

分散演算を用いたLMS適応フィルタの収束条件

Convergence Condition Analysis of LMS Adaptive Filters Using Distributed Arithmetic

○高橋 強*, 恒川佳隆**, 田山典男**, 関 享士郎**

○Kyo Takahashi*, Yoshitaka TSUNEKAWA**, Norio TAYAMA**,
Kyoushirou SEKI**

*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), 収束条件 (convergence condition), 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には高速性, 低消費電力, 良好な収束特性, 小さな出力滞在時間 (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りより効率的な実現法が望まれている. また, TV会議などで必要な音響エコーキャンセラにおいては, 室内音場のインパルス応答を高速に推定する能力とインパルス応答変動に対する追従性が要求される⁸⁾. そのため, 非常に高次の適応フィルタが必要とされている.

これまでCowanらは, 低消費電力, 小規模ハードウェ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を提案した^{1) 2) 3) 4)}. 従来, この分散演算は定係数の内積演算を効率的に計算する方法として知られていたが, 入力信号の符号化にオフセットバイナリ形式を用いると部分積を格納する関数空間が奇対称性を有し, この性質を利用して関数空間

を構成するためのメモリ容量を1/2に削減することが可能である. ところが2の補数形式ではこの性質は現れないため, 関数空間の奇対称性はオフセットバイナリ形式に特有の性質であるとされてきた^{10) 11)}. Cowanらはこの性質を利用するためにオフセットバイナリ形式を用いた分散演算を適応フィルタに適用したが, 収束速度が極端に劣化することが我々の計算機シミュレーションによって明らかになり, その原因は入力信号の符号化方式にあることを示した.^{5) 9)}

そこで, 我々は一般的な符号化方式である2の補数形式を用いて分散演算型LMS適応フィルタのアルゴリズムを統一的に展開することにより, 収束速度を大幅に改善することを可能にした^{5) 6) 9)}. また, 定係数の分散演算では現れなかった適応関数空間の奇対称性を初めて見出し, この性質を利用したハーフメモリアルゴリズムを提案し, さらに効率的なVLSIアーキテクチャを提案してきた. 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, 小規模ハードウェアを実現することが可能である. しかし, 分散演算型LMS適応フィルタの収束速度に関する解析はこれまで行われてこなかった.

本報告では、分散演算型LMS適応フィルタ(DA適応フィルタ)の収束速度について解析的に明らかにする⁷⁾。まず、分散演算型LMS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。拡張された更新式はもとの更新式と等価であり、これをLMSアルゴリズムと比較することによって適応関数空間全体に対する入力信号ベクトルを新たに定義する。次に、拡張した更新式に対する分散演算型LMS適応フィルタの収束式と収束条件を導出する。これにより、収束速度は新たに定義した入力信号ベクトルの自己相関行列の固有値分布に依存することを明らかにする。次に、新たに定義した入力信号ベクトルの性質を検討することにより、従来法の収束速度が大幅に劣化する原因について述べる。さらに、有色性のある入力信号に対する収束特性についても検討した。その結果、分散演算型LMS適応フィルタは有色性の影響を低減する一種の白色化効果を有することが明らかになった。

2. 分散演算型LMS適応フィルタ

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。

2.1 従来法の更新式の導出

従来法においては、入力信号 $s(k)$ の符号化にはオフセットバイナリ形式を用いるとされてきた^{1) 2) 3) 4)}。このとき、 N 次入力信号ベクトル $S'(k)$ は以下の式で表される。

$$\begin{aligned} S'(k) &= [s'(k), s'(k-1), \dots, s'(k-N+1)]^T (1) \\ &= A'(k) F' \end{aligned} \quad (2)$$

上式において、アドレスマトリクス $A'(k)$ は

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

また、アドレスマトリクスの各列

$$\begin{aligned} A'_{vi}(k) &= [b'_i(k), b'_i(k-1), \dots, b'_i(k-N+1)]^T \\ i &= 0, 1, \dots, B-1 \end{aligned}$$

をアドレスベクトルと呼び、スケーリングベクトル F' は

$$F' = [2^{-1}, 2^{-2}, \dots, 2^{-B}]^T \quad (3)$$

である。ここで、アドレスマトリクスとアドレスベクトルの要素は入力信号を構成するオフセットバイナリ形式におけるビットを表し、 $-1, 1$ の値を持つ。

タップ数 N の係数ベクトルを

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

とすると、フィルタ出力を求める式は次式で表される。

$$y'(k) = S'^T(k) W(k) = F'^T A'^T(k) W(k) \quad (4)$$

更新式を導出する。LMSアルゴリズムは以下の式で表される。

$$W(k+1) = W(k) + 2\mu e'(k) S'(k) \quad (5)$$

ここで、 $e'(k)$ は所望信号 $d(k)$ とフィルタ出力 $y'(k)$ の差を表す誤差信号

$$e'(k) = d(k) - y'(k) \quad (6)$$

である。(5)式の両辺に左から $A'^T(k)$ を掛け、

$$\begin{aligned} A'^T(k) W(k+1) &= \\ A'^T(k) \{ W(k) + 2\mu e'(k) A'(k) F' \} \end{aligned} \quad (7)$$

