A SYNCHRONIZED LEARNING ALGORITHM FOR NONLINEAR PART IN A LATTICE PREDICTOR BASED ADAPTIVE VOLTERRA FILTER

Kenji Nakayama *Akihiro Hirano* *Hiroaki Kashimoto*

Graduate School of Natural Science and Technology, Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920-8667, Japan
nakayama@t.kanazawa-u.ac.jp

ABSTRACT

A lattice predictor based adaptive Volterra filter (LP-AVF) is superior to the others with some whitening preprocessing. The LP-AVF has an asynchronous updating problem, which limit convergence property. Updating reflection coefficients of the lattice prediction error filter and an adaptive filter coefficients are not synchronized. A synchronized algorithm has been proposed for a linear model, which can be applied to the linear part of the LP-AVF. However, an asynchronous updating problem for the nonlinear part of the LP-AVF is still remain. In this paper, a new synchronized learning algorithm for the nonlinear part is proposed. An equivalent transfer function is introduced for the nonlinear part. The adaptive filter coefficients are compensated for during a learning process in order to maintain the transfer function to be the same for the next updated reflection coefficients. Simulation results using stationary and non-stationary colored input signals demonstrate efficiency of the proposed method.

1. INTRODUCTION

Laud 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.

An adaptive Volterra filter (AVF) is one of hopeful candidates [1],[2],[3]. 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 by the Volterra polynomial, and convergence is very slow for gradient methods.

Many kinds of fast and stable learning algorithms for adaptive Volterra filters have been proposed [4],[2]. One approach is to combine a whitening process and an adaptive FIR Volterra filter. The Discrete Cosine Transform (DCT) has been applied to the whitening process [6]. A linear FIR predictor based on an AR model of the signal is good for whitening [7]. However, it requires some time delay, and its application is limited [5]. In order to improve the whitening process without any time delay, a lattice predictor has been employed for a whitening process [9]. However, this approach has inherently an asynchronous updating problem [8].

In this paper, the asynchronous updating problem in the lattice predictor based AVF is analyzed in detail. Furthermore, a synchronized learning algorithm for the nonlinear part of the LP-AVF is proposed. Simulations under several conditions will be shown in order to confirm usefulness of the proposed method.

2. ADAPTIVE FIR VOLTERRA FILTER

Figure1 shows a blockdiagram of an adaptive FIR Volterra filter (AVF). When a second-order Volterra

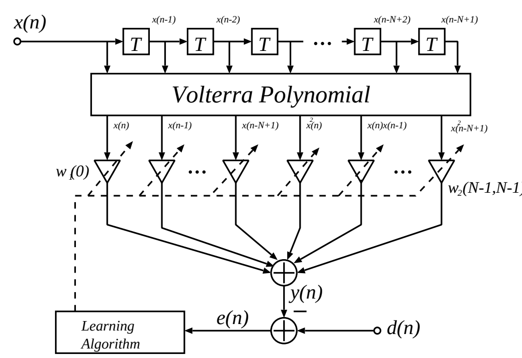

Figure 1: Adaptive FIR Volterra filter.

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

$$y(n) = \sum_{i=0}^{N-1} w_1(i)x(n-i)$$

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

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

3. LATTICE PREDICTOR BASED AVF

3.1 Circuit Structure

In practical applications, Type A is important. Therefore, the lattice predictor [7] has been employed for the whitening process [9]. The lattice predictor based AVF (LP-AVF)is shown in Fig.2. Letting the delay line order be N, order of the Volterra polynomial be M, order of the lattice predictor be L, and $N > L$, order of the transfer function $Y(z)/X(z)$, and the number of filter coefficients in both AVF in Fig.1 and Fig.2 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 LP-AVF.

