

A LATTICE PREDICTOR BASED ADAPTIVE VOLTERRA FILTER AND ITS CONVERGENCE ANALYSIS

Kenji Nakayama* Akihiro Hirano*
Hiroaki Kashimoto*

** Dept. of Information and Systems Eng., Faculty of Eng.,
Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920-8667, JAPAN
e-mail:nakayama@t.kanazawa-u.ac.jp*

Abstract: This paper proposes a lattice predictor based adaptive Volterra filter, and its convergence property is analyzed. In the adaptive Volterra filter (AVF), the eigenvalue spread of a correlation matrix is extremely amplified, and its convergence is very slow for gradient methods. A lattice predictor is employed for whitening the input signal. Its convergence property is analyzed. For colored signals, generated using an AR model, it can fast converge to the well reduced level. However, the convergence is sensitive to error of the whitening. When the reflection coefficients are updated, convergence is highly dependent on a time constant parameter used in updating the reflection coefficients. In the case of using a time variant AR model, that is nonstationary input signals, almost the same convergence is obtained compared with the fixed AR model. Furthermore, update of the reflection coefficients and the filter coefficients is not synchronized in the conventional lattice predictor based adaptive filters. This effect is also analyzed.

Keywords: Adaptive filters, Volterra filters, Lattice predictor

1. INTRODUCTION

Loud speakers in audio systems and small speakers embedded in a mobile phone have some nonlinearity. When they are used in a remote conference system and a visual phone, in which some echo are caused, nonlinear echo cancellers are very important.

Adaptive Volterra filters are one of hopeful candidates [1],[2],[3],[4]. It can express general nonlinearity. However, the Volterra polynomial has a huge number of terms, and the same number of filter coefficients are required. Furthermore, when the input signal is colored, the eigenvalue spread of a correlation matrix is extremely amplified, and convergence is very slow for gradient methods.

Many kinds of fast and stable learning algorithms for adaptive Volterra filters have been proposed [5],[3]. RLS algorithm is insensitive to the eigenvalue spread, at the expense of $O(N^2)$ computations. Another method is to combine a whitening process and an adaptive FIR Volterra filter. The Discrete Cosine Transform (DCT) has been applied to the whitening process [7]. Furthermore, an error surface and convergence property have been analyzed [8]. The DCT is not sufficient for the whitening process. A linear FIR predictor based on an AR model of the signal is good for whitening. However, it requires some time delay, and cannot be applied to some applications [6].

In this paper, in order to improve the whitening process without any time delay, a lattice predic-

tor is employed for a whitening process. Convergence properties are analyzed based on accuracy of whitening, a time constant parameter, and non-stationary input signals. The proposed and conventional methods are compared with each other through computer simulations by using colored signals and real speech signals.

2. ADAPTIVE FIR VOLTERRA FILTER

2.1 Structure of AVF

Figure 1 shows a block diagram of an adaptive FIR Volterra filter (AVF). When a second-order

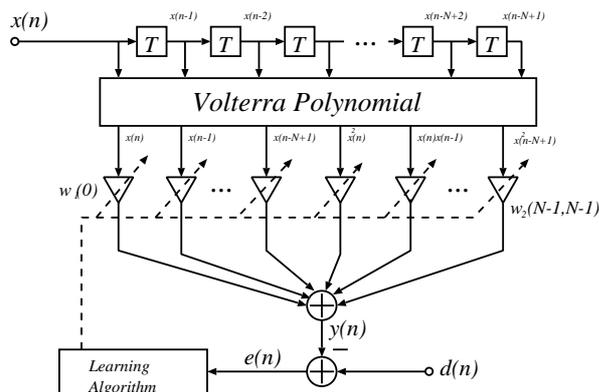


Fig. 1. Adaptive FIR Volterra filter.

Volterra polynomial is used, the output $y(n)$ is given by

$$y(n) = \sum_{i=0}^{N-1} w_1(i)x(n-i) + \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} w_2(j,k)x(n-j)x(n-k) \quad (1)$$

$$w_2(j,k) = w_2(k,j) \quad (2)$$

2.2 Eigenvalue Spread Amplification

The eigenvalue spread $\chi = \lambda_{max}/\lambda_{min}$ of the input signal $x(n)$ is extremely amplified by transforming it through the Volterra polynomial. Examples are shown in Table 1. The colored input signal $x(n)$ with $\chi = 780.9$ is transferred to the Volterra polynomial terms having $\chi = 657100$, which is 841 times as large as that of $x(n)$. For this reason, convergence of the adaptive Volterra filter is very slow for gradient methods.

