

適応 Volterra フィルタの選択学習法

A Selective Learning Algorithm for Adaptive Volterra Filters with Limited Number of Terms

山本 裕之 中山 謙二 平野 晃宏
Hiroyuki Yamamoto Kenji Nakayama Akihiro Hirano

金沢大学大学院 自然科学研究科 電子情報システム専攻
Dept. of Electrical and Computer Eng., Faculty of Eng., Kanazawa Univ.
E-mail: yamamoto@leo.ec.t.kanazawa-u.ac.jp

あらまし

本稿では、適応 Volterra フィルタの収束性の向上を目的として、その学習において、Volterra 関数の有効な項を選択学習する方法を提案する。有効な項は遺伝的アルゴリズムをベースとして、該当するフィルタ係数の大きさを考慮して選択される。また、フィルタ係数を次世代に継承することにより一層の収束性向上を図る。白色及び有色の入力信号を用いたシミュレーションにより、計算量は増加するが収束速度が向上することが確認できた。

ABSTRACT

This paper proposes a selective learning method for adaptive Volterra filters, in which useful terms are selected and their coefficients are trained. Useful terms are selected based on the genetic algorithm, and magnitude of the corresponding filter coefficients are used to determine order of selection. Furthermore, the filter coefficients are succeeded to the next generation to improve convergence more. Computer simulation using white and colored input signals show that even though the proposed method requires more computation, fast convergence can be achieved.

1 まえがき

LSI 技術の発達と共に DSP の性能も向上し、適応フィルタが通信、信号処理、制御等の広い分野で応用されている。高い情報品質が要求されるようになってきた背景から、マイクやスピーカ、回線特性の非線形性への対応も必要になっている [1][2]。

非線形適応フィルタの構成法として Volterra 関数を用いる方法は非線形特性のより一般的な表現が可能であるという特徴を有している。しかし、フィルタ次数の増加にともない、Volterra フィルタの項の数は膨大になるため、フィルタ規模、収束性、計算量において問題となる。特にその影響は入力が有色信号になるほど顕著に現れる。そこで近年、有色信号を白色化して収束性を向上させる [3] など、収束性の向上をはかる研究が盛んに行なわれている [4][5][6]。

一方、実際の応用では Volterra 関数のすべての項が用いられることは少ないと考えられている。本稿では、学習時に Volterra フィルタのすべての項を使用するのではなく、有効な項のみを選択し、その係数を学習することにより収束特性を向上させることを目的として、遺伝的アルゴリズム (GA) をベースとした項数制限アルゴリズム [7] を適用し、さらに、フィルタ係数の次世代への継承により、収束特性のさらなる向上を図る。最後に白色および有色の入力信号を用いたシミュレーションによりその有効性を確認する。

2 適応 Volterra フィルタ (AVF)

2.1 Volterra フィルタ

2 次 Volterra フィルタは次式で表される。ただし、本稿ではすべての Volterra 核が有限の記憶長 N をもつものとしているので、定数項は考えない。

$$\begin{aligned} y(k) &= \sum_{l_1=0}^{N-1} h_1(l_1)x(k-l_1) \\ &+ \sum_{l_1=0}^{N-1} \sum_{l_2=0}^{N-1} h_2(l_1, l_2)x(k-l_1)x(k-l_2) \quad (1) \\ &= \mathbf{H}_1^T \mathbf{x}(k) + \mathbf{x}^T(k) \mathbf{H}_2 \mathbf{x}(k) \quad (2) \end{aligned}$$

ただし

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \cdots \ x(k-N+1)]$$

$$\mathbf{H}_1 = [h_1(0) \ h_1(1) \ \cdots \ h_1(N-1)]$$

$$\mathbf{H}_2 = \begin{bmatrix} h_2(0,0) & \cdots & h_2(0,N-1) \\ h_2(1,0) & \cdots & h_2(1,N-1) \\ \vdots & \ddots & \vdots \\ h_2(N-1,0) & \cdots & h_2(N-1,N-1) \end{bmatrix}$$

