

ルックアヘッド Delayed LMS アルゴリズムに基づくパイプライン適応フィルタ

松原 勝重[†] 西川 清史[†] 貴家 仁志[†]

Pipelined Adaptive Filters Based on Look-Ahead-Based Delayed LMS Algorithm

Katsushige MATSUBARA[†], Kiyoshi NISHIKAWA[†], and Hitoshi KIYA[†]

あらまし パイプライン化が可能な適応アルゴリズムとして遅延最小平均 2 乗 (DLMS: delayed least mean square) アルゴリズムが知られている。従来の DLMS に基づくパイプライン適応フィルタは、動作クロックレートと収束特性がトレードオフの関係にあり、これらを同時に改善できない。本論文では、DLMS を新しいルックアヘッド変換に基づき拡張したルックアヘッド DLMS (LDLMS: look-ahead-based DLMS) アルゴリズムと、そのパイプライン処理構成を提案している。提案法により、適応フィルタは収束特性を劣化させることなく高い動作クロックレートでパイプライン処理が可能となることを示している。提案法は、収束特性に影響を与える遅延量 (実効遅延量) を、所望の動作クロックレートの下で最小化できる特徴をもつ。また計算機シミュレーションにより提案法の有効性を確認している。

キーワード パイプライン処理, DLMS アルゴリズム, 新しいルックアヘッド変換, LDLMS アルゴリズム

1. ま え が き

本論文では、最小平均 2 乗 (LMS: least mean square) アルゴリズム [1] に基づくパイプライン処理を可能とするアルゴリズムと、そのハードウェア構成 (アーキテクチャ) を提案する。提案法により、従来法では不可能である、良好な収束特性と高クロックレートでの動作を同時に満足できる、適応フィルタ (ADF: adaptive digital filter) のパイプライン処理が可能となる。

近年、高品位な信号処理への要求が高まり ADF は多量の信号を高速に処理することが要求されている。高速な処理を効率良く実現するための一手法としてパイプライン処理が知られている。従来の ADF のパイプライン処理は、再帰最小 2 乗 (RLS) アルゴリズムに基づく実現を中心として研究が行われてきた [2],[3]。しかしながら、RLS に基づく実現では多くの乗算や、除算・平方根といった複雑な演算を必要とし、膨大な量のハードウェアが必要となる。このた

め、構造がより単純な LMS に基づくパイプライン実現法の研究が進められており、いくつかのアルゴリズムやアーキテクチャが提案されている [4]~[10]。現在までに LMS のパイプライン処理実現を可能とするアルゴリズムとして relaxed look-ahead (近似ルックアヘッド) pipelined LMS (PIPLMS) アルゴリズム [4] と、遅延 LMS (DLMS: delayed LMS) アルゴリズム [5],[6] の二つが提案されている。このうち DLMS は、PIPLMS の一特殊形とみなせることが示されているが [4]、PIPLMS より収束特性が優れている。このため高速な収束特性が要求される分野において、DLMS の応用が期待されている。DLMS は LMS の係数更新式に遅延 (ディレー) を挿入することでパイプライン処理が可能とする。このとき挿入するディレーを大きくすることで、より高速な処理が可能となる。しかし反面、大きなディレーは収束特性を劣化させてしまうことが知られている [5]。

従来報告されている DLMS に基づくパイプライン処理のアーキテクチャ [7]~[10] は、高速処理あるいは収束特性のどちらか一方を犠牲にして実現している。すなわち文献 [7],[8] のアーキテクチャ (Arc 1) では

[†] 東京都立大学工学部電子情報工学科, 八王子市 Faculty of Technology, Tokyo Metropolitan University, Hachioji-shi, 192-03 Japan

ADF の 1 タップを 1 機能ユニット (セル) としてパイプライン処理を行っている。このことで ADF の動作クロックレートをタップ数と独立に高い値を保つことを可能とする。しかし、この Arc 1 では、挿入されるディレーが ADF のタップ数と等しくなり、高次のフィルタを必要とする応用において収束特性が劣化する問題が生じる。

これに対し我々は、文献 [9], [10] でこの問題を回避できる DLMS の新しいアーキテクチャ (Arc 2) を示した。Arc 2 では複数のタップにより 1 セルを構成することで、必要となるディレーを減少でき、収束特性の改善を可能としている。しかし Arc 2 では収束特性の改善は可能であるが、Arc 1 に比べ ADF の動作クロックレートが低下してしまう。実際の応用では、動作クロックレートを低下させず、かつ優れた収束特性を実現する ADF が要求される。