次に、 $A'^T(k) W(k)$ と $A'^T(k) W(k+1)$ を、

$$P'(k) = A'^T(k) W(k) \quad (8)$$

$$= [p'_0(k), \dots, p'_{B-1}(k)]^T \quad (9)$$

$$P'(k+1) = A'^T(k) W(k+1) \quad (10)$$

$$= [p'_0(k+1), \dots, p'_{B-1}(k+1)]^T \quad (11)$$

と定義することにより、(7)式は

$$P'(k+1) = P'(k) + 2\mu e'(k) A'^T(k) A'(k) F' \quad (12)$$

となる。ここで、入力信号が平均0の白色雑音であると仮定すると(12)式の $A'^T(k) A'(k)$ の平均値は、

$$E[A'^T(k) A'(k)] = NI \quad (13)$$

となる。なお、 I は $B \times B$ の単位行列である。更新式は、(12)式の $A'^T(k) A'(k)$ を(13)式で置き換えて

$$P'(k+1) = P'(k) + 2\mu e'(k) N F' \quad (14)$$

と簡略化される。ここで、 $P'(k)$ はアドレスベクトル $A'_{vi}(k)$ を引数として部分積を返す関数で適応

関数空間と呼ばれる。LMS適応フィルタでは、(5)式のように係数ベクトル $W(k)$ の要素 $w_i(k)$ を更新するのに対し、DA適応フィルタでは(14)式のように適応関数空間 $P'(k)$ の要素 $p'_i(k)$ を更新する。この適応関数空間はRAM(Random access memory)を用いて実現されるが、空間要素の指定はアドレスベクトル $A'_{vi}(k)$ をRAMのアドレス信号として用いて行うため、アルゴリズムの導出過程で-1, 1としていたビット値は0, 1として用いられることになる。また、(9)式を用いると(4)式のフィルタ出力は、

$$y'(k) = F^T P'(k) \quad (15)$$

$$= 2^{-1} \times p'_0(k) + \dots + 2^{-B} \times p'_{B-1}(k) \quad (16)$$

と表される。

従来法では、アルゴリズム導出が容易であることと適応関数空間の奇対称性を利用するためにオフセットバイナリ形式を用いた。しかし、我々の検討によりこの符号形式が原因となり収束速度が極端に劣化することが明らかになった^{5) 9)}。

2.2 提案法の更新式の導出

従来法の問題点を解決するために、我々は符号化方式に2の補数形式を用いて分散演算型LMSアルゴリズムを導出した^{5) 6) 9)}。

N 次入力信号ベクトルを

$$S(k) = [s(k), s(k-1), \dots, s(k-N+1)]^T$$

タップ数 N の係数ベクトルを

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

とすると、フィルタ出力は次式で表される。

$$y(k) = S^T(k) W(k) = F^T A^T(k) W(k) \quad (17)$$

$$S(k) = A(k) F \quad (18)$$

上式において、アドレスマトリクス $A(k)$ とスケーリングベクトル F は

$A(k) =$

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

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

であり、アドレスベクトル A_{vi} は

$$A_{vi}(k) = [b_i(k), b_i(k-1), \dots, b_i(k-N+1)]^T \\ i = 0, 1, \dots, B-1$$

である。提案法の更新式は、LMSの更新式の両辺に左から $A^T(k)$ を掛け、

$$A^T(k) W(k+1) = \\ A^T(k) \{ W(k) + 2\mu e(k) A(k) F \} \quad (19)$$

となる。ここで、誤差信号 $e(k)$ は

$$e(k) = d(k) - y(k) \quad (20)$$

である。ここで、適応関数空間を次のように定義する。

$$P(k) = A^T(k) W(k) \quad (21)$$

$$= [p_0(k), \dots, p_{B-1}(k)]^T \quad (22)$$

$$P(k+1) = A^T(k) W(k+1) \quad (23)$$

$$= [p_0(k+1), \dots, p_{B-1}(k+1)]^T \quad (24)$$

これより、(19)式は

$$P(k+1) = P(k) + 2\mu e(k) A^T(k) A(k) F \quad (25)$$

となる。(22)式を用いて、(17)式のフィルタ出力は、

$$y(k) = F^T P(k) \quad (26)$$

と表される。そして、これまで2の補数形式では不可能とされてきた $A^T(k) A(k)$ の対角化をスケーリングベクトル F も含めて考えることにより初めて可能にした⁹⁾。入力信号が平均0の白色雑音であると仮定すると、(25)式における $A^T(k) A(k) F$ の平均値は、

$$E[A^T(k) A(k) F] = 0.25N F \quad (27)$$

となり、更新式は(25)式中の $A^T(k) A(k) F$ を(27)式で置き換えて

$$P(k+1) = P(k) + 0.5\mu N e(k) F \quad (28)$$

と簡略化される。この更新式を用いた場合にも、多くの計算機シミュレーションにより収束することが確認されている⁹⁾。(28)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適応フィルタの構成をFig. 1に示す。

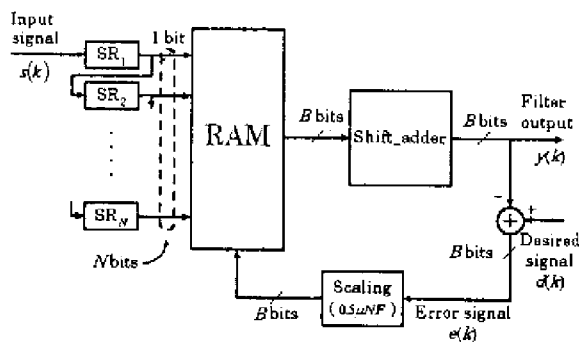


Fig. 1 Block diagram of DA adaptive filter.