Table 1. Eigenvalue spread amplification

Signal	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$
White Signal	0.716	1.359	1.898
White+Volterra	0.781	52.21	66.83
Colored Signal	0.08151	63.66	780.9
Colored+Volterra	0.01003	6590	657100

2.3 Whitening Input Signal

2.3.1. Discrete Cosine Transform Figure 2 shows the DCT with normalization [7]. The outputs of the tap delay line $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]$ are transformed through the DCT to $\mathbf{q}(n) = [q_0(n), q_1(n), \dots, q_{N-1}(n)]$, and they are normalized by its standard deviation $\sigma_{q,i}$. The outputs $s_i(n)$ of this block are applied to the Volterra polynomial generating 1st-order and high-order terms. These terms are multiplied by filter coefficients, and are accumulated, resulting in the final output $y(n)$. The DCT does not need any time delay.

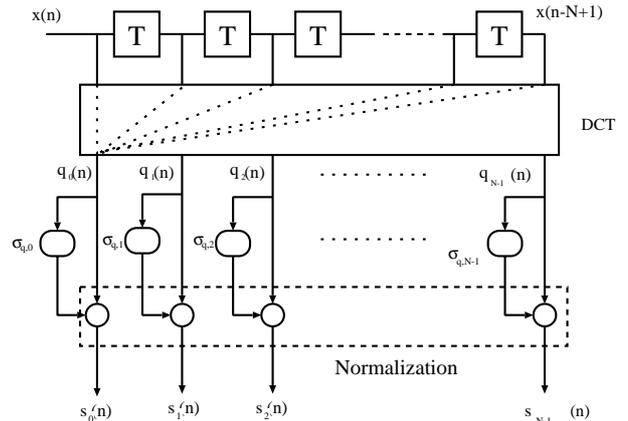


Fig. 2. DCT whitening process with normalization.

2.3.2. Linear FIR Predictor Based on AR Signal Model

When the signal can be modeled by the output of an AR circuit driven by the white noise, a linear FIR prediction error filter shown in Fig. 3 is good for whitening. $e(n)$ is used as the AVF input.

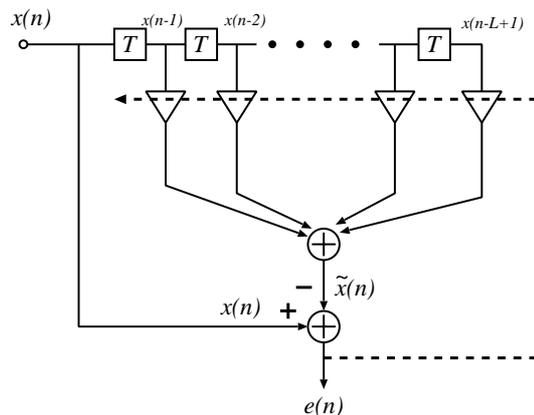


Fig. 3. Linear predictor based on FIR filter.

2.4 Position of Whitening in Nonlinear Filters

Two kinds of positions for the whitening are shown in Figs. 4 and 5, and are denoted Type-A and Type-B, respectively. Any time delay is

not allowed in Type-A. Reason can be explained as follows: Suppose the linear predictor shown in Fig.3 is used in Type-A, and both the unknown system and the AVF have the N -th order FIR filter and a 2nd-order Volterra polynomial. The output of the unknown system includes $x(n-i), i = 0, 1, \dots, N-1$ and their high-order terms. On the other hand, in the AVF part, $e(n)$, which consists of $x(n-i), i = 0, 1, \dots, L-1$, is the input of the AVF, then $x(n-i), i = 0, 1, \dots, N+L-1$ and their high-order terms are included in its output signal. Therefore, their transfer functions are inherently different.

On the other hand, in Type B, any kinds of the whitening process, with or without time delay, can be employed. However, since the output of the unknown system may be sound from a loud speaker located in a conference room, it cannot be applied to some applications, such as echo cancellation.