であり、 $\mathbf{H}_2^T = \mathbf{H}_2$ である。

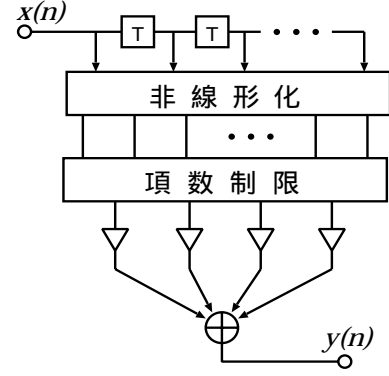


図 2: フィルタ構造

2.2 適応 Volterra フィルタ

図 1 に適応 Volterra フィルタ (AVF) を用いたシステム同定のブロック図を示す。 $v(k)$ は付加雑音である。

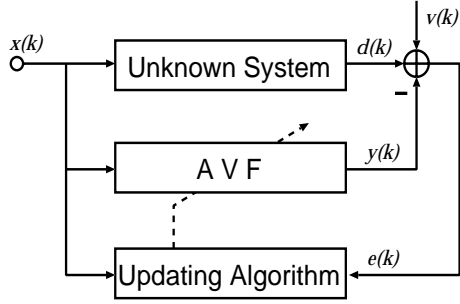


図 1: 適応 Volterra フィルタによるシステム推定

更新アルゴリズムとして NLMS アルゴリズムを用いることにする。

1 次項の更新式

$$\hat{h}_{1,n+1}(k_1) = \hat{h}_{1,n}(k_1) + \frac{\mu e_n x_n(k_1)}{k_1 - N + 1 + \sum_{i=k_1} x_n^2(i)} \quad (3)$$

2 次項の更新式

$$\begin{aligned} \hat{h}_{2,n+1}(k_1, k_2) &= \hat{h}_{2,n}(k_1, k_2) \\ &+ \frac{\mu e_n x_n(k_1) x_n(k_2)}{k_1 - N + 1 + k_2 - N + 1 + \sum_{i_1=k_1} \sum_{i_2=k_2} \{x_n(i_1) x_n(i_2)\}^2} \end{aligned} \quad (4)$$

3 有効な項を選択する AVF

3.1 フィルタ構造

フィルタ構造を図 2 に示す。'非線形化'では、先に述べた Volterra フィルタの項を生成する。生成される

N 次元入力 M 次 Volterra フィルタの項は次のようになる

$$x^{l_0}(n) x^{l_1}(n-1) \cdots x^{l_{N-1}}(n-N+1) \quad (5)$$

$$\text{ただし } \sum_{j=0}^{N-1} l_j \leq M$$

'項数制限'ではこれらの項に、次節で述べる方法で求めた遺伝子 $g_\alpha = 1$ or 0 をかけることにより、 $g_\alpha = 1$ の有効な項のみ出力されるように制限する。したがって、出力は次のようになる。

$$y(n) = \sum_i g_{\alpha_i} h_i x^{l_0}(n) x^{l_1}(n-1) \cdots x^{l_{N-1}}(n-N+1)$$

3.2 学習における項選択の必要性

式 (5) で表されるすべての高次多項式を用いてフィルタ係数の学習を行ない、フィルタ係数の絶対値が大きい項のみを選択することで、項数を制限することが可能であるが、AVF では項数が非常に多くなる場合があり、学習が遅くなる、最適解への収束が難しいという問題がある。本稿で提案する方法では、学習に使用する項を有効な項に制限することにより学習の高速化を図る。

4 有効な項の選択法

4.1 遺伝的アルゴリズム (GA)

本稿で用いている項数制限法は GA をベースとしているので、まず GA について簡単に述べる。

GA とは、選択淘汰、突然変異という生物進化過程の原理に基づき、離散的な組み合わせ問題を解くアルゴリズムである。問題の解候補を染色体に写像してその集団に選択・交叉・突然変異という操作を加えるこ

とによって最適な解を探索する。染色体は遺伝子の1次元配列となっており、

$$c = [g_1, g_2, \dots, g_L] \quad (6)$$

と表される。 g_i は遺伝子である。各個体は1個の染色体を持っている。

個体の最適化は次のようなステップを踏んで行なわれる。

1. 個体の初期集団生成
2. 個体の学習
3. 個体の優性度評価
4. 優性個体の選択・交叉・突然変異により次世代の個体を生成し、2に戻る。

この操作を繰り返すことにより、集団の優性度が増し、よりよい解を求めることができる。

4.2 有効な項の選択アルゴリズム

文献 [7] で提案された方法を AVF に適用する。ここでは1個の AVF が個体に対応する。AVF で選択されている項は染色体の遺伝子で表される。個体数を各世代で P 個とし、個体における遺伝子の総数を L 個とする。