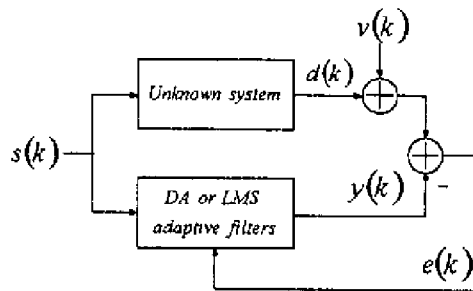


Fig. 2 Computer simulation model.

3. DA 適応フィルタの収束速度

計算機シミュレーションにより収束速度を検討する。シミュレーションモデルを Fig. 2 に示す。ここで、未知系はタップ数 16 の低域通過 FIR フィルタ、入力信号は平均零の白色ガウス雑音である。入力信号の分散 0.1, 0.05, 0.01 に対する提案法のシミュレーション結果を Fig. 3, 従来法を Fig. 4 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。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、従来法が収束するまでには、提案法に比較して分散 0.1 で約 6 倍, 0.05 で約 13 倍そして 0.01 では約 50 倍の繰り返しが必要である。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。

Table 1 Step size parameter used in the computer simulation.

Input variance	0.1	0.05	0.01
Proposed method	0.25	0.5	4.0
Conventional method	0.125	0.125	0.125

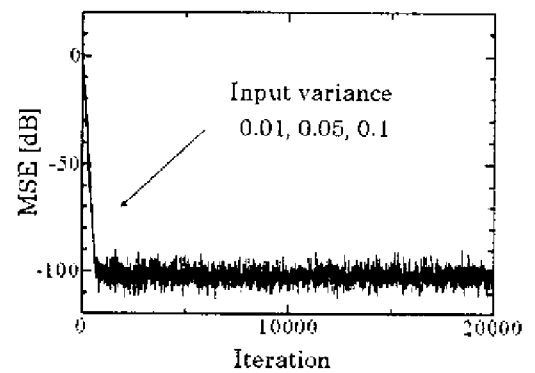


Fig. 3 Convergence properties of our proposed method for various variance of input signal.

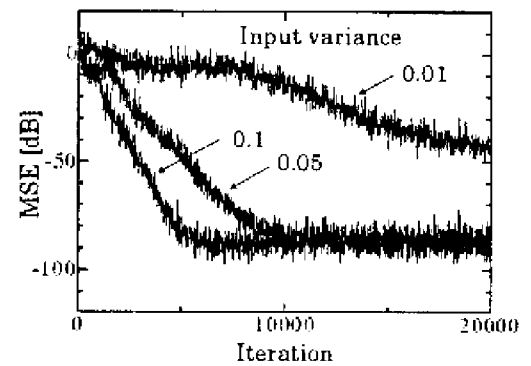


Fig. 4 Convergence properties of conventional method for various variance of input signal.

次に、タップ数 4, 8, 16 に対する提案法のシミュレーション結果を Fig. 5 に、従来法の結果を Fig. 6 に示す。なお、入力信号は分散 0.01 の白色ガウス雑音である。これらより、従来法はタップ数を増加させると収束速度が極端に劣化し、タップ数 $N=4$ で約 16 倍, $N=8$ で約 20 倍そして $N=16$ では約 56 倍もの繰り返しの必要とする。いずれの場合も、従来法に比べて提案法は良好な収束速度を示しており、数多くの例においても同様の結果が得られることを確認している。

さらに、有色信号に対するシミュレーション結果を Fig. 7 に示すが、提案法は LMS よりも高速な収束を示していることがわかる。なお、入力信号は係数 0.99 の 1 次 AR 過程で分散は 0.01 である。

4. 収束条件の導出

分散演算型 LMS 適応フィルタは未知システムの伝達関数を適応関数空間として推定するため、収束条件式は全適応関数空間を対象にする必要がある。しかし、更新式である (14) 式と (28) 式は、ある時刻 k において更新対象となる要素のみを記述

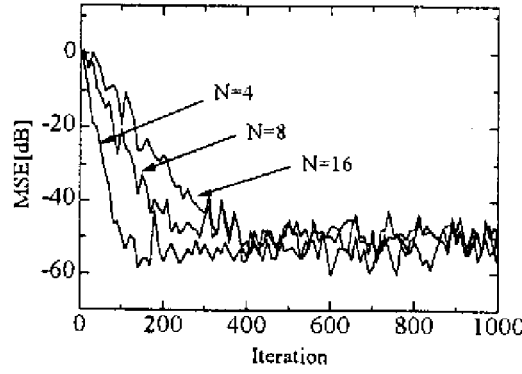


Fig. 5 Convergence properties of proposed method for various tap number N .