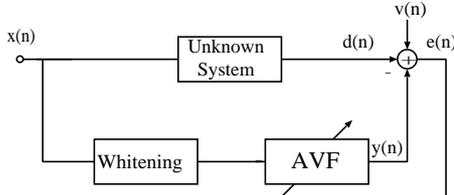


Fig. 4. Whitening adaptive filter input (Type-A).

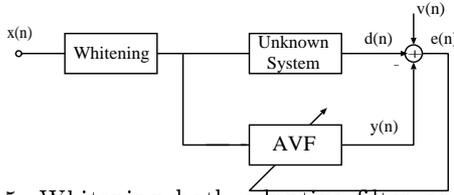


Fig. 5. Whitening both adaptive filter and unknown system inputs (Type-B).

3. LATTICE PREDICTOR BASED AVF

3.1 Circuit Structure

In practical applications, Type A is important. Therefore, we employ the lattice predictor [9] for the whitening process. The proposed lattice predictor based AVF is shown in Fig.6. Under the conditions the delay line order is N , the order of the Volterra polynomial is M , the order of the lattice predictor is L , and $N > L$, the order of the transfer function $Y(z)/X(z)$, and the number of filter coefficients in both AVF in Fig.1 and Fig.6 are the same. If the unknown system can be modeled by using the FIR Volterra filter, the same transfer function can be realized by the lattice predictor based AVF.

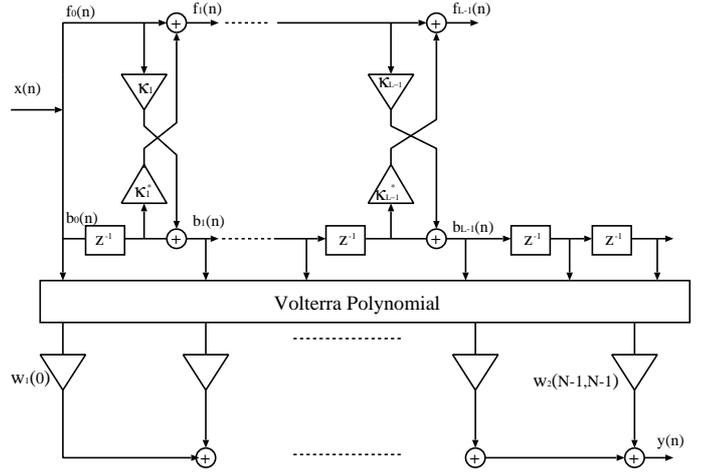


Fig. 6. Lattice predictor based AVF.

3.2 Reflection Coefficient Update

The reflection coefficients are updated by the following equations [9].

$$\kappa_m(n) = -\frac{2E[b_{m-1}(n-1)f_{m-1}^*(n)]}{E[|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2]} \quad (3)$$

$$\kappa_{N,m}(n) = \gamma\kappa_{N,m}(n-1) + b_{m-1}(n-1)f_{m-1}^*(n) \quad (4)$$

$$\kappa_{D,m}(n) = \gamma\kappa_{D,m}(n-1) + |f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2 \quad (5)$$

$$0 < \gamma < 1$$

$$\kappa_m(n) = -2\frac{\kappa_{N,m}(n)}{\kappa_{D,m}(n)} \quad (6)$$

3.3 Synchronized Learning Algorithm

3.3.1. Lattice Predictor Based Adaptive Filters

Convergence property of the lattice predictor based FIR adaptive filter has been analyzed, and the synchronized learning algorithm has been proposed [10]. Updating the reflection coefficients and the filter coefficients are not synchronized, and some error remain.

The synchronizing method [10] is described here. The linear adaptive filter with the lattice predictor is equivalent to the circuit shown in Fig.6, except for the Volterra polynomial block. The filter coefficients $w(n)$ is directly connected to $b(n)$. The output $y(n)$ is

$$\mathbf{b}(n) = \mathbf{K}(n)\mathbf{x}(n) \quad (7)$$

$$y(n) = \mathbf{w}^T(n)\mathbf{b}(n) \quad (8)$$

$\mathbf{b}(n)$ is a vector of the backward prediction error $b_m(n)$, $\mathbf{K}(n)$ is a matrix consists of the reflection coefficients, $\mathbf{x}(n)$ is the input, $\mathbf{w}(n)$ is the filter coefficients. In the next iteration step, $\mathbf{K}(n)$ is updated to $\mathbf{K}(n+1)$, and $y(n+1)$ and $e(n+1)$ are generated by using $\mathbf{K}(n+1)$ and $\mathbf{w}(n)$.

