

部分的転置形構造に基づく分散演算形適応デジタルフィルタ の高性能アーキテクチャ

High-Performance Architecture for LMS Adaptive Filter Using Distributed Arithmetic Based on Partially Transposed Structure

○猪股了*, 佐藤辰海*, 高橋強**, 恒川佳隆*

○ Inomata Ryo*, Sato Tatsumi*, Takahashi Kyo**,
Tsunekawa Yoshitaka*

*岩手大学, **岩手県工業技術センター

*Iwate University, **Iwate Industrial Research Inst

キーワード: 分散演算 (distributed arithmetic), 同時更新 (simultaneous update),
カットセットリタイミング (cutset retiming), 準奇対称性 (pseudo-odd symmetry property),
収束特性 (convergence characteristic)

連絡先: 〒 020-8551 盛岡市上田 4-3-5 岩手大学工学部 恒川佳隆
Tel.: (019)621-6468, Fax.: (019)621-6468, E-mail: tsune@iwate-u.ac.jp

1. はじめに

適応フィルタ (Adaptive Digital Filter, ADF) はノイズキャンセラ, エコーキャンセラ, 適応等化器などの様々な分野で応用されている. そのことから, 小規模ハードウェア, 低消費電力, 良好な収束特性, 短い出力滞在時間などが要求される. しかし, これらを同時に満たすことは困難であり, 高性能なアルゴリズムや効果的なアーキテクチャが望まれている.

これまで, LMS 適応フィルタ¹⁾ に分散演算 (Distributed Arithmetic, DA) を適用した, 分散演算形 LMS 適応フィルタ (LMS Adaptive Digital Filter using Distributed Arithmetic, DA-ADF) が提案された. 分散演算は出力計算における係数ベクトルと入力信号ベクトルの内積演算に適用し, 出力信号は N 次入力信号ベクトルのビットパターンで指定される部分積のシフト

加算により求められ, 2^N 個の部分積は適応プロセスにより逐次更新する. ここで部分積の全集合を全適応関数空間 (Whole Adaptive Function Space, WAFS) と呼ぶ. また, 高次の場合でのハードウェア規模増加と収束速度劣化を抑制する手法として, WAFS を分割するマルチメモリブロック構造 (Multi-memory Block Structure) を適応した DA-ADF (MDA-ADF) があり, 出力計算と更新動作を並列に実行することで高サンプリングレートを実現可能な同時更新アルゴリズム (Simultaneous Update Algorithm) を適用した MDA-ADF (SMDA-ADF)²⁾, カットセットリタイミングを適用することで出力滞在時間を短縮した, 部分的転置形構造を有する SMDA-ADF (NCMDA-ADF, 以下, 従来型適応フィルタ) が提案されている³⁾.

本報告では, 従来型適応フィルタの構造に対

して準奇対称性を利用し、収束速度の向上を図る。さらに、4入力2出力加算器(4-2加算器)を適用することで、ハードウェア規模の増加を抑制した構成を提案する。最後に、収束特性評価およびVLSI評価を行うことで、本提案型アーキテクチャの有効性を明らかにする。

2. 分散演算形 LMS 適応フィルタ

2.1 分散演算の適用

分散演算は内積演算をテーブルルックアップにより効率よく計算する手法であり、原理的には処理時間は語長 B にのみ依存する。以下、フィルタのタップ数を N 、各信号の語長を B とすると、フィルタ出力 $y(k)$ は N 次係数ベクトル $\mathbf{H}(k)$ と N 次入力ベクトル $\mathbf{X}(k)$ の内積演算であり、式 (1) で表される。