- Step1
初期集団としてランダムに $g_\alpha = 1$ の遺伝子を K_0 個持つ個体を P 個生成する。

- Step2
Step1 or 3 で生成された個体を用いて AVF の学習を行ない、それぞれの誤差を次式で求める。

$$10 \log \left(\frac{\text{出力誤差の } MSE}{\text{所望信号の } MSE} \right) \quad (7)$$

- Step3
 P 個の個体のうち、誤差の小さい順に優性個体を P' 個選ぶ。 P' 個の中から2個の個体 G_1, G_2 を選び、次の方法で子を生成する。

G_1 が G_2 より優性であるとする、まず G_1 の $g_\alpha = 1$ である遺伝子の中からフィルタ係数の絶対値が最も大きいものを選ぶ。次に G_2 から同様に選ぶ。さらに $G_1 \quad G_2 \quad \dots$ とこの操作を K_n 回行なう。もし、 G_1 と G_2 で選ばれた遺伝子がすべて異なる場合は K_n 個の遺伝子が選択されるが、選ばれた遺伝子に重複がある場合は K_n 個未満の遺伝子が選択されることになる。このようにして子を

P 個生成する。その中で $g_\alpha = 1$ の遺伝子が最も多い数を新たに K_{n+1} とし、Step2 に戻る。

(例題)

$L = 6, K_0 = 4$ の例を図3に示す。 G_1, G_2 の順に優性度が高いとする。また、遺伝子の上に書かれた数字は対応するフィルタ係数の絶対値の大きさ順を表している。実際に G_1, G_2 から子を生成すると、選ばれる遺伝子は G_1 (1番目) G_2 (3番目) G_1 (3番目) G_2 (5番目) の順に選ばれる。 G_1, G_2 とも3番目の遺伝子が選ばれているので K_1 は4 3 となる。

$$\begin{array}{l} G_1 = \overset{1}{1}, \overset{2}{0}, 1, 1, 0, 1 \\ G_2 = \overset{1}{0}, 1, \overset{2}{1}, 1, 1, 0 \end{array} \rightarrow G_{1,2} = \{1, 0, 1, 0, 1, 0\}$$

図3: 子の生成例

4.3 フィルタ係数の継承

第1世代におけるフィルタ係数の初期値は零に設定するが、次世代以降においては親のフィルタ係数を継承することにする。ある程度有効な項が集まってくれば、近似解が得られ、そのフィルタ係数も最適解に近づいたものになっているので、これを次世代で継承することにより学習を高速化できる。

4.4 計算量及び収束速度

提案する学習法における計算量/サンプル間隔と収束速度は次のように表される。

$$\text{計算量} : \max \{k M_i K_i\} / \text{サンプル} \quad (8)$$

$$\text{収束速度} : \sum_{i=1}^I N_i \quad (9)$$

ただし、 k は NLMS で決まる定数、 I は必要な世代数、 N_i は収束するまでのサンプル数、 M_i は第 i 世代における有効な項数、 K_i は第 i 世代における個体数である。収束速度の評価は、収束するまでに必要な入力サンプル数に相当する。たとえ計算量が多くても、1サンプル内に計算が終了することを前提にしている。式(9)の収束速度の評価では、1サンプル内で K_i 個の個体の計算が行われることを前提にしている。

一方、すべての項を使用する従来の AVF では、次のようになる。

$$\text{計算量} : k M_0 / \text{サンプル} \quad (10)$$

$$\text{収束速度} : N_0 \quad (11)$$

ただし、 M_0 は全項数、 N_0 は収束するまでのサンプル数である。

5 計算機シミュレーション

5.1 シミュレーション条件

提案法の有効性を確認するため、入力が白色と有色の2通りにおいて、本方法と項数制限を行わない従来の適応 Volterra フィルタとの比較を行なった。有色信号は2次のARモデルに白色雑音を入力して生成した。2次ARモデルは

$$C(z) = \frac{1}{1 - 2r\cos\theta \cdot z^{-1} + r^2 z^{-2}} \quad (12)$$

ただし $r = 0.9$ 、 $\theta = \frac{\pi}{4}$ とした。

他のシミュレーション条件を以下のように設定した。

各世代の個体数 $P = 10$

淘汰後の個体数 $P' = 7$

初期制限項数 $K_0 = 700$

世代数 = 10

入力次元 $N = 50$

Volterra 次数 $M = 2$

入力が50次元、Volterra関数が2次なので、個体の総遺伝子数=Volterraフィルタの項数は1325個となる。未知システムはVolterraフィルタとし、50個の項をランダムに選択し、その係数を乱数で与えて構成した。