However, $\mathbf{w}(n)$ is optimized for $\mathbf{K}(n)$ not $\mathbf{K}(n+1)$. Therefore, $e(n+1)$ is not guaranteed to be reduced. For this reason, $\mathbf{w}(n)$ is modified as,

$$\tilde{\mathbf{b}}(n+1) = \mathbf{K}(n)\mathbf{x}(n+1) \quad (9)$$

$$\tilde{y}(n+1) = \mathbf{w}^T(n)\tilde{\mathbf{b}}(n+1) \quad (10)$$

$$\mathbf{b}(n+1) = \mathbf{K}(n+1)\mathbf{x}(n+1) \quad (11)$$

$$y(n+1) = \tilde{\mathbf{w}}^T(n)\mathbf{b}(n+1) \quad (12)$$

$\tilde{y}(n+1)$ can reduce the output error. Therefore, the filter coefficients $\mathbf{w}(n)$ is modified as follows:

$$\mathbf{K}^T(n+1)\tilde{\mathbf{w}}(n) = \mathbf{K}^T(n)\mathbf{w}(n) \quad (13)$$

$$\tilde{\mathbf{w}}(n) = \frac{\mathbf{K}^T(n)}{\mathbf{K}^T(n+1)}\mathbf{w}(n) \quad (14)$$

$\tilde{\mathbf{w}}(n)$ is used in the next iteration $n+1$, instead of $\mathbf{w}(n)$, for generating $\tilde{y}(n+1)$ and $\tilde{e}(n+1)$. The filter coefficients are updated to $\mathbf{w}(n+1)$ by using $\tilde{e}(n+1)$ and $\tilde{\mathbf{w}}(n)$. A combination of $\mathbf{K}(n)$ and $\mathbf{w}(n)$ is equivalent to that of $\mathbf{K}(n+1)$ and $\tilde{\mathbf{w}}(n)$. Thus, the output error is guaranteed to be decreased.

4. SIMULATION AND DISCUSSIONS

4.1 Colored Signal

The colored signal is generated passing the white noise through a 2nd-order AR model, and is applied to the adaptive Volterra filter. The number of taps of the FIR filter is 50, and the order of Volterra function is 2nd-order. The number of the terms in the Volterra function is 1325.

4.1.1. DCT and Linear Predictor The learning curves for the AVF without whitening, and with the DCT method in Type A and the linear predictor in Type B are shown in Fig.7. The NLMS algorithm and stepsize=0.1 are employed.

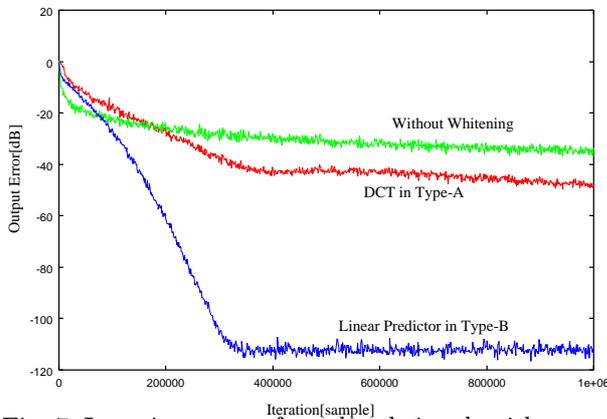


Fig. 7. Learning curves for colored signal, without whitening, and with DCT and linear prediction.

From these results, convergence is very slow, when the whitening process is omitted. The DCT can improve convergence to some extent. The FIR linear predictor in Type-B is good, because the whitening is complete. However, this type cannot be applied to echo canceler and so on.

4.1.2. Lattice Predictor Based AVF Figure 8 shows the learning curves of the lattice predictor based AVF. The reflection coefficients are calculated in advance and fixed. If the reflection coefficients are set to the ideal, convergence is fast and the residual error is very small. However, the convergence is sensitive to deviation of the reflection coefficients. Assuming the output error of -50dB, 5% deviation causes a long convergence time, which is 3 times as long as the ideal case. In this simulation, since the reflection coefficients are fixed, the asynchronous update problem of the reflection coefficients and the filter coefficients does not exist.