$$y(k) = \mathbf{H}^T(k) \mathbf{X}(k) \quad (1)$$

$$\mathbf{H}(k) = [h_0(k), h_1(k), \dots, h_{N-1}(k)]^T \quad (2)$$

$$\mathbf{X}(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T \quad (3)$$

ここで分散演算は式 (1) に対して適用される。また T は転置を表す。出力計算は B 個のアドレスベクトルによって順次指定される部分積を WAFS から読み出してシフト加算により行う。更新動作は、誤差信号 $e(k)$ を用いて出力計算時に用いた部分積を順に更新する。時刻 k における B 個の部分積は WAFS の部分集合であり、これを適応関数空間 (Adaptive Function Space, AFS) と呼ぶ。式 (3) の N 次入力信号ベクトルは以下のように表わされる。

$$\mathbf{X}(k) = \mathbf{A}(k) \mathbf{F} \quad (4)$$

また、ビットパターンに分解した $N \times B$ 次アドレスマトリクス $\mathbf{A}(k)$ 、スケーリングベクトル \mathbf{F} は、式 (5)、式 (6) で表される。

$$\mathbf{A}(k) = \begin{bmatrix} b_0(k) & \cdots & b_0(k-N+1) \\ b_1(k) & \cdots & b_1(k-N+1) \\ \vdots & \ddots & \vdots \\ b_{B-1}(k) & \cdots & b_{B-1}(k-N+1) \end{bmatrix}^T \quad (5)$$

$$\mathbf{F} = [-2^0, 2^{-1}, \dots, 2^{-(B-1)}]^T \quad (6)$$

ここで、 $b_i(k)$ 、 $i = 0, \dots, B-1$ 、は $x(k)$ の i 番目のビットである。

マルチメモリブロック構造は、DA-ADF の高次における RAM 容量の増加抑制と更新確率の減少に伴う収束速度の劣化を補償する。分割数を M 、アドレス線数を $R = N/M$ とすると、適応関数空間は M 個に分割される。また、 R 個のアドレスベクトルで指定される m 番目の適応関数空間 $\mathbf{P}_m(k)$ は式 (7) となる。

$$\begin{aligned} \mathbf{P}_m(k) &= [p_{m0}(k), p_{m1}, \dots, p_{m(B-1)}(k)]^T \\ &= \mathbf{A}_m^T(k) \mathbf{H}_m(k) \end{aligned} \quad (7)$$

$$m = 0, \dots, M-1$$

このように分割することで、容量 2^R の WAFS を M 個有するため、総容量は $M \cdot 2^R$ となる。またフィルタ出力 $y(k)$ 、更新式、誤差信号 $e(k)$ は、所望信号を $d(k)$ とすると、次式で表される。

$$y(k) = \sum_{m=0}^{M-1} \mathbf{F}^T \mathbf{P}_m(k) \quad (8)$$

$$\mathbf{P}_m(k+1) = \mathbf{P}_m(k) + 0.5\mu R e(k) \mathbf{F} \quad (9)$$

$$e(k) = d(k) - y(k) \quad (10)$$

2.2 同時更新アルゴリズム

同時更新アルゴリズムは、適応フィルタの処理過程である出力計算と更新動作を並列化する手法である。これにより1サンプルによる処理時間が短縮可能となる。また、並列化する手法では、同時更新アルゴリズムの他にディレードアップデートと呼ばれる手法もあり、この2つの手法は1時刻前の出力信号を用いて誤差信号を求めている。そのためディレードアップデートでは過去の誤差信号を使用することにより収束速度が劣化してしまう。しかし、同時更新アルゴリズムでは、出力計算はLSB側から、更新動作はMSB側から順次実行し、ビットレベルでのアップデートを行う。これにより、出力計