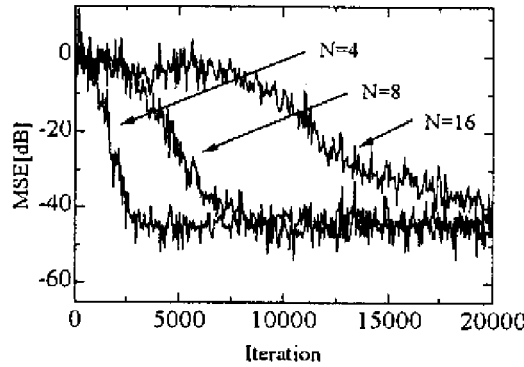


Fig. 6 Convergence properties of conventional method for various tap number N .

しているため、これまでの更新式では全適応関数空間に対する収束条件を導くことはできない。そこで、まず分散演算型LMS適応フィルタの更新式を全適応関数空間に拡張し入力信号ベクトルを新たに定義する。ついで、推定誤差を適応関数空間の最適値と推定値の差として収束条件式を定式化することにより、収束条件を時刻 k の経過とともに推定誤差が減少するための条件として導く。

4.1 更新式の全空間への拡張

更新式を全適応関数空間に対して拡張し、拡張された入力信号ベクトルを新たに定義する。

まず、タップ数 N 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

$$s(k) = 0 \times (-2^0) + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \quad (29)$$

の場合、ビット0, 1に対応する適応関数空間要素 $p_0(k)$, $p_1(k)$ は提案法の更新式を用いて次のように更新される。

$$\begin{aligned} p_0(k+1) &= p_0(k) + 0.5\mu N e(k) [-2^0 + 2^{-2}] \\ &= p_0(k) + 0.5\mu N e(k) \bar{s}(k) \end{aligned} \quad (30)$$

$$p_1(k+1) = p_1(k) + 0.5\mu N e(k) [2^{-1} + 2^{-3}]$$

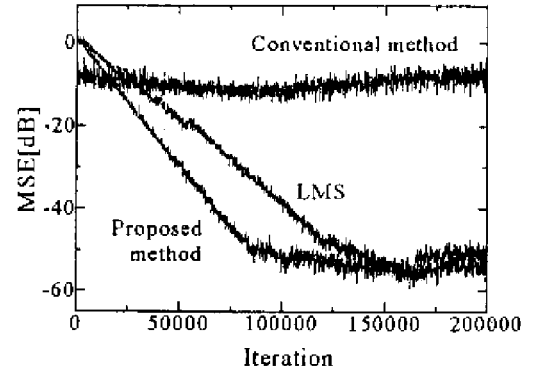


Fig. 7 Convergence properties of proposed, conventional and LMS methods for colored signal using an AR process of order 1.

Table 2 Relation between symbols and bit patterns.

Symbol	Bit pattern	Symbol	Bit pattern
a	$[00]^T$	c	$[10]^T$
b	$[01]^T$	d	$[11]^T$

$$= p_1(k) + 0.5\mu N e(k) s(k) \quad (31)$$

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 $p_0(k)$, $p_1(k)$ 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(30)式と(31)式をまとめて、

$$P_w(k+1) = P_w(k) + 0.5\mu e(k) S_{DA}(k) \quad (32)$$

$$S_{DA}(k) = N[\bar{s}(k), s(k)]^T \quad (33)$$

$$P_w(k) = [p_0(k), p_1(k)]^T \quad (34)$$

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 $S_{DA}(k)$ によって更新されている。

次にタップ数 N が2の場合について検討する。入力信号ベクトルを

$$S(k) = [s(k), s(k-1)]^T \quad (35)$$

とする。ここで、信号 $s(k)$, $s(k-1)$ を構成するビットは"0"と"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 $A_{vi}(k)$ 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、それらをTable 2のようにアルファベットで表して区別することにする。

今、一例として入力信号の組み合わせが以下の場合について考える。

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

Table 3 Access pattern of adaptive function space.

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

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

この入力信号ベクトルをシンボルを用いて表すと、

$$S_s = c \times (-2^0) + b \times 2^{-1} + a \times 2^{-2} + d \times 2^{-3} \quad (36)$$

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。

$$pa(k+1) = pa(k) + 0.5\mu \times 2 \times e(k)2^{-2} \quad (37)$$

$$pb(k+1) = pb(k) + 0.5\mu \times 2 \times e(k)2^{-1} \quad (38)$$

$$pc(k+1) = pc(k) - 0.5\mu \times 2 \times e(k)2^0 \quad (39)$$

$$pd(k+1) = pd(k) + 0.5\mu \times 2 \times e(k)2^{-3} \quad (40)$$

(37)式~(40)式をまとめると、

$$P_w(k+1) = P_w(k) + 0.5\mu N e(k) A_{ac}^T(k) F \quad (41)$$

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との対応関係をTable 3に示す。(41)式の $A_{ac}^T(k)$ 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。

