

RCD-NCLMS適応デジタルフィルタの高性能 パイプラインアーキテクチャ

High-performance Pipelined Architecture of the RCD-NCLMS Adaptive Digital Filter

○菅野 大輔[†], 高橋 強^{††}, 恒川 佳隆[†]

○Daisuke Kanno[†], Kyo TAKAHASHI^{††}, Yoshitaka TSUNEKAWA[†]

[†]岩手大学, ^{††}岩手県立産業技術短期大学校

[†]Iwate University, ^{††}Iwate Industrial Technology Junior College

キーワード: カットセットリタイミング(cutset retiming), NCLMS(Non-canonical LMS), 出力滞在時間(output latency), 収束速度(convergence speed), パイプラインアーキテクチャ(pipelined architecture)

連絡先: 〒020-8551 盛岡市上田4-3-5 岩手大学工学部

菅野大輔, Tel.:(019)621-6468, Fax.:(019)621-6468, E-mail: t3304009@iwate-u.ac.jp,

1. まえがき

適応フィルタは, エコーキャンセラやノイズキャンセラ, 適応等化器など幅広く用いられており, その応用範囲を広げている. この適応フィルタを実現するには高速なサンプリングレートや短い出力滞在時間(Latency), 小規模ハードウェア, 低消費電力, 良好な収束特性など, 多くの性能が要求される. しかし, これらの要求される性能は互いに相反する性能を含むため, これらを同時に満足させることは極めて困難である.

LMS適応デジタルフィルタ(LMS Adaptive Digital Filter, LMS-ADF)の構造にカットセットリタイミングを適用して得られるNCLMS適応デジタルフィルタ(Non-canonical LMS Adaptive Digital Filter, NCLMS-ADF)が提案されている¹⁾²⁾³⁾. この構成は, フィルタ出力部がタップ間隔のパイプライン構成となるため出力滞在時間が非常に短い

という長所を有する. しかし, 一方ではLMS-ADFよりも収束速度が大幅に劣化するということが知られている. この理由は, NCLMS-ADFの構成が通常のLMS-ADFとは異なり, 各タップ出力の加算のラインに“ディレー・アダー・ライン”と呼ばれる, 加算器と遅延器を連続に接続した構成を有する. これにより, NCLMS-ADFの出力はディレー・アダー・ラインを通過したタップ出力の和となるため, 係数の遅延(係数遅延)による誤差が含まれる. そして, この係数遅延は安定性を保証するステップサイズパラメータの上限を低下させること, 遅延量が増加するにつれて減少することが示された⁴⁾.

我々はこれまで, 非常に高速な収束を可能とした新たなNCLMSアルゴリズムを提案してきた⁵⁾. このアルゴリズムは, NCLMSの導出時に用いるカットセットの設定に多くの自由度があることに

着目し、フィルタ出力に現れる係数の遅延を最小にするカットセットを見出し、これを用いたタイミングによって導出される。このアルゴリズムを、RCD-NCLMSアルゴリズム (Reduced Coefficient Delay NCLMS Algorithm) と呼ぶ。

本報告では、タップ数に依存しない短い出力滞在時間、高速なサンプリングレート、高速な収束特性を達成可能なパイプライン RCD-NCLMS 適応フィルタを提案する。まず、RCD-NCLMS アルゴリズムが通常の NCLMS と同等の短い出力滞在時間を有すること、そしてフィルタ部、誤差生成部、係数更新部のそれぞれが同じ演算時間で動作していることに着目し、これら 3 つの動作を並列処理させるパイプライン RCD-NCLMS を導出する。次に、導出した構成の収束特性を計算機シミュレーションにより評価し、通常の NCLMS よりも高速な収束特性を有することを示す。最後に、提案するパイプライン RCD-NCLMS 適応フィルタに対し VLSI 評価を行い、通常の NCLMS と同等の出力滞在時間、ハードウェア規模を有しながら、高速なサンプリングレートを有することを示す。