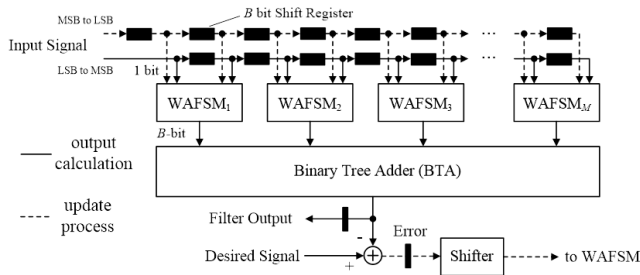


Fig. 1 Structure of SMDA-ADF

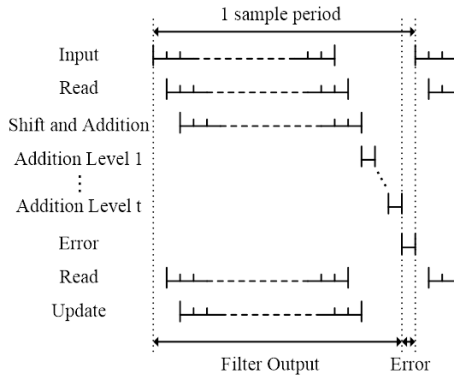


Fig. 2 Timing chart of SMDA-ADF

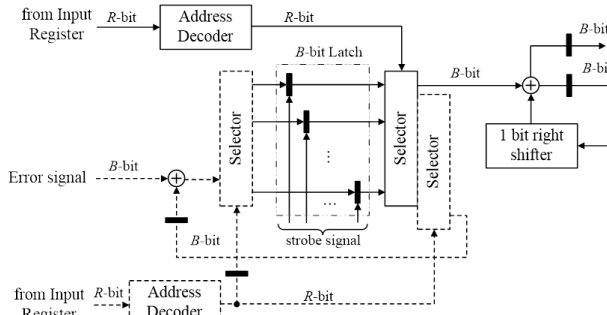


Fig. 3 WAFSM

算に更新された関数空間を使用でき、収束速度の劣化を抑制することが可能である。

2.3 SMDA-ADF

SMDA-ADFの構成を図1に、タイミングチャートを図2に示す。ここで、 M 個のWAFSM (WAFS Module)は部分積を格納するメモリ、入力信号により部分積を選択するセレクタ、部分積を加算するシフト加算器などから構成される。図3にWAFSMの構成を示す。WAFSMは更新機能も有しており、1時刻前の出力計算に寄与した部分積を出力計算と同時に更新する。し

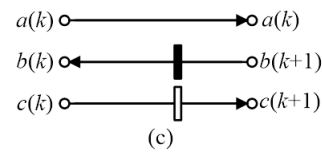
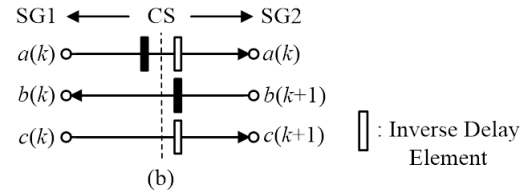
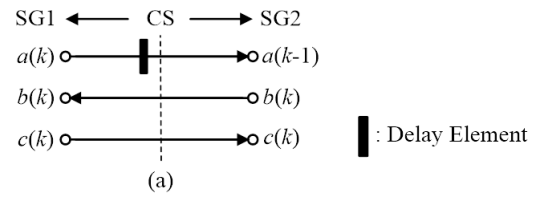


Fig. 4 cutset retiming

かしSMDA-ADFは、高次でBTA(Binary Tree Adder)の段数が増加してしまうために出力滞在時間が増加し、サンプリングレートが減少する。

2.4 カットセットリタイミング

転置形構造は従来の構造に対してカットセットリタイミングを実行することにより導出される。カットセットリタイミングとは、入出力の時間差関係を保持したまま挿入されている遅延器を再配置する手法である。図4にその仕組みを示す。まず図4(a)に示すように、シグナルフローグラフSGにカットセットCSを設定して、SG1とSG2に分割する。次いで、図4(b)に示すようにSG2からSG1に流れる信号線に対しては遅延器、SG1からSG2に流れる信号線に対しては逆時間の遅延器を挿入する。ここで逆時間の遅延器とは、時刻 k の信号 $n(k)$ に対して $n(k+1)$ を出力するような要素である。最後に、遅延器と逆時間の遅延器が存在する信号線に対してこれらを打ち消すことで、図4(c)の構成が得られる。以上がカットセットリタイミングのプロセスであるが、図4(c)の構成は逆時間の遅延器が挿入されているため実現不可能である。実際にカットセットリタイミングを適

