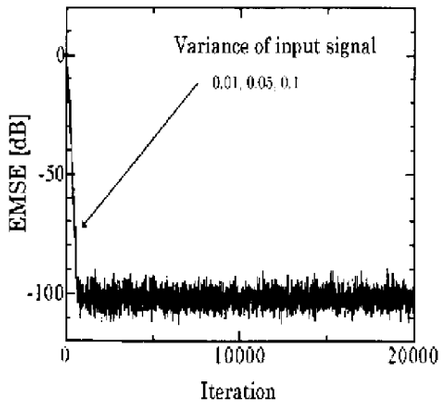

Fig. 4 Computer simulation results of our proposed method

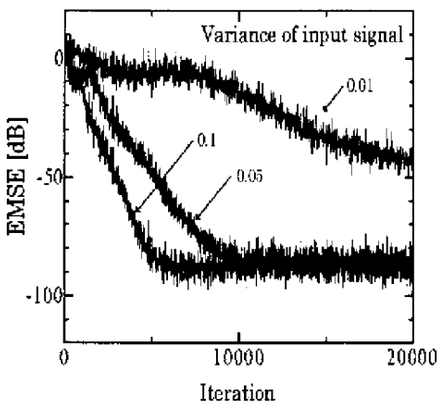

Fig. 5 Computer simulation results of conventional method

入力信号は平均零の白色ガウス雑音で，分散の値は0.1, 0.05, 0.01とした．提案法のシミュレーション結果をFig. 4，従来法をFig. 5に示す．ここで，ステップサイズパラメータμは最も高速な収束速度を示す値を最適値として選択した．計算機シミュレーションで用いたステップサイズパラメータの値を，Table 1に示す．

これらより，従来法は最適なステップサイズパラメータを選択したにも関わらず，入力信号の分散が小さくなるにしたがい収束速度が劣化している．また，最適なステップサイズは入力信号の分散には関係せず，一定の値を示していることがわかる．これに対して提案法のステップサイズは，分散に応じて最適値が変化しており，いずれの分散に対しても良好な収束速度を示している．この計算機シミュレーションは数多くの例に対しても行っており，同様の結果が得られることを確認している．

Table 1 Optimum step size parameter used in the computer simulation

Variance of input signal	0.1	0.05	0.01
Our proposed method	0.25	0.5	4.0
Conventional method	0.125	0.125	0.125

4. 収束速度の劣化原因

LMSアルゴリズムを用いた適応フィルタにおいて，係数が収束するステップサイズパラメータの範囲は，

$$0 < \mu < 1/\lambda_{max} \tag{21}$$

であることが知られており，μが大きいほど高速な収束速度を示す[8]．ここで，λ_{max}は入力信号の自己相関行列の最大固有値である．したがって，LMSアルゴリズムでは入力信号の自己相関行列の最大固有値とステップサイズパラメータμの設定値が収束速度を決定する大きな要因となる．

本章では，この性質に着目して，従来法によるDA適応フィルタの収束速度の劣化原因を解析的に明らかにする．

4.1 全適応関数空間を更新する入力信号ベクトルの定義

LMS適応フィルタでは，(7)式のように誤差$e(k)$と入力信号ベクトル$\boldsymbol{S}(k)$の積を更新値として，係数行列全体を直接更新している．これに対し，DA適応フィルタでは(15)式，(20)式のようにアドレスベクトル$\boldsymbol{A}v_i(k)$に対応する適応関数空間のみを更新する．したがって，これらの更新式は適応関

数空間全体を記述したものではない．そこで，これまで導出したDA適応フィルタの更新式を，適応関数空間全体に対する更新式に拡張することにより，全適応関数空間に対する入力信号を新たに定義し，入力信号の性質を明らかにする．

まず，タップ数が1の場合について適応関数空間全体を更新する入力信号を定義する．2の補数形式を用いる提案法において，一例として入力信号$s(k)$が

$$s(k)=0\times(-2^0)+1\times2^{-1}+0\times2^{-2}+1\times2^{-3} \quad (22)$$

の場合，ビット0，1に対応する適応関数空間要素$p0(k)$，$p1(k)$は提案法の更新式を用いて次のように更新される．

$$\begin{aligned} p0(k+1) &= p0(k)+0.5\mu Ne(k)[-2^0+2^{-2}] \\ &= p0(k)+0.5\mu Ne(k)\bar{s}(k) \quad (23) \\ p1(k+1) &= p1(k)+0.5\mu Ne(k)[2^{-1}+2^{-3}] \\ &= p1(k)+0.5\mu Ne(k)s(k) \quad (24) \end{aligned}$$