2. NCLMS アルゴリズム

時刻 k におけるフィルタ出力 $\tilde{y}(k)$ と誤差信号 $\tilde{e}(k)$ は次のように表される。

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

$$\tilde{e}(k) = d(k) - \tilde{y}(k) \quad (2)$$

$\mathbf{X}(k)$, $\mathbf{H}(k)$, $\tilde{\mathbf{H}}(k)$ はそれぞれ N 次の入力信号ベクトル、係数ベクトル、そして時間差を伴う係数ベクトルである。これらはそれぞれ以下のように表される。

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

$$\mathbf{H}(k) = [h_0(k), h_1(k), \dots, h_{(N-1)}(k)]^T \quad (4)$$

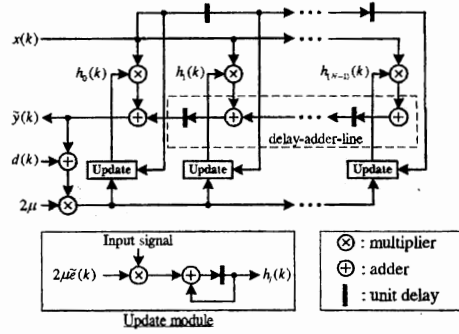


Fig. 1 Fundamental structure of NCLMS adaptive filter.

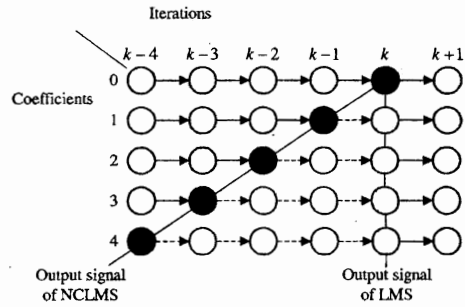


Fig. 2 State transition diagram of NCLMS adaptive filter for $N = 4$.

$$\tilde{\mathbf{H}}(k) = \mathbf{H}(k) - \boldsymbol{\gamma}(k) \quad (5)$$

$$= [\tilde{h}_0(k), \tilde{h}_1(k), \dots, \tilde{h}_{(N-1)}(k)]^T \quad (6)$$

$$= [h_0(k), \dots, h_{(N-1)}(k-N+1)]^T \quad (7)$$

ここで、 $\boldsymbol{\gamma}(k)$ は次のように表される。

$$\boldsymbol{\gamma}(k) = 2\mu \begin{bmatrix} 0 \\ \sum_{i=1}^1 \tilde{e}(k-i)x(k-1-i) \\ \sum_{i=1}^2 \tilde{e}(k-i)x(k-2-i) \\ \vdots \\ \sum_{i=1}^{N-1} \tilde{e}(k-i)x(k-N+1-i) \end{bmatrix} \quad (8)$$

NCLMS アルゴリズムの更新式は次のように表される。

$$\mathbf{H}(k+1) = \mathbf{H}(k) + 2\mu \tilde{e}(k) \mathbf{X}(k) \quad (9)$$

NCLMS-ADF の基本構成を図 1 に、出力計算における状態遷移図を図 2 に示す。図 1 に示すように、NCLMS のフィルタ出力は加算器と遅延器

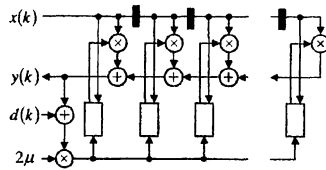


Fig. 3 Fundamental structure of LMS adaptive filter.