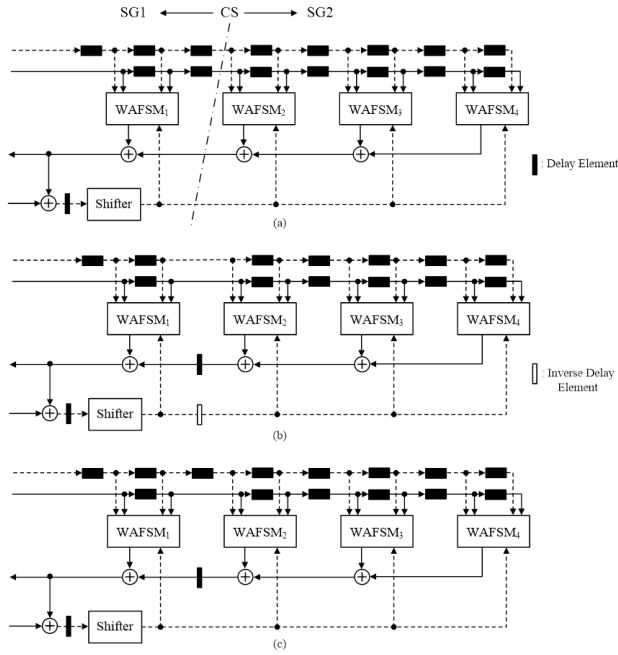


Fig. 5 Flow of cutset retiming

用し、逆時間の遅延器が挿入された際には時間のシフトを行い、因果律を満たす構成を導出しなければならない。

2.5 部分的転置形構造の導出

実際にSMDA-ADFに対してカットセットリタイミングを実行する。簡単化のためにタップ数を8、図5のBTA部を加算器が縦続に接続されているものとして考える。

まず、図5(a)に示すようにカットセットCSをWAFSM₁とWAFSM₂の間に設定しSG1とSG2に分割する。次いで、SG2からSG1に流れる信号線には遅延器、SG1からSG2に流れる信号線には逆時間の遅延器を挿入する。ここで、図5(b)では誤差フィードバックラインに逆時間の遅延器が挿入されており、このままでは因果律を満たさず実現不可能なため、誤差信号と入力信号の時間差を保持したまま時間のシフトを行う。これにより、更新用の遅延ラインには遅延器が挿入され、誤差フィードバックラインの逆時間の遅延器が打ち消されることで、図5(b)の非因果的な構成は図5(c)の因果律を満たす構成となる。最後に、フィルタ部をパ