$$A_{ac}(k) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (42)$$

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 $P_w(k)$ は

$$P_w(k) = [pa(k), pb(k), pc(k), pd(k)]^T \quad (43)$$

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、

$$P_w(k+1) = P_w(k) + 0.5\mu e(k) S_{DA}(k) \quad (44)$$

$$S_{DA}(k) = N A_{ac}^T(k) F \quad (45)$$

$$= [s_{DA,0}(k), \dots, s_{DA,2^N-1}(k)]^T \quad (46)$$

となる。ここで、 $A_{ac}^T(k)$ は $2^N \times B$ のアクセスマトリクス、適応関数空間 $P_w(k)$ は

$$P_w(k) = [p_0(k), p_1(k), \dots, p_{2^N-1}(k)]^T \quad (47)$$

スケーリングベクトルは、

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

である。また、出力信号 $y(k)$ は

$$y(k) = F^T A_{ac}(k) P_w(k) \quad (49)$$

$$= \frac{1}{N} S_{DA}^T(k) P_w(k) \quad (50)$$

となる。(44)式とLMSアルゴリズムの(5)式を比較すると、 $S_{DA}(k)$ はLMSアルゴリズムにおける入力信号ベクトル $S(k)$ 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の過程により導かれるため、ここでは結果のみを示す。

$$P'_w(k+1) = P'_w(k) + 2\mu e'(k) S'_{DA}(k) \quad (51)$$

$$S'_{DA}(k) = N A_{ac}^T(k) F' \quad (52)$$

$$= [s'_{DA,0}(k), \dots, s'_{DA,2^N-1}(k)]^T \quad (53)$$

ただし、入力信号の各ビットは値0,1である。ここで、 $A_{ac}^T(k)$ は $2^N \times B$ のアクセスマトリクス、適応関数空間 $P'_w(k)$ は

$$P'_w(k) = [p'_0(k), p'_1(k), \dots, p'_{2^N-1}(k)]^T \quad (54)$$

スケーリングベクトルは、

$$F' = [2^{-1}, 2^{-2}, \dots, 2^{-B}]^T \quad (55)$$

である。また、出力信号 $y'(k)$ は

$$y'(k) = F'^T A'_{ac}(k) P'_w(k) \quad (56)$$

$$= \frac{1}{N} S'^T_{DA}(k) P'_w(k) \quad (57)$$

となる。

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、我々は適応関数空間の更新状況を表すアクセスマトリクス $A_{ac}(k)$ 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DA適応フィルタは、(44)式、(46)式と(51)式、(53)式のように、入力信号 $S_{DA}(k)$ 、 $S'_{DA}(k)$ を用いて適応関数空間を更新することが示された。

4.2 収束条件の導出

まず、提案法の更新式である(44)式に出力計算式の(50)式を代入して、

$$\begin{aligned}
 P_w(k+1) &= P_w(k) + 0.5\mu e(k) S_{DA}(k) \quad (58) \\
 &= P_w(k) + 0.5\mu S_{DA}(k) \\
 &\quad \left[d(k) - \frac{1}{N} S_{DA}^T(k) P_w(k) \right] \quad (59) \\
 &= P_w(k) + 0.5\mu d(k) S_{DA}(k) \\
 &\quad - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) P_w(k) \quad (60) \\
 &= \left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] P_w(k) \\
 &\quad + 0.5\mu d(k) S_{DA}(k) \quad (61)
 \end{aligned}$$

となる。ここで、行列 I は $2^N \times 2^N$ の単位行列である。さて、適応関数空間の最適値を P_w^* とし適応関数空間誤差ベクトル $c(k)$ を次のように定義する。

$$c(k) = P_w(k) - P_w^* \quad (62)$$

この関係を用いて、(61)式は

$$\begin{aligned}
 P_w(k+1) &= \left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] \times \\
 &\quad \left[c(k) + P_w^* \right] + 0.5\mu d(k) S_{DA}(k) \quad (63) \\
 &= \left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] c(k) + P_w^* \\
 &\quad - 0.5\mu \left[\frac{1}{N} S_{DA}(k) S_{DA}^T(k) P_w^* - d(k) S_{DA}(k) \right] \quad (64)
 \end{aligned}$$

となり、両辺から P_w^* を引いて、

$$\begin{aligned}
 P_w(k+1) - P_w^* &= \left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] c(k) \\
 &\quad + 0.5\mu \left[d(k) S_{DA}(k) - \frac{1}{N} S_{DA}(k) S_{DA}^T(k) P_w^* \right] \\
 c(k+1) &= \left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] c(k) \\
 &\quad + 0.5\mu \left[d(k) S_{DA}(k) - \frac{1}{N} S_{DA}(k) S_{DA}^T(k) P_w^* \right] \quad (65)
 \end{aligned}$$

となる。ここで両辺の期待値をとり、 $c(k)$ と $S_{DA}(k)$ は直交原理¹²⁾により独立であると仮定すると、

$$\begin{aligned}
 E[c(k+1)] &= E \left[\left[I - 0.5 \frac{1}{N} \mu S_{DA}(k) S_{DA}^T(k) \right] c(k) \right] \\
 &\quad + 0.5\mu E \left[d(k) S_{DA}(k) - \frac{1}{N} S_{DA}(k) S_{DA}^T(k) P_w^* \right] \\
 &= \left[I - 0.5 \frac{1}{N} \mu E[S_{DA}(k) S_{DA}^T(k)] \right] E[c(k)] \\
 &\quad + 0.5\mu \left[E[d(k) S_{DA}(k)] - \frac{1}{N} E[S_{DA}(k) S_{DA}^T(k)] P_w^* \right] \quad (66) \\
 &= \left[I - 0.5 \frac{1}{N} \mu R \right] E[c(k)] + 0.5\mu \left[q - \frac{1}{N} R P_w^* \right] \quad (66)
 \end{aligned}$$