が縦続に接続された“ディレー・アダー・ライン”を伝播して得られる。過去の係数を用いて計算されたNCLMSのフィルタ出力には、式(8)の係数誤差に起因する出力誤差が含まれる。このため、NCLMSはLMSと比較して収束速度が大きく劣化する⁴⁾。これより、NCLMS適応アルゴリズムの収束速度を改善するためには、係数遅延を小さくする必要がある。

3. RCD-NCLMS アルゴリズムの導出

提案法をリタイミングにより導出する。対象は図3に示すLMS-ADFである。複雑さを避けるため、フィルタ部と係数更新部にわけて導出する。

3.1 フィルタ構成

従来のNCLMSとRCD-NCLMSのフィルタ部の構成を図4に示す。従来法はLMSの構成に対しタップ毎にカットセットを設定しリタイミングを行って得られた構成であり、各タップがパイプライン構成を有する。したがって、ピッチ間隔でステージを区切って(a)のように表すことができる。パイプラインのピッチ τ_p を

$$\tau_p = \tau_{add} + \tau_{multi} \quad (10)$$

とする。ただし、 τ_{add} 、 τ_{multi} はそれぞれ加算器、乗算器の演算時間である。この構成では各ステージに1タップ割り当てられており、係数遅延は0～ $N-1$ まで連続している。以下では、従来法と

同じ出力滞在時間を有し、同時に係数遅延を減少させる構成を導出する。RCD-NCLMSの基本構成を図4(b)に示す。これは、遅延のより少ないステージにより多くのタップを配置して係数遅延を減少させる構成である。そして、この構成は各ステージに多入力加算器を配置することにより可能になり、ここでは多入力加算にバイナリ・ツリー・アダー(Binary Tree Adder, BTA)を用いて実現する。BTAは、ステージ0では乗算器、ステージ1以降では τ_p と同じ演算時間を有するタップモジュールと並列に動作する。したがってBTAの加算段数は、そのステージで許容される滞在時間と加算器の出力滞在時間に依存し、これにより各ステージに配置可能なタップ数が決まる。ここで、加算器の演算時間を一定として、加算器と乗算器の出力滞在時間に次の条件を設定する。

$$m \cdot \tau_{add} \leq \tau_{multi}, \quad m = 1, 2, \dots \quad (11)$$

また、導出の際に設定するカットセットは、従来とは異なり、次の2種類を用いる。1つは、ステージに配置するタップを分離するためのカットセットであり、2つめはBTAを分割させるためのカットセットである。

以下、簡単化のため $m=2$ として、提案法を導出する。

[ステージ0, 1の導出]

ステージ0及び1を導出するカットセットリタイミングの過程を図5に示す。まず、(a)のようにステージ0とステージ1に配置するタップを分離するためにCS0とCS1を設定してリタイミングを行い、(b)の構成を得る。次に、CS0とCS1の間に配置されている加算器を(c)のようにBTAに構成を変更する。最後に、CS0-1を設定してリタイミングを行い、BTAを遷移させ(d)の構成を得る。以上の操作により、タップ0とBTAから構成されるステージ0と、タップ1～6から構成されるステージ1が導出される。

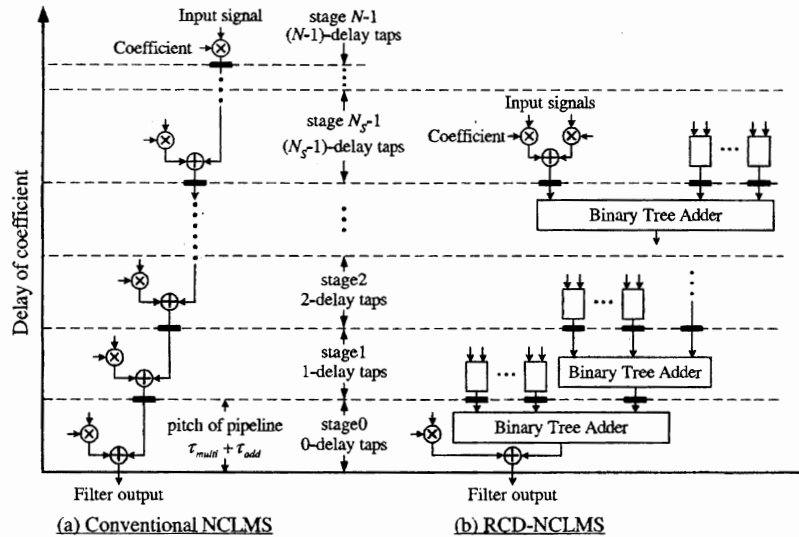


Fig. 4 Filter part of the conventional NCLMS and RCD-NCLMS adaptive filter.