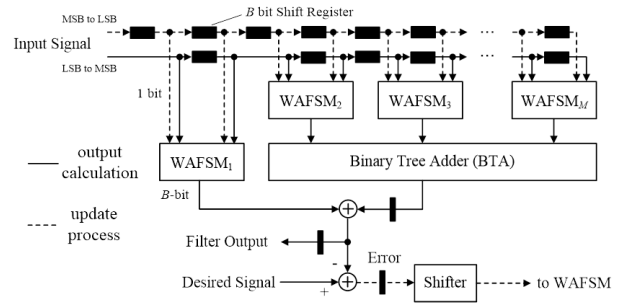


Fig. 6 Structure of NCMDA-ADF

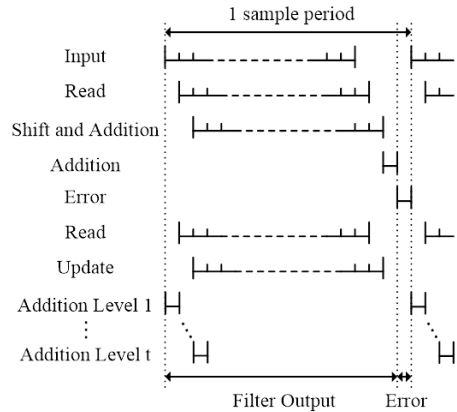


Fig. 7 Timing chart of NCMDA-ADF

イプライン化するためにWAFSM₂~WAFSM₄の出力加算ラインをBTA化する。

以上より導出した部分的転置形構造を有するSMDA-ADF(NCMDA-ADF)を図6に示す。この構成においてモジュール出力は、入力信号のビットパターンのLSB側から順次部分積を読み出してシフト加算することにより得られ、フィルタ出力は時刻 k におけるWAFSM₁の出力に、時刻 $k-1$ におけるWAFSM₂~WAFSM_Mのモジュール出力和を加えて得られる。また、更新方向は出力計算とは異なり、MSB側から順次行う。遅延ラインは出力計算と更新の2ラインを有する。部分的転置形構造はWAFSM₁とWAFSM₂の間だけにカットセットリタイミングを適用し、フィルタ部のWAFSM₂以降に多くのモジュールを配置でき、係数遅延を最小に抑えた特殊な転置形構造である。

また、NCMDA-ADFのタイミングチャートを図7に示す。NCMDA-ADFはWAFSM₁とBTAを並列に動作させている。SMDA-ADFは

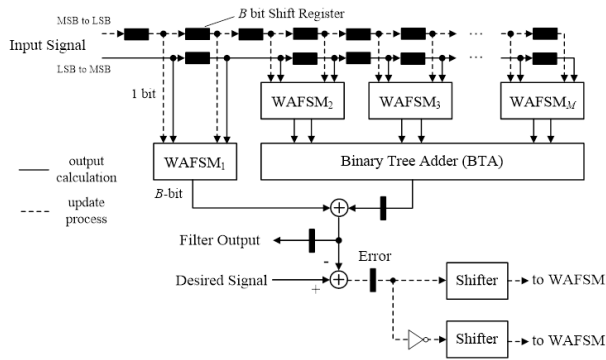


Fig. 8 Structure of HNCMDA-ADF

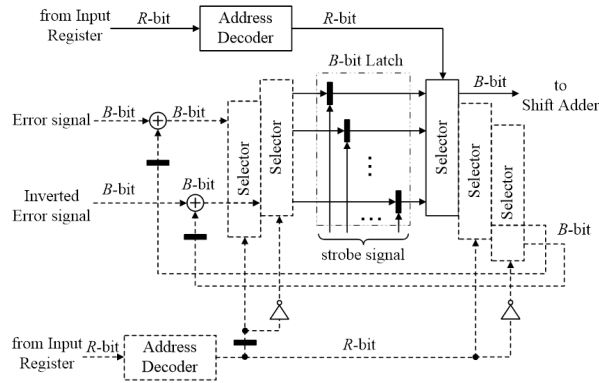


Fig. 9 WAFS of HNCMDA-ADF

高次において BTA 段数が増加し出力滞在時間が増加してしまう。しかし、NCMDA-ADF は BTA の出力滞在時間が WAFSM₁ の出力滞在時間以内であれば、モジュール出力の加算回数は 1 回のみで済むことから、出力滞在時間の短縮とサンプリングレートの向上を可能とする。