本論文では、Arc 1 と同等の動作クロックレートでのパイプライン処理を可能とし、かつ Arc 2 より優れた収束特性を実現できるパイプライン ADF を提案する。提案する ADF は、DLMS を新しいルックahead変換 [11], [12] に基づき拡張したルックahead DLMS (LDLMS: look-ahead-based DLMS) アルゴリズム [13], [14] に基づいている。LDLMS は DLMS に補正項を付加することにより、収束特性に影響を与えるディレーの量 (実効遅延量) を等価的に減少させる。このことで収束特性の改善を可能とする。LDLMS のシグナルフローグラフを検討することで、LDLMS をパイプライン処理するためのアーキテクチャを導出する。このアーキテクチャを用いることで、実効遅延量をフィルタのタップ数とは独立な最小の値にすることができる。この特徴により、提案アーキテクチャを用いることで ADF は優れた収束特性をもち、かつ高クロックレートでのパイプライン処理が可能となることを示す。

本論文の構成は次の通りである。2. で、新しいルックahead変換と、この変換に基づき DLMS を拡張した LDLMS アルゴリズムを提案する。3. では、LDLMS のシグナルフローグラフを導出し、4. において、LDLMS のパイプライン処理アーキテクチャを示す。最後に 5. でむすぶ。

2. パイプライン LMS アルゴリズム

ここでは DLMS について述べ、その問題点を明らかにする。次にこの問題を改善可能なアルゴリズムと

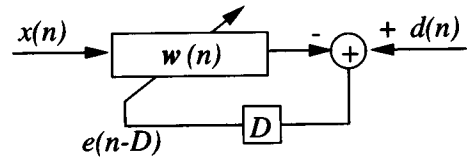


図1 DLMS アルゴリズムのブロック図
Fig.1 Block diagram of DLMS algorithm.

して LDLMS を提案し、その有効性をシミュレーションを通じ示す。

2.1 DLMS アルゴリズム

DLMS は、図 1 に示すように LMS の誤差ループにディレー D を挿入することで得られる。パイプライン処理は各タップ間にディレー D を再配置する (リタイミング [15]) ことで達成できる。

DLMS は以下の式で表せる [5]。

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n-D)\mathbf{x}(n-D) \quad (1)$$

$$e(n-D) = d(n-D) - \mathbf{x}^T(n-D)\mathbf{w}(n-D) \quad (2)$$

ここで $\mathbf{w}(n)$ と $\mathbf{x}(n)$ はそれぞれ、適応フィルタのタップ係数ベクトル、入力信号ベクトルで、

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T \quad (3)$$

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (4)$$

である。また $d(n)$ は所望信号、 μ はステップサイズパラメータ、 N はフィルタのタップ数、 $[\cdot]^T$ は行列の転置を表す。式 (1) の D は挿入されるディレー量を表す。挿入するディレー量 D を大きくすることで、より細かいパイプライン処理が可能となる反面、平均 2 乗誤差 (MSE: Mean Squared Error) での収束を保証する μ の上限が減少することが知られている [5], [6]。すなわち D の増大により、ADF を高クロックレートで動作させることが可能となるが、同時に収束特性の劣化を引き起こす。DLMS [5] において、 D はフィルタのタップ数 N に等しくされる。そのため N が増大すると D も増大してしまい、高次のフィルタを必要とする応用では収束特性が劣化してしまう。

2.2 新しいルックahead変換

ディレー D を一定のまま収束特性を改善することを考えよう。式 (2) を式 (1) に代入すると

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu d(n-D)\mathbf{x}(n-D) - \mu \mathbf{x}(n-D)\mathbf{x}^T(n-D)\mathbf{w}(n-D) \quad (5)$$

となる。式 (5) の右辺第 1 項の $w(n)$ と第 3 項の $w(n-D)$ の時刻差 D が、DLMS における収束特性の劣化を引き起こす。すなわち収束特性を改善するためには、式 (2) の $w(n-D)$ に含まれる D の値を減少させる必要がある。しかし、ADF を高いクロックレートで動作させるためには D を減少させることはできない。そのため、本論文では D を等価的に減少させることを考える。