CS0, CS1を設定する位置は、以下のように決定される。まず、ステージ0に配置されるタップ数は以下のように求められる。ステージ0に許容される滞在時間はフィルタ出力端から τ_p である。これを満足できるのはタップ0のみであるため、CS0はタップ0と1を分割するように設定する。次に、ステージ1に配置されるタップ数は以下のようになる。図4のように、ステージ0の乗算器と並列に動作するBTAの加算段数は m (ここでは $m=2$)であるため、BTAの入力数は4である。ディレー・アダー・ラインからの入力を考慮すると、ステージ1に配置されるタップが使用するBTA入力数は3となる。そして、これに接続されるタップモジュールは2タップで構成されているため、ステージ1には $3 \times 2 = 6$ タップが配置される。これより、CS1はこの6タップを分離するためにタップ6と7の間に設定される。最後に、BTAをステージ0に遷移させるためにCS0-1を設定してリタイミングを行う。

[ステージ2の導出]

ステージ2を導出するカットセットリタイミングの過程を図6に示す。これはステージ0, 1と同様に、ステージ2に配置するタップを分離する

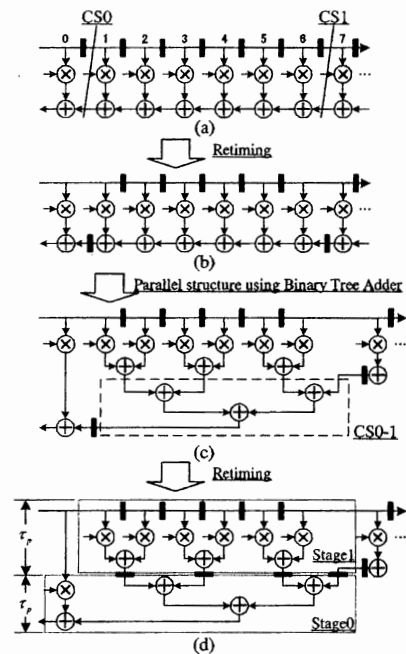


Fig. 5 Deriving process of stage 0 and stage 1.

ためのCS2を設定してリタイミングを行い、分割したタップをBTAに構成変更する。最後に、CS1-2を設定してリタイミングを行い、BTAを遷移させる。

CS2を設定する位置は以下のように決定される。まず、ステージ1に配置されるBTAの許容時間は $\tau_p = 3\tau_{add}$ であるため、段数は3段、入力数は

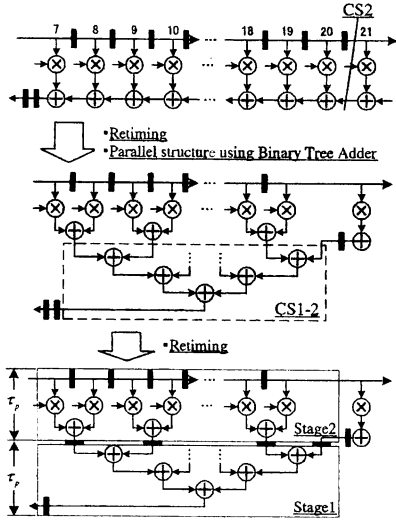


Fig. 6 Deriving process of stage 2.

8となる。ディレー・アダー・ラインからの入力を考慮すると、ステージ2に配置されるタップが使用できるBTA入力数は7となる。そして、これに接続されるタップモジュールは2タップで構成されているため、ステージ2には $7 \times 2 = 14$ タップが配置される。これより、CS2はこの14タップを分離するためにタップ20と21の間に設定される。最後に、BTAをステージ1に遷移させるためにCS1-2を設定してリタイミングを行う。

ステージ3以降の導出については、ステージ2と同様の操作を行うことにより、一様に構成することができる。最終ステージではディレー・アダー・ラインによる入力を考慮する必要はないため、BTAに接続されるタップ数の最大値は16となる。