図 6 の構造から導出されるアルゴリズムを以下に示す。部分積の指定は入力信号ベクトルのビットパターンを用いる。フィルタ出力は式 (11) で示される。

$$y(k) = y_0(k) + \sum_{m=1}^{M-1} y_m(k-1) \quad (11)$$

$$y_m(k) = \mathbf{F}^T \mathbf{P}_m(k) \quad (12)$$

また、部分積の集合である m 番目の適応関数空間の更新式は、式 (13) のようになる。

$$\mathbf{P}_m(k+1) = \mathbf{P}_m(k) + 0.5\mu R e(k-1) \mathbf{F} \quad (13)$$

3. 提案型適応フィルタ

3.1 準奇対称性の利用

2 の補数形式を用いた分散演算形適応フィルタのアルゴリズムでは、収束状態の適応関数空間においてアドレスがビット反転の関係にある部分積は、互いに異符号で絶対値が近似的に等しい準奇対称性を有することが知られている⁴⁾。

提案法では、更新動作時に通常の更新要素と準奇対称の関係にある要素を同時に更新する。これまでは所望信号 $d(k)$ と出力信号 $y(k)$ の差から誤差信号 $e(k)$ を一つだけ求めていたが、提案法では $e(k)$ と $-e(k)$ の二つを同時に求める。 $e(k)$ は通常の更新要素であるアドレスベクトルによって指定される部分積を更新する誤差信号で、 $-e(k)$ は準奇対称の更新要素、つまりビット反転を行ったアドレスベクトルによって指定される部分積を更新する誤差信号である。適応関数空間の 2 箇所を同時に更新するため、収束速度は従来の 2 倍に向上する。図 8 に準奇対称性を利用した NCMDA-ADF(HNCMDA-ADF) の構成を、図 9 に準奇対称性を利用した WAFS を示す。

3.2 4 入力 2 出力加算器

提案型適応フィルタでは 4 入力 2 出力加算器 (4-2 加算器) を BTA とシフト加算器に用いる。4-2 加算器は桁上げ保存形式を用いており、桁上げ信号を伝搬させずに次の加算に持ち越して処理することが可能である。また、4-2 加算器はハードウェア効率がよく、多入力の加算に使用する場合、CLA 加算器を用いる場合と比べてハードウェア量を削減可能となる。よって、提案型適応フィルタでは CLA 加算器を使用していた部分と 4-2 加算器を適切に置き換えることでハードウェア利用の効率化を図る。

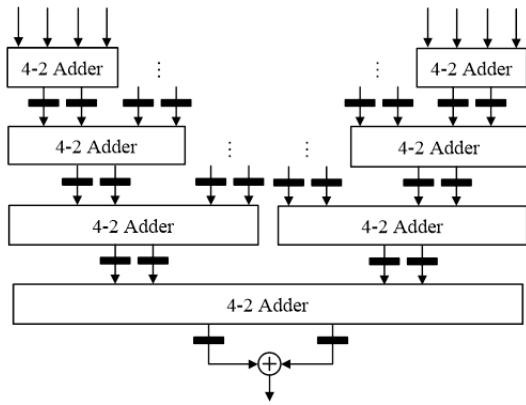


Fig. 10 BTA using 4-2 adder

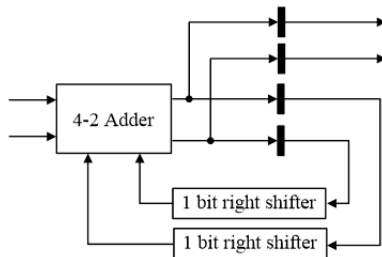


Fig. 11 Shift Adder using 4-2 adder

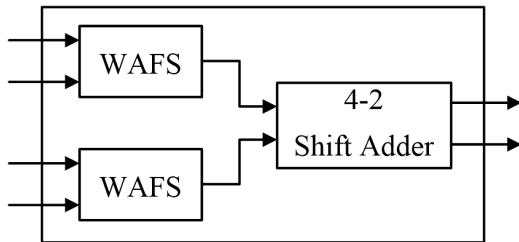


Fig. 12 Proposed WAFSM

3.3 アーキテクチャ

提案型適応フィルタはHNCMDA-ADFのフィルタ部を改良する。改良点はBTAとシフト加算器についてであり、4-2加算器を使用した構成を提案する。

まずBTAについて、従来のBTAではCLA加算器を $M - 1$ 個使用していたのに対し、提案型適応フィルタに用いるBTAは4-2加算器を $M/2 - 1$ 個とCLA加算器1個で構成でき、BTAのみの比較では面積を約25%削減可能である。4-2加算器を使用したBTAを図10に示す。