ここで，$\bar{s}(k)$は$s(k)$のビットパターンを反転した信号を表す．このように，適応関数空間要素$p0(k)$，$p1(k)$はそれぞれ入力信号$\bar{s}(k)$，$s(k)$により更新されることがわかる．任意の入力信号に対して，更新式は(23)と(24)式をまとめて，

$$\begin{aligned} \boldsymbol{P}_w(k+1) &= \boldsymbol{P}_w(k)+0.5\mu Ne(k)\,\boldsymbol{S}_{DA}(k) \quad (25) \\ \boldsymbol{S}_{DA}(k) &= [\bar{s}(k),s(k)]^T \quad (26) \\ \boldsymbol{P}_w(k) &= [p0(k),p1(k)]^T \quad (27) \end{aligned}$$

となる．このように，適応関数空間全体は新たな入力信号ベクトル$\boldsymbol{S}_{DA}(k)$によって更新されている．

従来法の場合，(22)式の入力信号はオフセットバイナリのビットパターン（MSBが反転）となり

$$s(k)=1\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4} \quad (28)$$

となる．この場合，ビット0，1に対応する適応関数空間要素$p'0(k)$，$p'1(k)$は，従来法の更新式を用いて次のように更新される．

Table 2　Relation between symbol and bit pattern

Symbol name	Bit pattern
a	00
b	01
c	10
d	11

$$\begin{aligned} p'0(k+1) &= p'0(k)+2\mu Ne(k)[2^{-3}] \\ &= p'0(k)+2\mu Ne(k)\bar{s}(k) \quad (29) \\ p'1(k+1) &= p'1(k)+2\mu Ne(k)[2^{-1}+2^{-2}+2^{-4}] \\ &= p'1(k)+2\mu Ne(k)s(k) \quad (30) \end{aligned}$$

(29)と(30)式をまとめると，

$$\begin{aligned} \boldsymbol{P}'_w(k+1) &= \boldsymbol{P}'_w(k)+2\mu Ne(k)\,\boldsymbol{S}'_{DA}(k) \quad (31) \\ \boldsymbol{S}'_{DA}(k) &= [\bar{s}(k),s(k)]^T \quad (32) \\ \boldsymbol{P}'_w(k) &= [p'0(k),p'1(k)]^T \quad (33) \end{aligned}$$

となる．ここで，$\bar{s}(k)$は$s(k)$のビットパターンを反転した信号を表す．

次にタップ数が2の場合について検討する．入力信号ベクトルを

$$\boldsymbol{S}(k)=[s(k),s(k-1)]^T \quad (34)$$

とする．ここで，信号$s(k)$，$s(k-1)$を構成するビットは"0"，"1"の2値を持つため，適応関数空間要素を指定するアドレスベクトル$\boldsymbol{A}v_i(k)$の種類は，"0"と"1"の組み合わせ数の2^N種類存在する．タップ数2では4種類のアドレスベクトルが存在するため，表現の簡単のために，それらをTable 2のようにアルファベットで表すことにする．

一例として，入力信号の組み合わせが

$$s(k)=-0\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3} \quad (35)$$

$$s(k-1)=-0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3} \quad (36)$$

Table 3　Access pattern of adaptive function space

Symbol	Bit pattern	-2^0	2^{-1}	2^{-2}	2^{-3}
a	00	1	0	0	0
b	01	0	1	0	0
c	10	0	0	1	0
d	11	0	0	0	1

の場合，入力信号ベクトルをシンボルを用いて表すと，

$$\boldsymbol{S}_s = -a \times 2^0 + b \times 2^{-1} + c \times 2^{-2} + d \times 2^{-3} \quad (37)$$

となる．なお，語長を4[bit]とし，入力信号は4種類の全パターンが現れるビットパターンを選択した．

この場合，適応関数空間は提案法の更新式を用いて次のように更新される．

$$pa(k+1) = pa(k) - 0.5\mu \times 2 \times e(k) 2^0 \quad (38)$$

$$pb(k+1) = pb(k) + 0.5\mu \times 2 \times e(k) 2^{-1} \quad (39)$$

$$pc(k+1) = pc(k) + 0.5\mu \times 2 \times e(k) 2^{-2} \quad (40)$$

$$pd(k+1) = pd(k) + 0.5\mu \times 2 \times e(k) 2^{-3} \quad (41)$$

(38) ～ (41)式をまとめると，

$$\boldsymbol{P}_w(k+1) = \boldsymbol{P}_w(k) + 0.5\mu N e(k)\, \boldsymbol{A}_{ac}^T(k)\, \boldsymbol{F} \quad (42)$$