以上の操作により、提案法のフィルタ出力部が導出される。そして、1つの遅延量に割り当てられるタップ数は乗算器と加算器の演算時間に依存しており、乗算器に対して相対的に加算器を高速化できれば、より多くのタップを割り当てることができる。

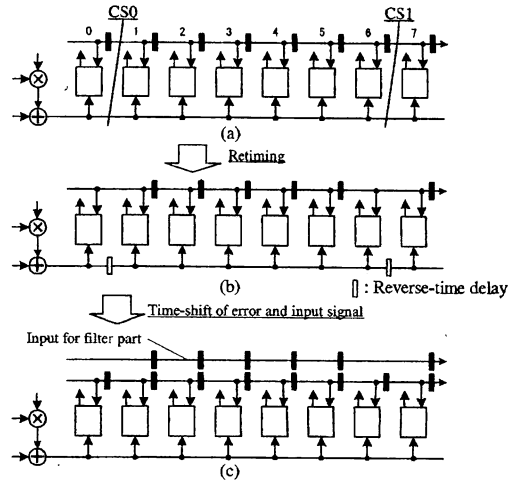


Fig. 7 Retiming of the update part by the CS0 and CS1.

3.2 係数更新部の構成

係数更新部の構成は、フィルタ出力部を生成するために設定したカットセットを用いて導出する。ここで、BTAを遷移させるために用いたカットセットCS0-1, CS1-2, ... は、閉ループで設定しているため係数更新部には影響しない。よって、考慮するのはステージを分割するために設定したCS0, CS1, ... である。CS0とCS1によるカットセットリタイミングの過程を図7に示す。まず、(a)のようにCS0とCS1に対してリタイミングを行い、(b)を得る。(b)の構成は、誤差フィードバックライン上に“逆時間の遅延器”が挿入されるが、これは因果律を満たさないため実現不可能である。そこで、従来のNCLMSの導出と同様に、誤差信号と入力信号の時間差を保ちながら時間のシフトを行うことにより、(c)の構成を得る。

CS2以降についても同様の操作を行うことにより、係数更新部の構成を得ることができる。

3.3 アルゴリズム

導出した構成より、RCD-NCLMSのアルゴリズムを導出する。ステージ数を S 、ステージ S_i ($i =$

0, 1, ..., S-1) に配置されるタップ数を N_i とする。なお,

$$N = \sum_{i=0}^{S-1} N_i \quad (12)$$

である。RCD-NCLMS の出力は次式で表される。

$$\hat{y}(k) = \hat{\mathbf{H}}^T(k) \mathbf{X}(k) \quad (13)$$

ここで, $\hat{\mathbf{H}}(k)$ は時間差を伴う係数ベクトルであり, 次のように表される。

$$\begin{aligned} \hat{\mathbf{H}}(k) = [& \\ & h_0(k), \\ & h_1(k-1), \dots, h_{(N_1)}(k-1), \\ & h_{(N_1+1)}(k-2), \dots, h_{(N_1+N_2)}(k-2), \\ & \dots, \\ & h_{(N-1-N_{(S-1)})}(k-S+1), \dots, \\ & h_{(N-1)}(k-S+1)]^T \quad (14) \end{aligned}$$

また, 入力ベクトル $\mathbf{X}(k)$ は次式である。

$$\mathbf{X}(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T \quad (15)$$

そして, ステージ S_i に配置されるタップ数 N_i は,

$$N_i = \begin{cases} 1 & i=0 \\ 2^{(m+1)} - 2 & i=1 \\ 2^{(m+2)} - 2 & i=2, \dots, S-2 \\ N - N_p & i=S-1 \end{cases} \quad (16)$$