となる。なお、

$$q = E[d(k) S_{DA}(k)] \quad (67)$$

$$R = E[S_{DA}(k) S_{DA}^T(k)] \quad (68)$$

とおいた。しかし、正規方程式(導出過程を付録に示す)は

$$q = \frac{1}{N} R P_w^* \quad (69)$$

となるので、(66)式の右辺第2項は零となり

$$E[c(k+1)] = \left[I - 0.5 \frac{1}{N} \mu R \right] E[c(k)] \quad (70)$$

$$= \left[I - \mu_a R \right] E[c(k)] \quad (71)$$

と簡略化される。なお、

$$\mu_a = \frac{0.5}{N} \mu \quad (72)$$

とおいた。この式は適応関数空間誤差ベクトル $c(k)$ の更新式を表しており、時刻 k が経過するにつれて $c(k)$ が減少するかどうかは μ_a および R に依存していることがわかる。この性質を明確にするために、

$$R = Q D Q^T \quad (73)$$

と変形する。ただし、 Q は R の固有ベクトルを列ベクトルに持つ直交行列

$$Q^T = Q^{-1} \quad (74)$$

また、 D は R の固有値を対角要素とする対角行列

$$\begin{aligned}
 D &= \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_{2^N}) \quad (75) \\
 &\quad (\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{2^N})
 \end{aligned}$$

である。したがって、(71)式は

$$E[c(k+1)] = Q \left[I - \mu_a D \right] Q^T E[c(k)] \quad (76)$$

となる。(76)式より、時刻 k が経過するにつれて適応関数空間誤差 $E[c(k)]$ が減少するための条件は

$$0 < \mu_a < \frac{1}{\lambda_{\max}} \quad (77)$$

となる。ここで、 λ_{\max} は R の最大固有値である。

従来法の収束条件も提案法と同様の過程で導かれるため、ここでは結果のみを示す。適応関数空間誤差ベクトル $c'(k)$ の更新式は、

$$E[c'(k+1)] = \left[I - 2 \frac{1}{N} \mu' R' \right] E[c'(k)] \quad (78)$$

$$= \left[I - \mu'_a R' \right] E[c'(k)] \quad (79)$$

$$c'(k) = P'_w(k) - P_w^* \quad (80)$$

Table 4 Variance of eigen value of autocorrelation matrix.

Input variance	Conventional	Proposed
0.10	0.149	4.47e-4
0.05	0.375	2.72e-5
0.01	0.764	2.68e-6

$$\mu'_a = \frac{2}{N} \mu' \quad (81)$$

である。したがって、(79)式より収束条件は、

$$0 < \mu'_a < \frac{1}{\lambda'_{max}} \quad (82)$$

となる。ここで、 λ'_{max} は R' の最大固有値である。

提案法と従来法のいずれの場合も、新たに定義した入力信号ベクトルの自己相関行列 R 、 R' の固有値分布が小さいほど高速な収束速度を示すため、提案法や従来法の固有値分布を検証することにより収束速度を評価することができる。

5. 収束速度の評価

5.1 収束速度の劣化要因

計算機シミュレーションにより、タップ数 $N=6$ に対する拡張された入力信号ベクトルの自己相関行列の固有値分散を求めた結果をTable 4に示す。なお、入力信号は白色ガウス信号で分散は0.10,0.05,0.01である。これより従来法の分散は非常に大きく、また入力信号の分散を小さくすると固有値分散が大きくなるという負の相関を有する。これに対して、提案法の固有値分散は非常に小さく、また入力信号の分散を小さくすると固有値分散も小さくなるという正の相関を有していることがわかる。前章において、分散演算型LMS適応フィルタの収束速度は拡張された入力信号の固有値分布が小さいほど高速な収束速度を示すことを示したが、Table 4より提案法は従来法よりも固有値の分散つまり分布が小さいため高速な収束速度を有することになる。

従来法の固有値分散が大きくなる原因は、入力信号の符号化方法にある。簡単のためにタップ数 $N=1$ 、語長4bitとし信号値 -0.1875 が入力された場合について考える。拡張されたDA適応フィルタの入力信号ベクトル $S'_{DA}(k)$ は

$$S'_{DA}(k) = N A'^T_{ac}(k) F' \quad (83)$$

Table 5 Total of average of input signal vector element.