となる．ここで，各適応関数空間と更新に寄与するスケーリングベクトル要素の対応関係をTable 3に示す．(42)式の$\boldsymbol{A}_{ac}^T(k)$はTable 3に相当し，この場合は以下に示す4×4のマトリクスである．

$$\boldsymbol{A}_{ac}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (43)$$

この行列は，入力信号ベクトルのビットパターンにより一意に決定され，これをアクセスマトリクスと呼ぶことにする．また適応関数空間$\boldsymbol{P}_w(k)$は

$$\boldsymbol{P}_w(k) = [pa(k), pb(k), pc(k), pd(k)]^T \quad (44)$$

である．

これを，任意の入力信号ベクトルに対してタップ数Nに一般化すると，

$$\boldsymbol{P}_w(k+1) = \boldsymbol{P}_w(k) + 0.5\mu N e(k)\, \boldsymbol{S}_{DA} \quad (45)$$

$$\boldsymbol{S}_{DA}(k) = \boldsymbol{A}_{ac}^T(k)\, \boldsymbol{F} \quad (46)$$

となる．ここで，$\boldsymbol{A}_{ac}^T(k)$は$2^N \times B$のアクセスマトリクス，適応関数空間$\boldsymbol{P}_w(k)$は

$$\boldsymbol{P}_w(k) = [p_0(k), p_1(k), \cdots, p_{2^N-1}(k)]^T \quad (47)$$

スケーリングベクトルは，

$$\boldsymbol{F} = [-2^0, 2^{-1}, \cdots, 2^{-B+1}]^T \quad (48)$$

である．(45)式とLMSアルゴリズムの(7)式を比較すると，$\boldsymbol{S}_{DA}(k)$はLMSアルゴリズムにおける入力信号ベクトル$\boldsymbol{S}(k)$に相当しており，適応関数空間の更新値を決定する入力信号ベクトルである．

従来法の更新式も同様に導かれ，以下のようになる．

$$\boldsymbol{P'}_w(k+1) = \boldsymbol{P'}_w(k) + 2\mu N e(k)\, \boldsymbol{S'}_{DA} \quad (49)$$

$$\boldsymbol{S'}_{DA}(k) = \boldsymbol{A'}_{ac}^T(k)\, \boldsymbol{F'} \quad (50)$$

ここで，$\boldsymbol{A'}_{ac}^T(k)$は$2^N \times B$のアクセスマトリクス，適応関数空間$\boldsymbol{P'}_w(k)$は

$$\boldsymbol{P'}_w(k) = [p'_0(k), p'_1(k), \cdots, p'_{2^N-1}(k)]^T \quad (51)$$

スケーリングベクトルは，

$$\boldsymbol{F'} = [2^{-1}, 2^{-2}, \cdots, 2^{-B}]^T \quad (52)$$

である．

これまでDA適応フィルタにおける入力信号は，適応関数空間を指定するために用いられると解釈されていた．しかし，適応関数空間の更新状況を表すアクセスマトリクス$\boldsymbol{A}_{ac}(k)$を導入して，これまでの更新式を全適応関数空間に拡張した．これにより，(45)，(49)式のように，適応関数空間はそれぞれ入力信号$\boldsymbol{S}_{DA}(k)$，$\boldsymbol{S'}_{DA}(k)$を直接的に用いて更新される．

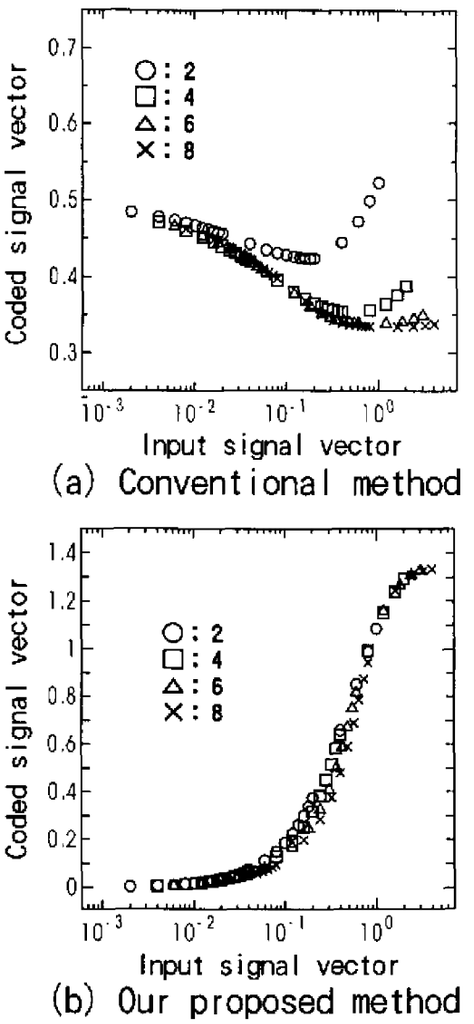

Fig. 6 Trace of autocorrelation matrix for order N= 2, 4, 6, 8

4.2 入力信号ベクトルの性質

本章では，計算機シミュレーションにより(46)式と(50)式で定義される提案法と従来法の入力信号ベクトルの性質を，本来の入力信号ベクトル$\boldsymbol{S}(k)$と比較することにより明らかにする．Fig. 6は，入力信号$s(k)$の分散を変化させて各入力信号ベクトルの自己相関行列のトレースを求め，比較した結果である．これは，入力信号$s(k)$として平均0の白色信号を，50の異なる初期値に対して100,000回発生させた試行平均で，タップ数は2，4，6，8とした．