である。ここで,

$$N_p = \sum_{j=0}^{S-2} N_j \quad (17)$$

とおいた。また, 係数更新式は次のようになる。

$$\mathbf{H}(k+1) = \mathbf{H}(k) + 2\mu\hat{e}(k) \mathbf{X}(k) \quad (18)$$

$$\hat{e}(k) = d(k) - \hat{y}(k) \quad (19)$$

3.4 パイプラインRCD-NCLMSアルゴリズム

次にパイプラインを実現して高速化を図った構成である提案するパイプラインRCD-NCLMSを導

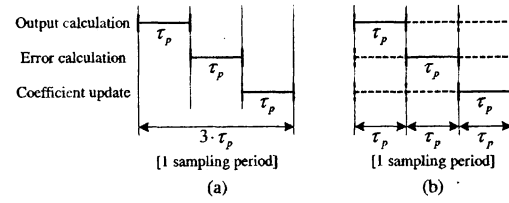


Fig. 8 Comparison of timing chart of NCLMS and pipelined RCD-NCLMS.

(a)Conventional NCLMS, (b)Proposed pipelined RCD-NCLMS.

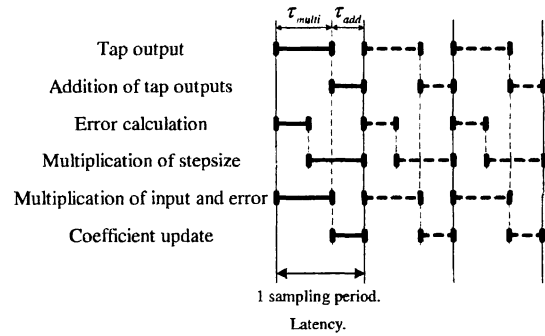


Fig. 9 Timing chart of proposed pipelined RCD-NCLMS.

出する。RCD-NCLMS の出力滞在時間は, LMS とは異なりタップ数に依存せず一定の値 $\tau_p (= \tau_{add} + \tau_{multi})$ で求められる。また, 図 8 (a) に示すようにフィルタ出力部, 誤差生成部, 係数更新部ともに τ_p である。これらのことに着目し, フィルタ部と誤差生成部の間, 及び誤差生成部と係数更新部の間にそれぞれ遅延器を挿入してパイプライン化を実現する。これにより 図 8 (b) に示すように出力計算, 誤差信号と並列に係数更新を実行することが可能になる。つまり, 1 サンプルを τ_p の間隔で演算が可能であり, 出力滞在時間を増加させずに従来の 3 倍の速度でフィルタリングを実行することが可能になる。図 9 に提案するパイプライン RCD-NCLMS のタイミングチャートの詳細を示す。

そして, 時刻 k の入力に対応する係数更新は 2 サンプル時間後に行われるため, この処理遅延を

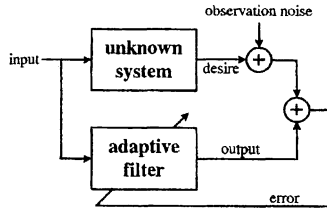


Fig. 10 Simulation model.

考慮すると係数更新式は以下のように変更される.

$$\mathbf{H}(k+1) = \mathbf{H}(k) + 2\mu\hat{e}(k-2)\mathbf{X}(k-2) \quad (20)$$

4. 収束特性

提案するパイプライン RCD-NCLMS アルゴリズムの収束特性をシミュレーションにより評価する. シミュレーションモデルは 図 10 に示すシステム同定問題である. 未知システムは低域通過 FIR フィルタを用いた. 入力信号は平均 0.0, 分散 1.0 の白色ガウス信号を用い, 未知システムの入力には観測雑音として -70.0 [dB] の入力とは無相関な白色雑音を加えた. ステップサイズパラメータはそれぞれ -66.0 [dB] の MSE を与える値を設定した. まず, 通常の NCLMS, RCD-NCLMS の LMS に対する収束速度比を表 1 に示す⁵⁾. これより, RCD-NCLMS は NCLMS よりも高速な収束特性を示しており, m やタップ数が増加するにしたがって収束速度が向上する.