Tap number	Conventional	Proposed
2	2.0	0.0
3	3.0	0.0
4	4.0	0.0
5	5.0	0.0

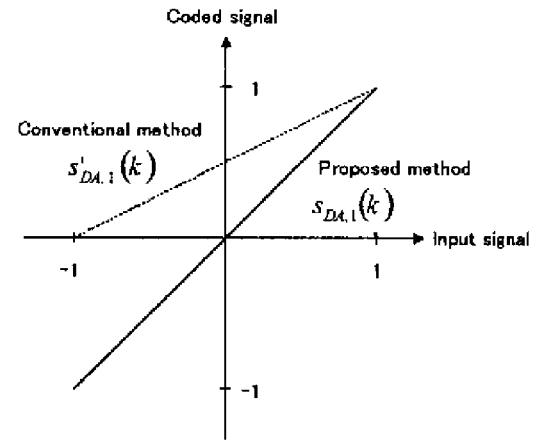


Fig. 8 Relation between input signal and coded signal.

$$\begin{aligned} \begin{bmatrix} s'_{DA,0}(k) \\ s'_{DA,1}(k) \end{bmatrix}^T &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} F' \\ &= \begin{bmatrix} 0.5625 \\ 0.3750 \end{bmatrix} \end{aligned} \quad (84)$$

となり値が大きく異なる。タップ数 $N=1$ に対する、入力信号と符号化された入力信号の関係をFig. 8に示す。これより、従来法の符号化方式では入力信号が負であっても符号化後は正の値として表現されている。このことは、平均0の入力信号を符号化しても平均値は0とはならずオフセット値を有することを意味する。これに対して2の補数形式では正負の値が表現可能であるためオフセット値は0となる。タップ数 $N=2,3,4,5$ に対する入力信号ベクトルの平均値の総和を計算機シミュレーションにより求めた結果をTable 5に示す。なお、入力信号は平均0、分散0.1の白色ガウス信号である。これより、従来法ではタップ数 N が増加するにしたがいオフセット値が増加している。ある入力信号要素に対するオフセット値は、その要素をアクセスする確率を $1/2^N$ とすると(83)式より

$$N \left\{ \frac{1}{2^N} \times 2^{-1} + \frac{1}{2^N} \times 2^{-2} + \dots + \frac{1}{2^N} \times 2^{-B} \right\}$$

$$= \frac{N}{2^N} \sum_{i=1}^B 2^{-i} \approx \frac{N}{2^N}$$

となるため、 2^N 要素の総和は

$$\frac{N}{2^N} \times 2^N = N \quad (85)$$

となり Table 5 に一致する。これに対して提案法は

$$N \left\{ \frac{1}{2^N} \times (-2^0) + \frac{1}{2^N} \times 2^{-1} + \dots + \frac{1}{2^N} \times 2^{-B+1} \right\}$$

$$= \frac{N}{2^N} \left\{ -2^0 + \sum_{i=1}^{B-1} 2^{-i} \right\} \approx 0$$

となり、タップ数 N に依存せず 0 である。

従来法において固有値の分散が大きくなる原因は、入力信号がオフセットを有するために自己相関行列の固有値の一つが他に比べて非常に大きくなるためである¹³⁾。したがって、(82)式の表す収束を保証するステップサイズパラメータの上限はこの最大固有値で制限されるため、他の小さい固有値に対する収束が極端に遅くなる。しかし、提案法では固有値がほぼ同程度の値となりステップサイズパラメータに対して全ての固有値が同等の収束を示すために良好な収束速度を有する。

5.2 有色信号に対する収束速度

有色性を有する入力信号について、自己相関行列の固有値の分散を求めた結果を Table 6 に示す。ここで有色信号は係数 0.99 の 1 次 AR 過程を用いており、未知システムのタップ数 $N=4$ である。これより、提案法の固有値の分散が最も小さいことがわかる。この原因は、DA 適応フィルタと LMS 適応フィルタにおける入力信号の取り扱いが異なるためである。LMS 適応フィルタと DA 適応フィルタの入力信号の関係を Fig. 9 に示す。LMS 適応フィルタは連続して入力される信号 $s(k) \sim s(k-N+1)$ を入力信号ベクトルとして用いるが、信号の有色性は連続するサンプル間の相関であるため、LMS 適応フィルタの入力信号は有色性の影響を直接受ける。これに対して、DA 適応フィルタの入力信号はアドレスベクトルのビットパターンにより入力信号が決定される。この場合、有色性信号に対しても下位ビット程 0 と 1 がランダムに生起するため、 $S_{DA}(k)$ の有色性は軽減される。その結果、提案法における固有値の分散は LMS よりも小さくなり高速な収束速度を示すのである。従来法においても有色性が軽減される白色化の効果は提案法と同様であるが、入力信号のオフセットによって自己相関行列の固有値の分布が大きいため有色信号に対しても収束速度は大きく劣化する。

Table 6 Variance of eigen value of autocorrelation matrix for colored process.

Variance	LMS	Conventional	Proposed
0.10	5.90e-4	9.08e+1	4.13e-4
0.05	1.48e-4	9.50e+1	1.03e-4
0.01	5.90e-6	1.01e+2	4.13e-6

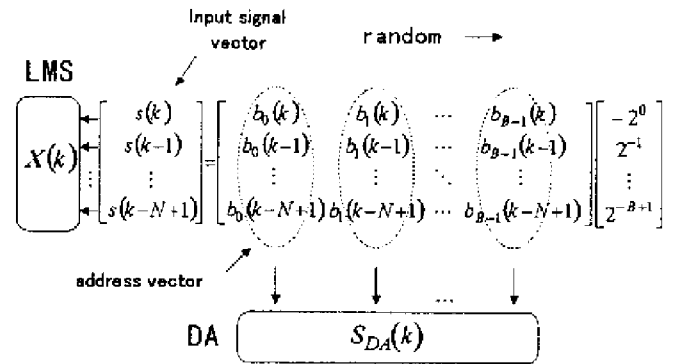


Fig. 9 Difference of input signal vector between LMS and DA.