5.2 シミュレーション結果

i) 入力信号が白色の場合

各世代での誤差が最小である個体における有効な項数、収束誤差を表1に示す。

表 1: 各世代での有効項数と収束誤差

世代 - 個体	有効な項数	収束誤差 [dB]
1 - 7	700	-4.3
2 - 4	598	-12.0
3 - 5	475	-43.7
4 - 9	420	-59.1
5 - 4	289	-59.5

世代を経るにつれて、有効な項数が制限されていき、誤差が減少しているのがわかる。第4、5世代でほぼ収束している。

次に、フィルタ係数を継承する場合 (Succession) と継承しない場合 (No-Succession) における各世代の学習曲線を図4に示す。第1世代では初期値が零であるので同じ収束特性であるが、2世代、3世代と世代を経るうちに No-Succession は収束速度が遅くなる。一方、Succession は3世代目からは逆に速くなっており、重み係数継承の効果がでてきていることがわかる。第5世代では非常に速い収束特性を示している。

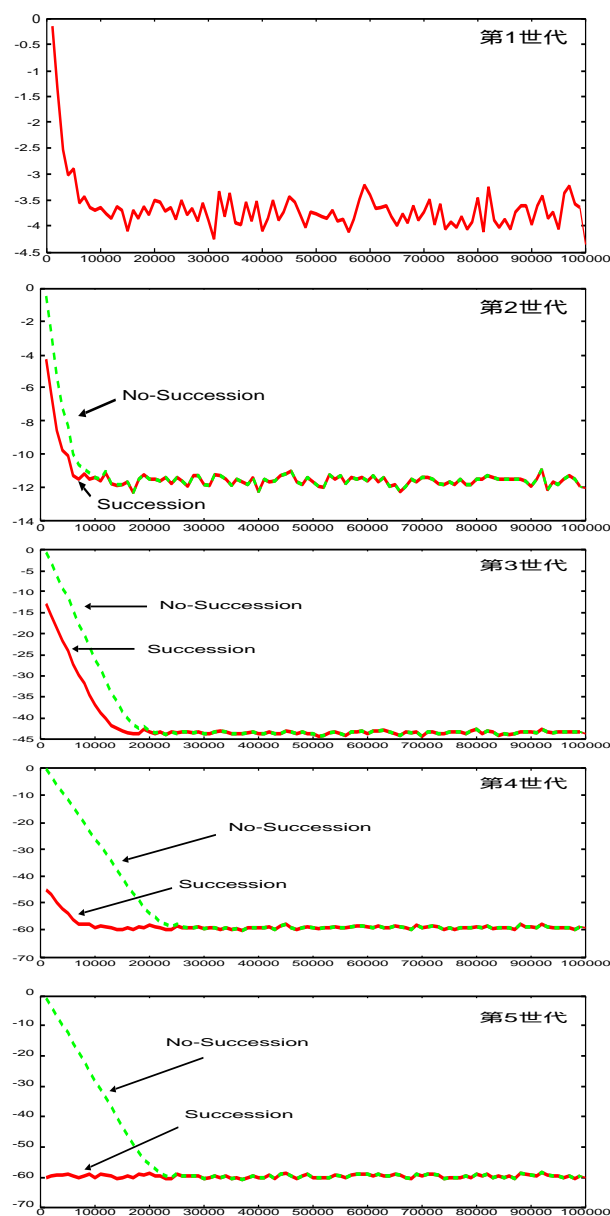


図 4: 継承する場合 (Succession) と継承しない場合 (No-Succession) の各世代の学習曲線

このシミュレーション結果で分かるもう一つの特徴は、世代を経るごとに誤差の収束値が段階的に減少していることである。初期の世代では誤差は十分に減少

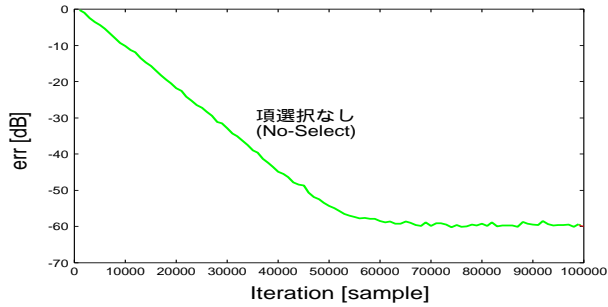


図 5: 項選択をしない場合 (No-Select) の学習曲線

していない。しかし、この結果においても有効な項に対するフィルタ係数が大きくなっており、次世代で選択することが可能である。有効な項が増えるにつれて、誤差は小さくなり、第 4、5 世代で収束している。