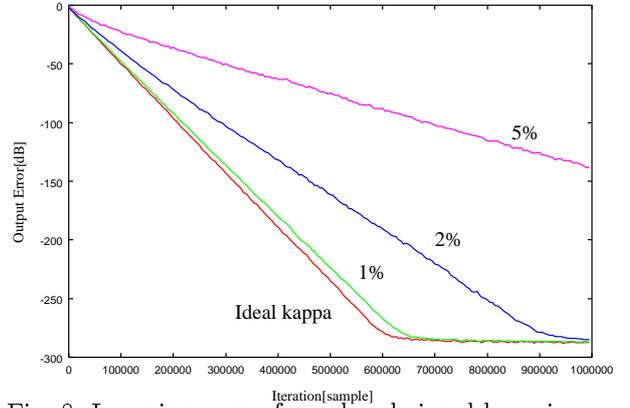


Fig. 8. Learning curve for colored signal by using lattice predictor based AVF. Reflection coefficients are fixed to ideal, 1%, 2%, and 5% deviation.

Figure 9 shows the learning curves, in which the reflection coefficients are updated based on Eqs.(3) through (6). Convergence depends on a time constant γ , which control the reflection coefficient update. From Eqs.(4) and (5), when $\gamma(< 1)$ is very close to unity, the reflection coefficients κ_i are very gradually adjusted. Until $\gamma = 0.999999$, the convergence can be improved. The learning curves for the lattice predictor almost saturate around 50dB, however, it requires only 85,000 iterations until -40dB, while the DCT and the linear predictor need 320,000 iterations and 150,000 iterations as shown in Fig.7, respectively.

Figure 10 shows the learning curves for the time variant AR model generating the colored signals. The pole, $re^{\pm\theta}$, of the AR model is controlled by

$$r = 0.9 \quad (15)$$

$$\theta(n) = \frac{\pi}{4} \left(1 + a \sin\left(\frac{2\pi n}{500}\right) \right) \quad (16)$$

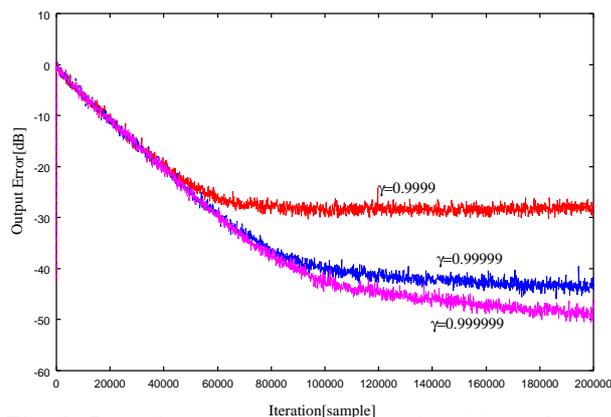


Fig. 9. Learning curve for colored signal by using lattice predictor based AVF.

a is changed as 0.05, 0.1 and 0.2. γ is set to 0.999999. The learning curves are almost the same as in Fig.9, where the AR model is fixed. From these results, the lattice predictor based AVF can be useful for nonstationary input signals.

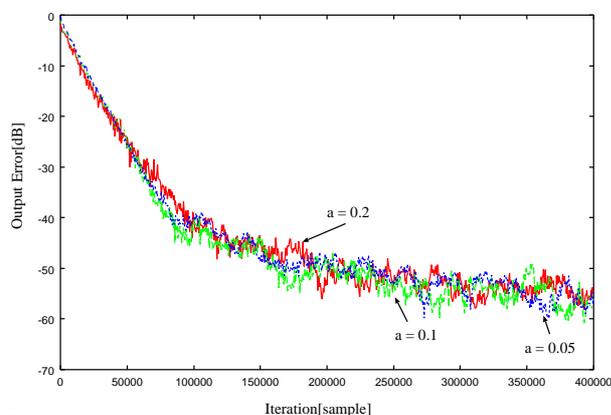


Fig. 10. Learning curve for colored signal generated by time variant AR model.

4.2 Speech Signal

Speech signal is used as the input signal of the AVF. Figures 11 and 12 show the learning curves by using the DCT and the linear predictor, and by using the lattice predictor, respectively. In the former figure, the mean squared error in some interval is normalized by the mean squared signal in the same interval. In the latter case, it is normalized by the mean squared signal in the enter interval. So, the curves are different. From these results, the DCT cannot improve from the NLMS without whitening. The linear predictor almost converges at 300,000 iterations. On the other hand, the lattice predictor converges at 30,000 iterations. Thus, convergence time can be reduced to 1/10 of the FIR linear predictor.