6. あとがき

本報告では、これまで行われていなかった分散演算型 LMS 適応フィルタの収束速度について解析的に検討した。そのために、まず分散演算型 LMS 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し、この更新式を LMS アルゴリズムと比較することによって、適応関数空間全体に対する入力信号ベクトルを新たに定義した。そして、拡張した更新式に対する分散演算型 LMS 適応フィルタの収束式と収束条件を導出した。これより、収束速度は入力信号ベクトルの自己相関行列の固有値分布に依存し、固有値が広く分布するほど収束速度は劣化することを明らかにした。次に、新たに定義した入力信号ベクトルの性質より、提案法に比較して従来法の固有値が広く分布することを示した。この原因は、従来法の符号化方式が特殊なために符号化された入力信号にオフセットが加わり、ある固有値が他に比べて非常に大きな値を持つためである。これにより、従来法の収束速度は提案法に比較して大幅に劣化することになる。さらに、入力信号が有色信号の場合について収束速度を検討した。

今後は、有色信号に対する収束速度の詳細な検討と適応関数空間を分割化したマルチブロック構成に対する収束速度について検討する予定である。

参考文献

- 1) 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.
- 2) C.F.N. Cowan and J. Mavor, "New digital adaptive-filter implementation using distributed-arithmetic techniques," IEE Proc., vol.128, Pt.F, no.4, pp.225-230, Aug. 1981.
- 3) 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.
- 4) 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.
- 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) 豊田真嗣, 高橋 強, 恒川佳隆, 三浦 守, "ハーフメモリアルゴリズムを用いた分散演算型LMS適応フィルタのVLSI実現," 信学技報, DSP98-23, May, 1998.
- 7) 高橋 強, 豊田真嗣, 恒川佳隆, 三浦 守, "分散演算型LMS適応フィルタの収束特性解析," 計測自動制御学会東北支部第178回研究集会, Nov. 1998.
- 8) 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.
- 9) 恒川佳隆, 高橋 強, 豊田真嗣, 三浦 守, "分散演算によるマルチプライヤレスLMS適応フィルタの高性能アーキテクチャ," 信学論(A), vol.J-82-A, no.10, pp.1518-1528, Oct.1999.
- 10) Keshab K.,Parhi, "VLSI digital signal processing systems:design and implementation," John Wiley & Sons,Inc., New York, 1999.
- 11) C. F. N. Cowan and P. M. Grant, "Adaptive Filters," Prentice-Hall, Inc.,New Jersey, 1985.

12) Simon Haykin, "Introduction to Adaptive Filters," Macmillan publishing Company, New York, 1984.

13) Gilbert Strang, "Linear Algebra and its Applications," Academic Press, New York, 1976.

A 正規方程式の導出

(69)式に示したDA適応フィルタの正規方程式を導く。誤差の自乗平均値を最小化することを考えると評価量 J は,

$$J = E[e^2(k)] = E[(d(k) - y(k))^2] \quad (86)$$

で与えられる。ここで、 $d(k)$ は所望信号、 $y(k)$ はフィルタ出力を表す。これに出力式

$$y(k) = \frac{1}{N} S_{DA}^T(k) P_w(k) \quad (87)$$

を代入すると,

$$\begin{aligned} J &= E\left[\frac{1}{N^2} P_w^T(k) S_{DA}(k) S_{DA}^T(k) P_w(k)\right] \\ &- 2E\left[\frac{1}{N} d(k) S_{DA}^T(k) P_w(k)\right] + E[d^2(k)] \quad (88) \\ &= \frac{1}{N^2} P_w^T(k) E[S_{DA}(k) S_{DA}^T(k)] P_w(k) \\ &- 2\frac{1}{N} E[d(k) S_{DA}^T(k)] P_w(k) + E[d^2(k)] \quad (89) \\ &= \frac{1}{N^2} P_w^T(k) R P_w(k) \\ &- 2\frac{1}{N} q P_w(k) + E[d^2(k)] \quad (90) \end{aligned}$$

となる。ここで,

$$R = E[S_{DA}(k) S_{DA}^T(k)] \quad (91)$$

$$q = E[d(k) S_{DA}^T(k)] \quad (92)$$

とおいた。(90)式は適応関数空間に関する2次式になっていることがわかる。したがって、 J は $P_w(k)$ に関する凸関数で唯一の最小値を持つ。ここでは、時刻 k において J を最小にする推定ベクトルを P_w^* と表記する。 P_w^* は、(90)式の両辺を $P_w(k)$ で偏微分して,

$$\frac{\partial J}{\partial P_w(k)} = 2\frac{1}{N^2} R P_w(k) - 2\frac{1}{N} q \quad (93)$$

となり、これを零とおいて正規方程式は

$$q = \frac{1}{N} R P_w^* \quad (94)$$

となる。