いま、 $w(n-D)$ のかわりに $w(n-(D-\delta))$ を用いて計算された誤差 $\varepsilon_\delta(n-D)$ を

$$\varepsilon_\delta(n-D) = d(n-D) - \mathbf{x}^T(n-D)\mathbf{w}(n-(D-\delta)) \quad (6)$$

と定義する。なお δ は $0 \leq \delta \leq D$ の範囲をとる整数である。式 (2) の $e(n-D)$ のかわりに $\varepsilon_\delta(n-D)$ を用いて、フィルタ係数を更新することが可能であれば、 D の影響を減少可能となることがわかる。以下では、 $\varepsilon_\delta(n-D)$ の $e(n-D)$ による表現を導出する。その結果 $\varepsilon_\delta(n-D)$ にもとづく適応アルゴリズムとして、LDLMS アルゴリズムを提案する。

まず、 $\varepsilon_\delta(n-D)$ と $e(n-D)$ との関係を求めよう。そのために $w(n-D)$ を過去の δ 個の $e(n)\mathbf{x}(n)$ で展開することを考える [11],[12]。すなわち、式 (1) より $w(n-D)$ を

$$\begin{aligned} w(n-D) &= w(n-D_{eff}) \\ &\quad - \sum_{i=0}^{\delta-1} \mu e(n-2D+i)\mathbf{x}(n-2D+i) \end{aligned} \quad (7)$$

と表す。ここで D_{eff} は

$$D_{eff} = D - \delta \quad (8)$$

と定義する。式 (7) の $w(n-D_{eff})$ を用いて式 (2) は

$$e(n-D) = \varepsilon_\delta(n-D) + \Lambda_\delta(n) \quad (9)$$

と変換される。ここで $\Lambda_\delta(n)$ は

$$\begin{aligned} \Lambda_\delta(n) &= \sum_{i=0}^{\delta-1} \mu e(n-2D+i) \\ &\quad \cdot \mathbf{x}^T(n-D)\mathbf{x}(n-2D+i) \end{aligned} \quad (10)$$

である。すなわち、式 (9) より、 $\varepsilon_\delta(n-D)$ は

$$\varepsilon_\delta(n-D) = e(n-D) - \Lambda_\delta(n) \quad (11)$$

と等価的に表現できることがわかる。

式 (2)、(9)~(11) より、誤差信号 $e(n-D)$ に $-\Lambda_\delta(n)$ を付加することで $\varepsilon_\delta(n-D)$ が求まり、その結果ディレイを等価的に D から $D_{eff}(=D-\delta)$ に減少可能となることがわかる。 D_{eff} は、実際に収束特性に影響を与えるディレイ量とみなすことができ、以下では実効遅延量とよぶ。式 (7)~(10) で表せる DLMS を誤差ループにおいて展開する手法を、新しいルックアヘッド変換とよぶ [11],[12]。

2.3 LDLMS アルゴリズム

前節で導出した新しいルックアヘッド変換に基づき DLMS を拡張した LDLMS アルゴリズムを提案する [13],[14]。LDLMS は以下の式で表せる。

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mu \varepsilon_\delta(n-D)\mathbf{x}(n-D) \quad (12) \\ \varepsilon_\delta(n-D) &= d(n-D) \\ &\quad - \mathbf{x}^T(n-D)\mathbf{w}(n-D) - \Lambda_\delta(n) \end{aligned} \quad (13)$$

ここで δ はアルゴリズムの収束特性を変化させるパラメータとみなすことができ

$$0 \leq \delta \leq D \quad (14)$$

の範囲をとる。LDLMS の収束特性は、 $\delta=0$ のとき DLMS の収束特性と一致し、 $\delta=D$ のとき LMS の収束特性に一致する。 δ の値を調整することで、実効遅延量 D_{eff} を調整可能となる。この特徴により LDLMS に基づく ADF は、高い動作クロックレートを保ったまま収束特性の改善が可能となる。ところで $\delta=D$ のときの LMS 等価アルゴリズムは文献 [16] で報告されており、これは LDLMS において $\delta=D$ の特別な場合とみなせる。図 2 に LDLMS のブロック図とその等価構成を示す。