従来のWAFSMは、WAFSとシフト加算器をそれぞれ1つずつ使用して構成していた(図3)。

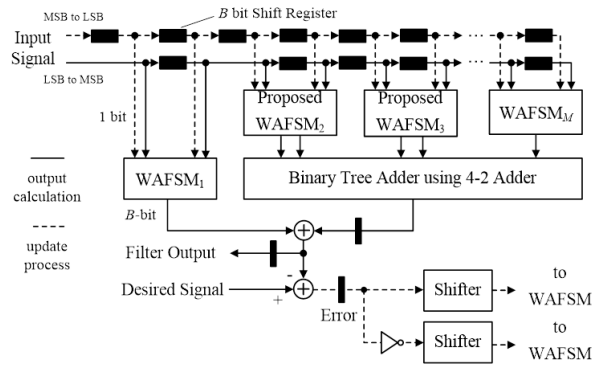


Fig. 13 Proposed Filter

ここでは図11に示す4-2加算器を用いたシフト加算器を使用して、2つのWAFSに対して1つのシフト加算器を使用する新たなWAFSMを提案する。その構成を図12に示す。提案型WAFSMを使用することで、使用するシフト加算器の総数を削減することができる。

準奇対称性を利用し、4-2加算器を使用したBTA、シフト加算器を適用した提案型適応フィルタの構成を図13に示す。この構成においてWAFS₁とWAFS_Mの部分には従来のWAFSMを使用することとする。理由として、WAFS₁に新たなWAFSMを使用するとシフト加算後の出力が2つあるため、加算を1回追加する必要がある。出力滞在時間が増加してしまうからである。また、図13では分割数が偶数の場合を想定しているが、奇数の場合はWAFS_Mに対しても4-2シフト加算器を適用することができる。

提案型適応フィルタは4-2加算器を適用することで出力滞在時間は従来のまま、ハードウェア規模を削減できる。

4. 収束特性評価

各適応フィルタの収束特性をシステム同定問題に対する計算機シミュレーションにより評価する。評価対象はLMS-ADF、従来型適応フィルタのNCMDA-ADF、提案型適応フィルタのProposedである。評価量はインパルス応答誤差比(Impulse Response Error Ratio, IRER)、未

Table 1 Comparison of the VLSI evaluations ($N = 128, B = 16$)

| | LMS-ADF | NCMDA-ADF | Proposed |
|----------------------------|---------|-----------|----------|
| Number of address lines | - | 2 | 2 |
| Number of divisions | - | 64 | 64 |
| Sampling rate [MHz] | 2.41 | 2.78 | 2.78 |
| Latency [ns] | 302 | 342 | 342 |
| Power dissipation [mW/MHz] | 781 | 3,753 | 3,888 |
| Area [mm ²] | 102.31 | 27.51 | 29.76 |
| Number of gates | 932,828 | 243,955 | 255,549 |