ここで、実際に式 (8) ~ (11) にしたがって項選択をする場合としない場合の収束速度と計算量を計算し、比較する。項選択をしない場合の学習曲線を図 5 に示す。

まず、項選択をする場合の計算量は

$$\begin{aligned} \text{計算量 (Select)} &= k \times 700 \times 10 / \text{サンプル} \\ &= 7000k / \text{サンプル} \end{aligned}$$

一方、項選択をしない場合の計算量は

$$\begin{aligned} \text{計算量 (No-Select)} &= 1325k / \text{サンプル} \end{aligned}$$

となり、項選択をした方が約 5 倍計算量が多い。同様に、収束速度を求める。

収束速度 (Select)

$$\begin{aligned} &= \{10000 + 8000 + \dots + 1000\} \\ &= 40000 \end{aligned}$$

収束速度 (No-Select)

$$= 60000$$

となり、項選択をした方が収束速度が速いことがわかる。

要約すると、提案した方法は従来法に比べて、計算量が 5 倍になるが、収束時間は 2/3 に短縮されることに相当する。

ii) 入力有色雑音の場合

白色雑音の場合と同様に、フィルタ係数を継承する場合としない場合において奇数世代の学習曲線を図 6 に示す。

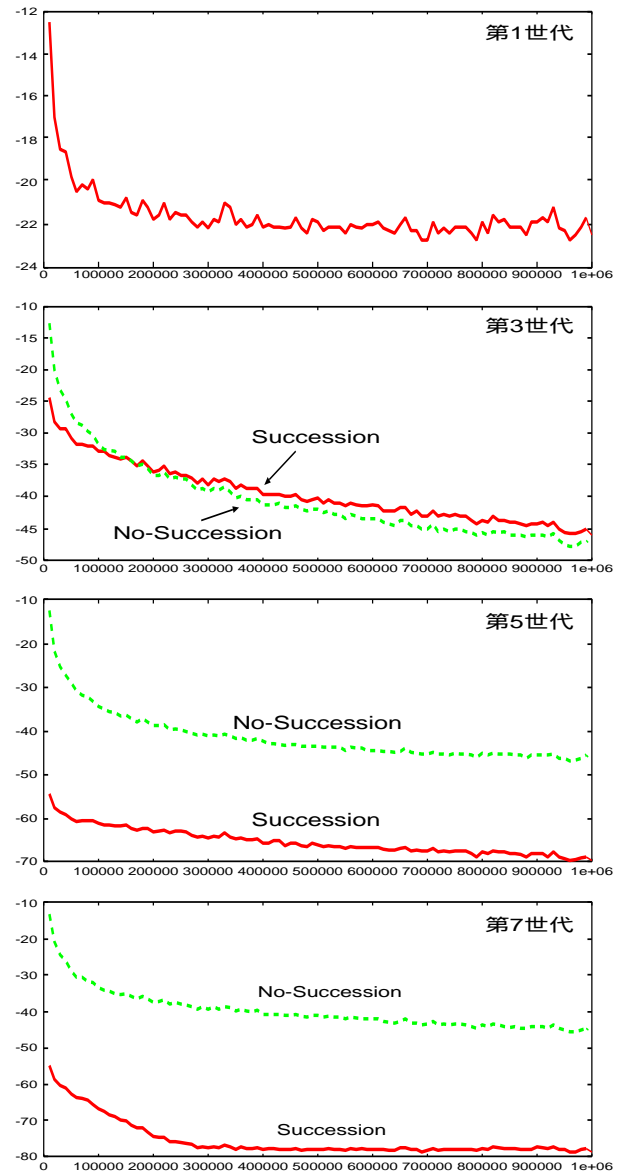


図 6: 継承する場合 (Succession) と継承しない場合 (No-Succession) の各世代の学習曲線

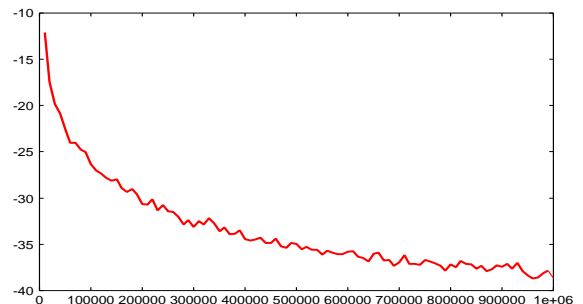


図 7: 項選択をしない場合の学習曲線

このシミュレーション結果の特徴は、フィルタ係数を継承しない場合は第5、7世代で収束が非常に遅くなっている。入力信号が有色の場合でも、フィルタ係数が最適解に近づいていけば、NLMSでも高速に収束することを示している。シミュレーションでは、 10^6 サンプルまで表示されているが、実際の学習では $10^5 \sim 2 \times 10^5$ サンプルにおける結果を次世代に送ることができる。