2.4 シミュレーション

ここでは LDLMS の有効性をシミュレーションを通じ示す。シミュレーションとして、20 タップの低域フィルタを未知システムとするシステム同定を行った。条件として適応フィルタのタップ数 $N=20$ 、ディレイ $D=20$ とした。この条件下で $\delta=0, 12, 17, 20$ それぞれの値でシミュレーションを行った。ここで $\delta=0$ は DLMS に、 $\delta=20$ は LMS に相当する。またステップサイズパラメータ μ は、MSE での収束を保証する範囲で最大の値を用いた [6]。図 3 に結果を示す。結果は独立な 10 回の試行の集合平均をとったものである。

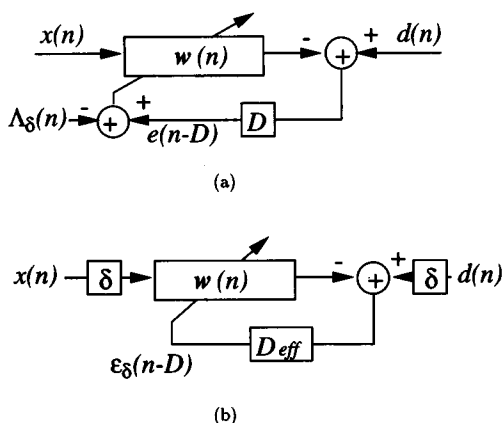


図2 LDLMS アルゴリズム
(a) ブロック図, (b) 等価構成図

Fig.2 LDLMS algorithm. (a) and (b) show block diagram and equivalent structure, respectively.

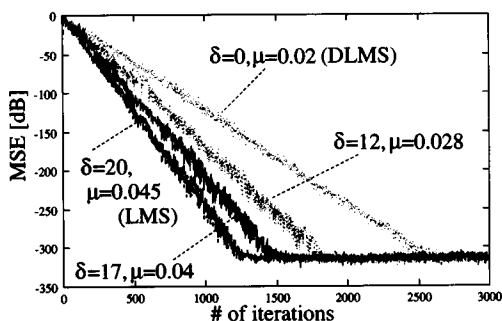


図3 異なる δ における LDLMS の収束特性

Fig.3 Convergence characteristics of LDLMS with various values for δ .

図3より δ を大きくすることで、より大きなステップサイズが選択でき収束特性が改善できることがわかる。これより LDLMS に基づくパイプライン ADF は、ディレーとは独立に収束特性を改善可能なことがわかり、LDLMS の有効性が確認できる。

3. LDLMS のシグナルフローグラフ表現

本章では LDLMS のハードウェア構成を表すために、LDLMS のシグナルフローグラフ (SFG: Signal Flow Graph) 表現を考える。ここでは最大の処理時間 (クリティカルパス) が最小となる SFG を求めることとする。なぜなら同じ条件でパイプライン処理を行った場合、元となる SFG のクリティカルパスが小さいほど動作クロックレートを高くできるためである。

LDLMS において、式 (10) の $\Lambda_\delta(n)$ はフィルタ係

数 $w(n)$ には直接依存していない。このため $\Lambda_\delta(n)$ の演算と $w(n)$ の更新は、それぞれ独立に行うことができ、 $w(n)$ の更新には従来の DLMS に基づく構成法 [7]~[9] が適用可能である。よって以下では式 (10) の $\Lambda_\delta(n)$ の SFG 表現のみ考える。

3.1 依存グラフ表現

一つのアルゴリズムを実現する SFG は複数存在する [17]。そのためハードウェアを構成するとき、目的に応じて適切な SFG を選ぶ必要がある。そのため、ここではまず LDLMS アルゴリズムにおける各演算の依存関係を表した依存グラフ (DG: Dependence Graph) 表現を考える。

まず式 (10) を変形し、

$$\Lambda_\delta(n) = \sum_{i=0}^{\delta-1} \mu e(n-2D+i) X_i(n) \quad (15)$$

$$X_i(n) = \sum_{k=0}^{N-1} \hat{x}_i(n-k) \quad (16)$$

$$\hat{x}_i(n) = x(n-D)x(n-2D+i) \quad (17)$$

と表す。式 (15)~(17) の演算の依存関係を表す DG を、それぞれ図 4(a)~(c) に示す。各 DG 内にある“Hyper-plane” (HP) は、演算を同時刻に行う節点 (ノード) を結んだ直線あるいは平面を表す。この HP の選び方とグラフの射影を行う方向により、いくつかの SFG がこれらの DG より得られる [17]。