次に, 本提案のパイプライン RCD-NCLMS のシミュレーション結果を図 11 に示す. なお, ここではタップ数を 128, $m=3$ とし, 比較対象に通常の LMS, NCLMS アルゴリズムを用いた. これより, 提案法は NCLMS よりも収束速度が飛躍的に向上し, LMS に近い収束特性を持つことがわかる. また, パイプライン化による影響も小さく, RCD-NCLMS アルゴリズムの約 96% と高速な収束特性を保っている. 多くのシミュレーション計算を行い, m の値やタップ数変動に対してもパイ

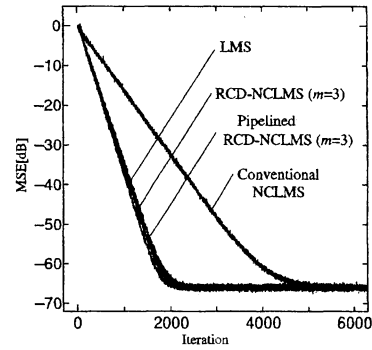


Fig. 11 Convergence characteristics for 128taps.

Table 1 Comparison of the convergence speed against the LMS⁵⁾.

Taps	Ratio of the LMS to NCLMSes[%]			
	RCD-NCLMS			Conventional NCLMS
	$m=2$	$m=3$	$m=4$	
32	77.1	77.1	77.4	42.5
64	81.4	82.6	86.0	39.2
128	86.2	87.8	94.2	41.2
256	86.6	92.6	95.5	39.5
512	92.4	93.1	95.1	41.4
1024	91.7	95.8	97.9	41.7

プライン化による影響は数 % 程度と微小な値であることを確認している.

5. VLSI評価

提案するパイプライン RCD-NCLMS-ADF を VLSI 設計システム PARTHENON により評価する. セルライブラリの設計ルールは 0.6 μ m CMOS スタンダードセル(VLSIテクノロジー社)であり, 電源電圧は 5.0[V] である. 演算に用いるデータ形式は, 2 の補数表現による 16 ビット固定小数点形であり, タップ数は 64, 128 タップとした. 使用した加算器は CLA 加算器, 乗算器は [Boothアルゴリズム]+[Wallace tree]+[CLAA] の組み合わせを用いた. 比較対象は通常の NCLMS-ADF と 3 種類の LMS-ADF のパイプラインアーキテクチャを用いた⁶⁾⁷⁾. その結果を表 2, 表 3 に示す. これより, 本提案法の出力滞在時間, サンプリングレート

Table 2 VLSI evaluation for 64-taps.

	Conventional NCLMS	Arc1 ⁽⁶⁾	Arc2 ⁽⁶⁾	Arc3 ⁽⁷⁾	Proposed pipelined RCD-NCLMS	
					m=2	m=3
Machine cycle[ns]	171	113	82	79	60	60
Sampling rate[MHz]	5.85	8.85	12.20	12.66	16.67	16.67
Latency[ns]	57	113	164	237	60	60
Power dissipation[W]	2.50	8.55	14.21	5.55	7.06	7.06
Area[mm ²]	55.50	124.90	151.42	57.05	54.87	54.92
Number of gates	506,075	1,136,741	1,375,513	519,476	500,162	500,413

Table 3 VLSI evaluations for 128-taps.

	Conventional NCLMS	Arc1 ⁽⁶⁾	Arc2 ⁽⁶⁾	Arc3 ⁽⁷⁾	Proposed pipelined RCD-NCLMS	
					m=2	m=3
Machine cycle[ns]	171	113	82	79	60	60
Sampling rate[MHz]	5.85	8.85	12.20	12.66	16.67	16.67
Latency[ns]	57	113	164	316	60	60
Power dissipation[W]	5.00	17.09	28.41	10.81	14.08	14.10
Area[mm ²]	110.13	249.74	302.78	111.23	109.52	109.74
Number of gates	1,010,779	2,272,869	2,750,521	1,012,708	998,059	1,000,004