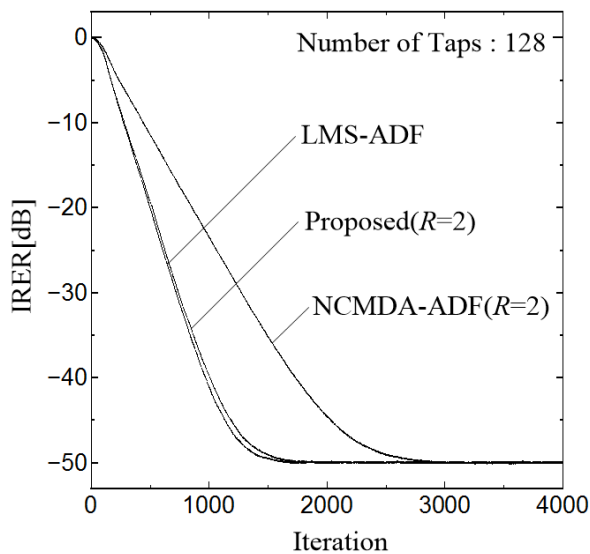


Fig. 14 Comparison of the convergence characteristics

知システムは低域通過 FIR フィルタ，入力信号は $-1\sim 1$ の範囲の値を取る一様乱数とした．未知システムの出力には観測雑音として -50dB の入力とは無相関な白色雑音を加え，ステップサイズパラメータは IRER が -50dB となるように選択した．またタップ数は全て 128 とし，NCMDA-ADF，Proposed のアドレス線数は 2 である．

図 14 に収束特性を示す．Proposed は準奇対称性を利用しているため，NCMDA-ADF と比較して約 2 倍の収束速度を示している．また Proposed は LMS-ADF と比較しても収束速度はほぼ同等であり，提案型適応フィルタは良好な収束速度を有することが分かる．

5. VLSI 評価

LMS-ADF，従来型の NCMDA-ADF，提案型適応フィルタの Proposed を VLSI 設計システム PARTHENON により設計および評価を行う．設計ルールは $0.6\mu\text{m}$ CMOS スタンダードセルとし，電源電圧は 5.0V である．演算に用いたデータ形式は 2 の補数表現による 16 ビット固定小数点型であり，タップ数は全て 128 とし，NCMDA-ADF，Proposed のアドレス線数は 2 とした．表 1 に VLSI 評価結果を示す．まず，LMS-ADF と分散演算を適用している他 2 つを比較すると，面積とゲート数は 70 % 以上の削減が可能となる．分散演算を適用することでマルチプライヤレスな構成となるためハードウェア規模は大幅に削減可能である．次に，NCMDA-ADF と Proposed の比較を行う．Proposed は，消費電力が約 3.6 %，面積が約 8.2 %，ゲート数が約 4.8 % 増加している．Proposed は 4-2 加算器を使用することで面積等を約 8 % 程度の増加に留ませた上で，収束速度を約 2 倍に向上させるため，ハードウェア効率の良い構成である．

6. まとめ

本報告では，従来の NCMDA-ADF に準奇対称性を利用した．さらに 4-2 加算器を適用することで，出力滞在時間はそのままハードウェア規模の増加を抑制しながら収束速度を向上させる構成を提案した．このことから，提案型適応フィルタは適応フィルタに要求される多くの

性能を同時に満たすことが可能な高性能アーキテクチャであることを明らかにした。

今後の課題としては、カットセットリタイミングの適用による転置形構造の導出は自由度が高いことから、複数個所に設定するなど他の構造について検討を行うことである。

参考文献

- 1) B. Widrow and M. E. Hoff, "Adaptive Switching Circuit," IRE EWSCON Conv. Rec., pp96-104, 1960.
- 2) 橋内慎次郎, 内田勝也, 佐藤慎悟, 高橋 強, 恒川佳隆, "分散演算を用いたパイプライン LMS 適応デジタルフィルタの高性能アーキテクチャ", 計測自動制御学会東北支部第 242 回研究集会, 242-2, May 13, 2008.
- 3) 佐藤辰海, 高橋強, 恒川佳隆, "転置形構造を有する分散演算形 LMS 適応フィルタ", 平成 24 年度電気関係学会東北支部連合大会講演論文集, 1|04, Aug 30, 2012.
- 4) 高橋 強, 恒川佳隆, 豊田真嗣, 三浦守, "ハーフメモリアルゴリズムに基づく分散演算形 LMS 適応フィルタの高性能アーキテクチャ", 信学論 (A), vol.J84-A, no.6, pp777-787, June 2001.