3.2 シグナルフローグラフの導出

ここでは図 4 の DG から、クリティカルパスが最小になる SFG を求める。このとき注意することは、各 DG における HP 上のノード間に演算の依存関係が存在すると、SFG に変換したときにクリティカルパスが増大することである。これはノード間に演算の依存関係があると、それらの演算を並列に処理することができず動作クロックレートの低下を引き起こしてしまうためである。よってクリティカルパスが最小となる SFG を導出するためには、演算の依存関係が最小となる HP を導出する必要がある。

まず図 4(a) の DG1 について考える。DG1 では演算の依存は斜方向 ($B_{in} - B_{out}$) のみに存在し、水平方向 ($A_{in} - A_{out}$) には存在しない。よって図 4(a) に示すように HP を $A_{in} - A_{out}$ 方向に選択することで、HP 上のすべてのノード間には演算の依存関係がなくなる。このときのクリティカルパスは、各ノードにおける乗算と加算それぞれ 1 回に要する時間となる。

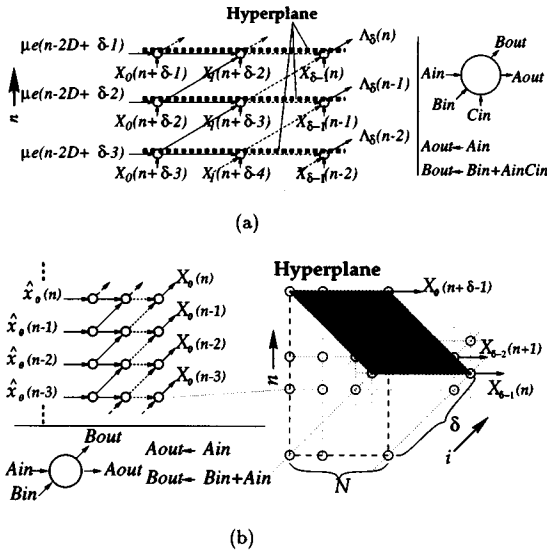


図4 $\Lambda_\delta(n)$ の依存グラフ (a) DG1: 式 (15), (b) DG2: 式 (16), (c) DG3: 式 (17)
 Fig.4 Dependence graph for $\Lambda_\delta(n)$. (a) DG1 for Eq. (15), (b) DG2 for Eq. (16), (c) DG3 for Eq. (17).

次に図 4(b) の DG2 を考える。DG1 において HP を図 4(a) のように選択したため、DG1 には時刻 n において信号列 $\{X_{\delta-1}(n), X_{\delta-2}(n+1), \dots, X_0(n+\delta-1)\}$ を入力する必要がある。DG1 の入力信号は DG2 より出力されるので、DG2 は時刻 n において上記の信号列を出力しなければならない。この信号列を出力するための HP を図 4(b) に示す。ここで DG2 は i 方向にはノード間の依存関係はなく、また水平方向には演算の依存が無いことに注意する。すなわち図 4(b) の HP 上のすべてのノード間には演算の依存関係は存在しない。またこのときのクリティカルパスは、各ノードにおける加算 1 回に要する時間となる。

最後に図 4(c) の DG3 を考える。DG2 と同様を考える と DG3 は時刻 n において信号列 $\{\hat{x}_{\delta-1}(n), \hat{x}_{\delta-2}(n+$

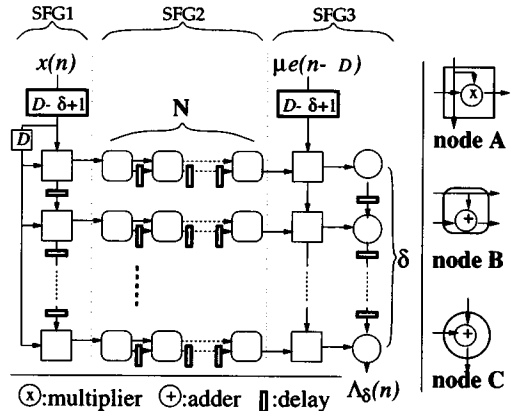


図5 $\Lambda_\delta(n)$ のシグナルフローグラフ。SFG1 は DG1 に、SFG2 は DG2 に、SFG3 は DG3 に対応している部分を表す

Fig.5 Signal flow graph of $\Lambda_\delta(n)$. SFG1, SFG2 and SFG3 are parts of the SFG of $\Lambda_\delta(n)$ which correspond to DG1, DG2 and DG3, respectively.