次に，タップ数4に対して自己相関行列の固有値を実際に求めた結果をTable 4に示す．

Table 4 Maximum Eigen value of autocorrelation matrix,N=4

Variance of input signal	0.1	0.05	0.01
Our proposed method	0.061	0.030	0.006
Conventional method	0.063	0.063	0.063

以上の結果をまとめると，次のようになる．

[提案法]

・自己相関行列のトレースは，入力信号ベクトルと正の相関を持つ．

・自己相関行列の最大固有値は，入力信号の分散に応じて変化する．

・最適なステップサイズパラメータは，分散に応じて変化する．

[従来法]

・自己相関行列行列のトレースは，入力信号ベクトルと負の相関を持つ．

・入力信号の分散が小さくなるにしたがい0.5に漸近する．

・自己相関行列の最大固有値は，分散によらず一定値を示す．

・最適なステップサイズパラメータは，分散によらず一定値を示す．

従来法において，入力信号の分散が小さい場合に自己相関行列のトレースが0.5に漸近することは，符号化後の入力信号ベクトルに定常的なオフセットが含まれていると考えられる．このオフセットは入力信号の分散には関係せず一定値を持つため，自己相関行列のある固有値も一定値を有する．LMS適応フィルタにおいては，(21)式のようにステップサイズパラメータの上限が規定され，大きな値を設定するほど高速な収束を示す．したがって，最大固有値が入力信号の分散に依存しない従来法では，最適なステップサイズパラメータ

を設定できず収束速度が大きく劣化する．

5. まとめ

本報告では，従来型の分散演算型LMS適応フィルタの収束速度が劣化する原因について解析的に明らかにした．従来型のDA適応フィルタでは，入力信号の符号化に特殊な符号化を用いていた．それは，入力信号に対するビットパターンがオフセットバイナリ形式であり，ビットの値が通常の"0"，"1"に対して"-1"，"1"を有するという符号である．従来法のアルゴリズムは，この符号化形式を前提に導出されていた．ところが，実現の段階ではビットを"0"，"1"として取り扱っているため，符号ビットを持たない正値のスケーリングベクトルでは正の値しか表現できず，符号化された入力信号は定常的なオフセットを有することになる．従来法は，この入力信号のオフセットが原因となり，収束速度を決定するステップサイズパラメータが最適値に設定できないため，収束速度が大きく劣化した．

これに対して，我々の提案した方法では2の補数形式を採用しているため，このような問題は発生しない．したがって，これまで我々の提案してきた分散演算型LMS適応フィルタは収束速度において非常に有利であることも同時に示された．

参考文献

1) 牧野昭二，小泉宣夫，"エコーキャンセラの室内音場における適応特性の改善について，" 信学論(A)，vol.J71-A，no.12，pp.2212–2214，Dec. 1988.

2) A. Peled and B. Liu, "A new hardware realization of digital filters," IEEE Trans. Acoust., Speech & Signal Process., vol.22, no.12, pp.456–462, Dec. 1974.

3) C.F.N.Cowan and J.Mavor, "New Digital Adaptive Filter Implementation using Distributed Arithmetic Techniques," IEE Proc., Pt.F, vol.128, no.4, pp.225–233, Aug. 1981.

4) C.H. Wei, J.J. Lou, "Multimemory block structure for implementing a digital adaptive filter using distributed arithmetic," IEE Proc., vol.133, Pt.G, no.1, pp.19–26, Feb. 1986.

5) 豊田真嗣，高橋　強，恒川佳隆，三浦　守，"分散演算型LMS適応フィルタのVLSI実現，" 第12回ディジタル信号処理シンポジウム講演論文集，B8-3，pp.645–650，Nov. 1997.

6) C.F.N. Cowan, S.G. Smith and J.H. Elliott, "A digital adaptive filter using a memory-accumulator architecture:theory and realization," IEEE Trans. Acoust., Speech & Signal Process., vol.31, no.3, pp.541–549, Jun. 1983.

7) G.Strang,"Linear Algebra and its Application, "Academic Press,Inc.,New York,1976

8) S.Haykin,"Introduction to Adaptive Filters, " Macmillan publishing Company, New York, 1984