はタップ数に全く依存せず、通常の NCLMS-ADF と同等のハードウェア規模と出力滞在時間で 3 倍の高速なサンプリングレートを実現できていることがわかる。Arc1, Arc2⁽⁶⁾は、DLMS アルゴリズムに基づくパイプライン構成に、補正項を導入した構成である。本提案法は Arc1, Arc2 の約 1/2 のハードウェア量、消費電力で、約 2 倍の高速なサンプリングレートを達成している。Arc3⁽⁷⁾は、LMS-ADF の構成をパイプライン化したものであるが、タップ数の増加に伴って出力滞在時間が増加していく。また、サンプリングレートは本提案法の方が高い値を示している。

表 4 に消費電力をサンプリングレートで正規化した表を示す。これより、本提案法の 1 [MHz] 当たりの電力効率は通常の NCLMS と同等であり、Arc1 の約 44 %、Arc2 の約 36 % で演算を行うことが可能である。

以上から、提案するパイプライン RCD-NCLMS-ADF は良好な収束特性、高速なサンプリングレート、短い出力滞在時間、消費電力効率、ハードウェア効率など要求される多くの性能を同時に満たすことが可能である。

Table 4 Normalization of power dissipation to sampling rate[W/MHz].

Number of taps	64	128
Conventional NCLMS	0.4274	0.8547
Arc1	0.9661	1.9311
Arc2	1.1648	2.3287
Arc3	0.4384	0.8539
Proposed pipelined RCD-NCLMS (m=2)	0.4235	0.8446

6. まとめ

本報告では、NCLMS アルゴリズムの収束特性を改善させた RCD-NCLMS アルゴリズムに基づいた適応フィルタの高性能パイプラインアーキテクチャを提案した。本提案法は通常の NCLMS アルゴリズムと比較して大幅な収束特性の改善を達成し、良好な収束特性を有することを計算機シミュレーションにより明らかにした。次いで、提案するパイプライン RCD-NCLMS 適応フィルタの VLSI 評価を行った。その結果、通常の NCLMS 適応フィルタと比較してハードウェア量の増加なしに約 3 倍の高速なサンプリングレート、タップ

数に依存しない非常に短い出力滞在時間を有することを明らかにした。

今後は、RCD-NCLMS アルゴリズムの収束特性に関する詳細な解析などを行う予定である。

参考文献

- 1) R.W.Stewart, J.J.Soraghan, T.S.Durrani, "Non-canonical FIR Filters and Adaptive Signal Processing," ELECTRONICS LETTERS 16th March 1989, Vol.25 No.6, pp.414-415.
- 2) W.S.Gan, J.J.Soraghan, R.W.Stewart, T.S.Durrani, "The Non-canonical LMS algorithm(NCLMS):Characteristics and Analysis," IEEE International Conference Acoustics, Speech and Signal Processing, Vol.3, pp.2137-2140, 1991.
- 3) W.S.Gan, J.J.Soraghan, R.W.Stewart, T.S.Durrani, "Comparative Analysis between Noncanonical LMS and LMS Adaptive Filtering, " ELECTRONICS LETTERS 23rd May 1991, Vol.27 No.11, pp.947-950.
- 4) K.Takahashi, Y.Tsunekawa, "Analysis of the Convergence Condition of Non-canonical LMS Adaptive Algorithm, " Proceedings of the ITC-CSCC2004, 6B3L4-1, Matsushima, Japan, 2004-07.
- 5) 高橋強, 菅野大輔, 恒川佳隆, "NCLMS適応アルゴリズムの収束速度改善," SIP研究会, SIP103(2004-103), 2005-01-20.
- 6) Akio Harada, Kiyoshi Nishikawa, Hitoshi Kiya, "Pipelined Architecture of the LMS Adaptive Digital Filter with the Minimum Output Latency", IEICE Trans.Fundamentals, pp.1578-1585, Vol.E81-A No.8, 1998
- 7) Tadaaki kimijima, Kiyoshi Nishikawa, Hitoshi Kiya, "An Effective Architecture of the Pipelined LMS Adaptive Filters", IEICE Trans.Fundamentals, pp.1428-1434, Vol.E82-a No.8, 1999