$1), \dots, \hat{x}_0(n+\delta-1)\}$ を出力しなければならない。この信号を出力するための HP を図 4(c) に示す。ここで DG3 はすべてのノード間で演算の依存関係が無いことに注意する。すなわち HP を任意に選んでも、クリティカルパスは各ノードにおける乗算 1 回に要する時間となる。

DG1, DG2, DG3 それぞれにおいて上記の HP を用いて n 方向へ射影した結果得られる SFG を図 5 に示す。この SFG は式 (10) の演算を実現する SFG の中で、最小のクリティカルパスをもつ。このときのクリティカルパスは乗算加算それぞれ 2 回に要する時間となる。

4. LDLMS のパイプライン処理化

ここでは 3.2 で得られた SFG を基に、LDLMS をパイプライン処理するためのアーキテクチャを提案する。提案するアーキテクチャにおけるパラメータ δ の満たすべき条件を導出する。また提案アーキテクチャの有効性を構成例を通じ示す。

4.1 提案するアーキテクチャ

提案するアーキテクチャは、図 5 の SFG 内部に存在する $D - \delta + 1$ 個のディレーをリタイミングすることにより得られる。リタイミングを行うことで δ の条件が式 (14) とは異なってくる。以下では、LDLMS のパイプライン実現のための δ に対する条件を示す。

まず 1 回の加算と 1 回の乗算の処理に必要な時間をそれぞれ $\tau_a, \alpha\tau_a$ とする。またフィルタの所望の動作

クロック周期が $\beta\tau_a$ と与えられたとし、 α と β は

$$1 \leq \alpha \leq \beta \tag{18}$$

を満たすと仮定する。ここで図 5 の SFG のクリティカルパスを t_{cp} とすると 3.2 より

$$t_{cp} = 2(1 + \alpha)\tau_a \tag{19}$$

となる。所望のクロック周期での動作を満足するためには、クリティカルパスが

$$t_{cp} \leq \beta\tau_a \tag{20}$$

を満たすよう処理を細分化する必要がある。そのため SFG 内部のディレーをリタイミングする必要がある。図 6 にリタイミングの例を示す。図 5 の SFG のパイプライン処理に必要なとなるディレー量の最大値を D_{sfg} とすると、

$$D_{sfg} = \max \left(\left\lceil \frac{2(1 + \alpha)\tau_a}{\beta\tau_a} \right\rceil \right) = 4 \tag{21}$$

となる。ここで $\max(x)$ は x の最大値を表し、 $\lceil x \rceil$ は x 以上の最小の整数を表す。式 (21) より D_{sfg} はフィルタのタップ数 N やディレー量 D に依存せず、式 (18) の仮定の下で最小の値をとる。所望のクリティカルパスでの動作を得るためには、SFG 内部の $D - \delta + 1$ 個のディレーが D_{sfg} より少なくなければならない。よって式 (14) と式 (21) より δ の満たすべき条件は

$$0 \leq \delta \leq D - 3 \tag{22}$$

となる。式 (22) は、式 (18) の仮定の下でクリティカルパスが $\beta\tau_a$ 以下のパイプライン処理を行うときの δ の条件である。式 (14) と式 (22) との比較により、パイプライン化のためには δ の範囲が狭くなることがわかる。しかし式 (8) と式 (22) より、実効遅延量 D_{eff} はフィルタのタップ数とは独立な、最小値 $D_{eff} = 3$ をとることが可能となる。このため、高次のフィルタを必要とする応用になるほど、収束特性の劣化がほとんど生じなくなる。また文献 [16] の方法では式 (22) を満たさないため、パイプライン処理を行えないことに注意する。

図 7 に $\beta = \alpha$ としたときの提案アーキテクチャの例を示す。

4.2 構成例

ここでは提案法の有効性を構成例により示す。構成の条件としてタップ数 $N = 20$ で、 $\alpha = 4$ とした。ま

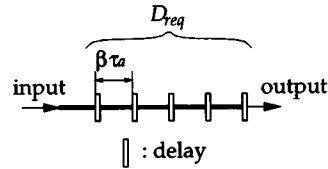


図 6 リタイミングによる処理の細分化
Fig. 6 Retiming technique for partitioning a process.