一方、Volterra関数の全項を使用する従来の方法は図7に示すように 10^6 サンプルにおいても-38dB程度であり、また、この学習曲線を延長すると-80dBに達するまでに $4 \times 10^6 \sim 5 \times 10^6$ サンプルを要することがわかる。

ここで、白色の場合と同様に式(8)~(11)に基づき、計算量と収束速度を計算する。

項選択をする場合

計算量 (Select)

$$= k \times 700 \times 10 / \text{サンプル}$$

$$= 7000k / \text{サンプル}$$

収束速度 (Select)

$$= \{250000 + \dots + 280000\}$$

$$= 1.53 \times 10^6$$

項選択をしない場合

計算量 (No-Select)

$$= 1325k / \text{サンプル}$$

収束速度 (No-Select)

$$= 4 \times 10^6 \sim 5 \times 10^6$$

提案法は計算量は5倍であるが、収束時間は約35%に短縮される。適応Volterraフィルタは有色信号に対して収束特性が劣化する。この場合でも、学習で使用する項を制限することにより高速化を図ることが出来る。

6 むすび

適応Volterraフィルタにおいてすべての項を使用するのではなく、有効な項を選択して使用方法を提案した。その方法は遺伝的アルゴリズムをベースとしたものであり、フィルタ係数の大きさを利用して項を制限するものである。また、次世代にフィルタ係数を継承することによって収束速度の改善を図った。計算機シミュレーションによれば、1サンプルあたりのピーク計算量は提案方法が5倍程度多く、収束時間は入力信号が白色の場合で2/3、有色の場合で1/3程度に削減されることが確認できた。

参考文献

- [1] A.Stenger, L.Trautmann, R.Rabenstein, "Non-linear Acoustic Echo Cancellation With 2nd Order Adaptive Volterra Filters," ICASSP'99, Phoenix, Arizona, May 1999.
- [2] S.Kinosita, Y.Kajikawa, Y.nomura, "Volterra Filters Using Multirate Signal Processing And Their Application To Loudspeaker Systems," ICASSP'2001, Salt Lake City, May 2001.
- [3] Xiaohui Li and W.Kenneth Jenkins and Charles W.Therrien, "Computationally Efficient Algorithms For Third Order Adaptive Volterra Filters," ICASSP'98, Seattle, Washington, May 1998.
- [4] 梶川 嘉延, 野村 康雄, "適応 Volterra フィルタを用いた非線形逆システムのオンライン設計," 信学論, Vol.J82-A, No.1, pp1-10, 1999-1.
- [5] 趙 晋輝, 猪股 篤, "Volterra 形非線形適応フィルタの誤差曲面および収束特性の解析," 信学論, Vol.J82-A, No.6, pp809-816, 1999-6.
- [6] 趙 晋輝, 宇野 晋平, 久保田 智規, 猪股 篤, "Volterra 形非線形 FIR 適応フィルタの高速収束算法に関する考察," 信学論, Vol.J82-A, No.4, pp370-378, 2000-4.
- [7] K.Nakayama, A.Hirano, M.Fusakawa, "A Selective Learning Algorithm for Nonlinear Synapses in Multilayer Neural Networks," IJCNN'01, Washington DC., pp.1704-1709, 2001.
- [8] 高濱 優里, 梶川 嘉延, 野村 康雄, "NLMS 法を用いた 2 次適応 Volterra フィルタの収束特性の定式化," 信学論, Vol.J82-A, No.7, pp944-953, 1999-7.
- [9] Shue-Lee Chang, Tokunbo Ogunfunmi, "Performance Analysis of Third-Order Nonlinear Wiener Adaptive Systems," ICASSP'99, Phoenix, Arizona, May 1999.
- [10] A.Stenger, W.Kellermann, "Nonlinear Acoustic Echo Cancellation With Fast Converging Memoryless Preprocessor," ICASSP'2000, Istanbul, Turkey, June 2000.