5. CONCLUSIONS

In this paper, the lattice predictor based AVF has been proposed. If the reflection coefficients are

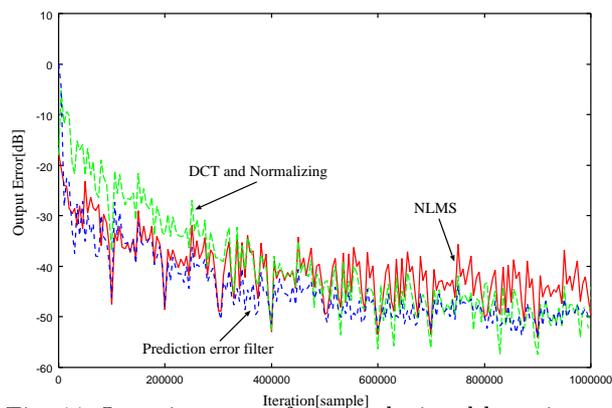


Fig. 11. Learning curve for speech signal by using DCT and linear predictor.

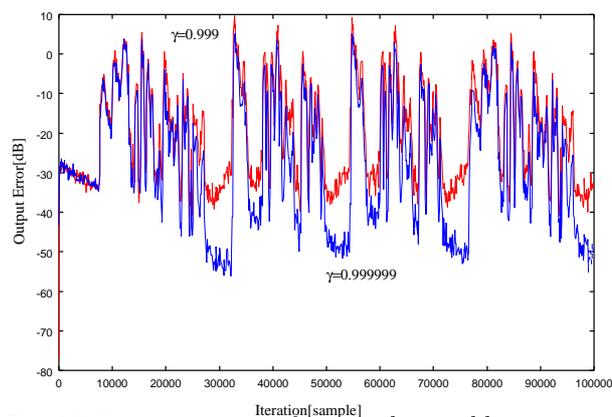


Fig. 12. Learning curves for speech signal by using lattice predictor.

ideal and are fixed, convergence is very good. It is also insensitive to nonstationary input signals generated by a time variant AR model. Its convergence is dependent on accuracy of whitening. 5% deviation of whitening requires a 3 times longer convergence time. When the reflection coefficients are updated, convergence depends on the time constant parameter. Compared with the conventional methods, convergence time can be improved to 1/3~1/10 for stationary and nonstationary colored signals.

REFERENCES

- [1] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, New York: Wiley, 1980.
- [2] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.*, pp. 10-26, July 1991.
- [3] A. Stenger, L. Trautmann and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters," *IEEE Proc. ICASSP'99*, AE1.7, pp877-880, 1999.
- [4] B.S. Nolle and D.L. Jones, "Nonlinear echo cancellation for hands-free speakerphones," *1997 IEEE Workshop on Nonlinear Signal and Image Proc*

- [5] J.Lee and V.J.Mathews, "A fast recursive least squares adaptive second order Volterra filter and its performance analysis," *IEEE Trans on Signal Processing*, Vol.41, No.3, pp.1087-1102, Mar 1993.
- [6] A.Strenger and W.Kellermann, "Nonlinear acoustic echo cancellation with fast converging memoryless preprocessor," *Proc. IEEE Workshop on Acoustic Echo and Noise Control*, Pocono Manor, PA, USA, September, 1999.
- [7] X.Li and W.K.Jenkins, "Computationally efficient algorithms for third order adaptive Volterra filters," *IEEE Proc. ICASSP'98*, DSP5.3, 1998.
- [8] J.Chao, A.Inomata, T.Kubota, and S.Uno, "Error surface analysis and fast RLS adaptive algorithms of quadratic Volterra ADF," *2001 IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, June 3-6, 2001.
- [9] S.Haykin, *Adaptive Filter Theory*, 3rd Ed., Prentice Hall Inc., New York, 1996.
- [10] N.Tokui, K.Nakayama and A.Hirano, "A synchronized learning algorithm for reflection coefficients and tap weights in a joint lattice predictor and transversal filter", *IEEE Proc. ICASSP'2001*, Salt Lake City, pp.1472-1475, May 2001.