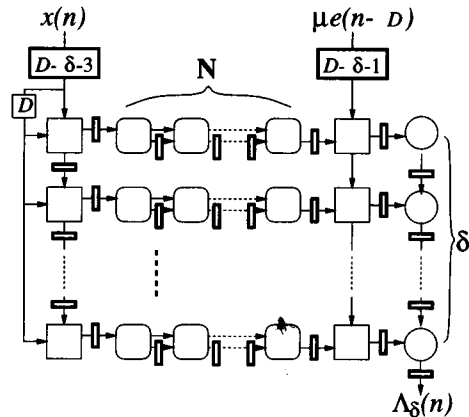


図 7 $\Lambda_\delta(n)$ のパイプライン処理アーキテクチャ
Fig. 7 Pipelined architecture of $\Lambda_\delta(n)$.

たパイプライン ADF の評価基準として、以下に定義する ADF の動作クロックレート S を用いる。

$$S = \frac{1}{t_{cp}} \tag{23}$$

ここで t_{cp} はパイプライン処理を行うときのクリティカルパスである。

図 8 に提案法と DLMS に基づく従来法 [9] の、アルゴリズムのディレーに対する動作クロックレートを示す。ここで縦軸は動作クロックレートの上限 $S \leq 1/(\alpha\tau_a) = 1/(4\tau_a)$ を 100% としたときの百分率を、横軸は収束特性を変化させるディレーを表す。但し提案法では実効遅延量 D_{eff} で、従来法では D である。

図 8 より、DLMS に基づく従来法ではディレーと動作クロックレートはトレードオフの関係にあるが、提案法では、常にクロックレートの上限で動作可能であることがわかる。すなわち、提案法に基づく ADF は収束特性を劣化させることなく、クロックレートの上限 $1/(4\tau_a)$ でパイプライン処理可能なことがわかる。

実際の応用においては、 δ と D は要求される仕様とハードウェア量を考慮にいれ式 (22) より決定され

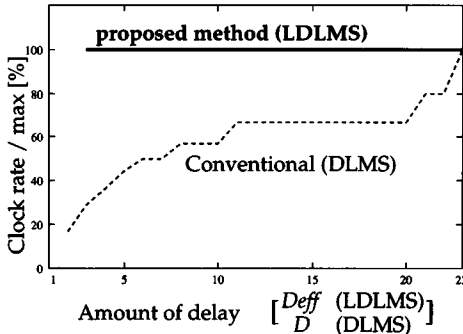


図8 ディレイ量に対する動作クロックレート特性
Fig.8 Clock rate versus the amount of delay.

る。決定された δ と D の下で、適応フィルタ部には、 $D = N$ のときは文献 [7], [8] の構成法を、 $D < N$ のときは文献 [9] の構成法を用いることで、所望の仕様を満たすパイプライン ADF のアーキテクチャを得られることがわかる。

5. むすび

本論文では、収束特性を劣化させずにパイプライン処理を可能とする LDLMS アルゴリズムと、そのハードウェアアーキテクチャを提案した。LDLMS を用いることで、収束特性に影響を与える実効遅延量を減少可能で、収束特性を改善できることを示し、シミュレーションによりその有効性を確認した。また、クリティカルパスが最小となる LDLMS の SFG を導出し、そのパイプライン処理アーキテクチャを示した。提案したアーキテクチャを用いることで、所望のクロックレートの下で、実効遅延量を最小にできることを示した。この結果、ADF は収束特性をほとんど劣化させることなく、高クロックレートでパイプライン処理を可能とすることを示した。最後に構成例を通じその有効性を確認した。

謝辞 本研究の一部は、科学研究費奨励研究 (A) 研究課題番号 08750453 の成果である。

文 献

- [1] B. Widrow and S.D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [2] K.J. Raghunath and K.K. Parhi, "High-speed RLS using scaled tangent rotation (star)," Proc. IEEE ISCAS '93, Chicago, USA, pp.1959-1962, May 1993.
- [3] K.J. Raghunath and K.K. Parhi, "A 100 MHz pipelined RLS adaptive filter," Proc. IEEE ICASSP '95, Detroit, Michigan, pp.3187-3190, May 1995.
- [4] N.R. Shanbhag and K.K. Parhi, "Relaxed look-ahead pipelined LMS adaptive filters and their application to ADPCM coder," IEEE Trans. Circuits & Syst. II, vol.40, no.12, pp.753-766, Dec. 1993.
- [5] G. Long, F. Ling, and J.G. Proakis, "The LMS algorithm with delayed coefficient adaptation," IEEE Trans. Acoust., Speech, & Signal Process., vol.37, no.9, pp.1397-1405, Sept. 1989.
- [6] G. Long, F. Ling, and J.G. Proakis, "Corrections to "the LMS algorithm with delayed coefficient adaptation"," IEEE Trans. Signal Process., vol.40, no.1, pp.230-232, Jan. 1992.
- [7] H. Herzberg, R. Haimi-Cohen, and Y. Be'ery, "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation," IEEE Trans. Signal Process., vol.40, no.11, pp.2799-2803, Nov. 1992.
- [8] M.D. Meyer and D.P. Agrawal, "A high sampling rate delayed LMS filter architecture," IEEE Trans. Circuits & Syst. II, vol.40, no.11, pp.727-729, Nov. 1993.
- [9] 松原勝重, 西川清史, 貴家仁志, "Delayed LMS アルゴリズムに基づくパイプライン適応フィルタ," 信学論 (A), vol.J79-A, no.5, pp.1050-1057, May 1996.
- [10] K. Matsubara, K. Nishikawa, and H. Kiya, "2-D pipelined adaptive filters based on 2-D delayed LMS algorithm," IEICE Trans. Fundamentals, June 1997 (To be published).
- [11] 松原勝重, 西川清史, 貴家仁志, "新しいルックアヘッド変換を用いたパイプライン適応フィルタのアーキテクチャ," 1996 信学秋季全大, vol.A-94, Sept. 1996.
- [12] 松原勝重, 西川清史, 貴家仁志, "新しいルックアヘッド変換を用いたパイプライン適応フィルタ," 第 11 回デジタル信号処理シンポジウム講演論文集, pp.469-474, Nov. 1996.
- [13] K. Matsubara, K. Nishikawa, and H. Kiya, "A new pipelined architecture of the LMS algorithm without degradation of convergence characteristics," Proc. IEEE ICASSP '97, Munich, Germany, pp.4125-4128, April 1997.
- [14] K. Matsubara, K. Nishikawa, and H. Kiya, "Pipelined LMS adaptive filter using a new look-ahead transformation," *ibid.*, pp.2309-2312.
- [15] C.E. Leiserson, F. Rose, and J. Saxe, "Optimizing synchronous circuitry by retiming," Proc. of the Third Caltech Conference on VLSI, pp.87-116, March 1983.
- [16] R.D. Poltmann, "Conversion of the delayed LMS algorithm into the LMS algorithm," IEEE Signal Proc. Letters, vol.2, no.12, p.223, Dec. 1995.
- [17] S.Y. Kung, VLSI Array Processors, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1988.

(平成 9 年 3 月 27 日受付, 5 月 28 日再受付)



松原 勝重 (学生員)

平7都立大・工・電子・情報工卒。平9同大大学院修士課程了。同年日電入社。在学中は適応信号処理に関する研究に従事。



西川 清史 (正員)

平2都立大・工・電気工卒。平4同大大学院修士課程了。同年新日本製鉄エレクトロニクス研究所勤務。平5都立大工学部電子・情報工学科助手。現在に至る。工博。適応信号処理および情報源符号化に関する研究に従事。IEEE 会員。



貴家 仁志 (正員)

昭55長岡技科大・工・電気電子システム卒。昭57同大大学院修士課程了。同年東京都立大工学部電気工学科助手。現在、同大電子・情報工学科助教授。工博。1995年シドニー大(オーストラリア)客員研究員。マルチレート信号処理、画像処理および信号処理アルゴリズムのVLSI実現に関する研究に従事。著書「高速フーリエ変換とその応用」、「デジタル信号処理」、「マルチレート信号処理」(昭晃堂)、「よくわかるデジタル画像処理」(CQ出版)。電子画像学会、映像情報メディア学会、IEEE各会員。