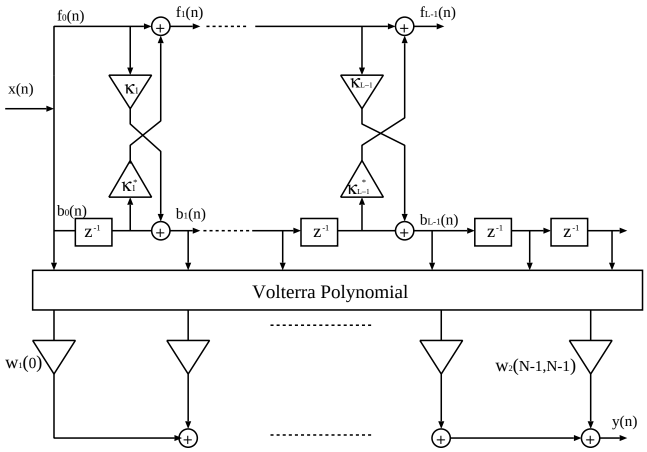

Figure 2: Lattice predictor based AVF.

3.2 Reflection Coefficient Update

The reflection coefficients are updated by the following equations [7].

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

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

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

$$0 < \gamma < 1$$

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

3.3 Asynchronous Updating Problem

Convergence property of the lattice predictor based linear adaptive filter (LP-AF) has been analyzed, and the synchronized learning algorithm has been proposed [8]. Updating the reflection coefficients and the filter coefficients are not synchronized, and the output error is lower bounded . The linear LP-AF is equivalent to the circuit shown in Fig.2, except for the Volterra polynomial block. The filter coefficients $\boldsymbol{w}(n)$ is directly connected to $\boldsymbol{b}(n)$.

$$\boldsymbol{b}(n) = \boldsymbol{K}^T(n)\boldsymbol{x}(n) \quad (7)$$

$$y(n) = \boldsymbol{w}^T(n)\boldsymbol{b}(n) = \boldsymbol{w}^T(n)\boldsymbol{K}^T(n)\boldsymbol{x}(n) \quad (8)$$

$$e(n) = d(n) - y(n) \quad (9)$$

$\boldsymbol{b}(n)$ is a vector of the backward prediction error $b_m(n)$, $\boldsymbol{K}(n)$ is a matrix consists of the reflection coefficients, $\boldsymbol{x}(n)$ is the input, $\boldsymbol{w}(n)$ is the filter coefficients and $d(n)$ is a desired response. $\boldsymbol{w}(n)$ is updated to $\boldsymbol{w}(n+1)$ using $\boldsymbol{b}(n)$ and $e(n)$. In the next sample, $\boldsymbol{K}(n)$ is updated to $\boldsymbol{K}(n+1)$, and $y(n+1)$ and $e(n+1)$ are generated by using $\boldsymbol{K}(n+1)$ and $\boldsymbol{w}(n+1)$. However, $\boldsymbol{w}(n+1)$ is optimized by using $\boldsymbol{K}(n)$. Therefore, the combination of $\boldsymbol{K}(n)$ and $\boldsymbol{w}(n+1)$ can reduce $e(n+1)$. However, $\boldsymbol{w}(n+1)$ is combined to $\boldsymbol{K}(n+1)$, and the output error $e(n+1)$ cannot be well reduced. This is an asynchronous updating problem [8]

4. SYNCHRONIZED LEARNING ALGORITHM

4.1 Linear Part of LP-AVF

The synchronized learning algorithm for the linear LP-AF [8] can be directly applied to the linear part of the LP-AVF. This algorithm is described here.

From Eq.(8), $\boldsymbol{w}^T(n)\boldsymbol{K}^T(n)$ can be regarded as a transfer function. $\boldsymbol{w}(n+1)$ is modified so as to maintain the transfer function using $\boldsymbol{K}(n+1)$ to be the same as that using $\boldsymbol{K}(n)$ as follows:

$$\tilde{\boldsymbol{w}}^T(n+1)\boldsymbol{K}^T(n+1) = \boldsymbol{w}^T(n+1)\boldsymbol{K}^T(n) \quad (10)$$

$$\boldsymbol{K}(n+1)\tilde{\boldsymbol{w}}(n) = \boldsymbol{K}(n)\boldsymbol{w}(n) \quad (11)$$

$$\tilde{\boldsymbol{w}}(n+1) = \frac{\boldsymbol{K}(n)}{\boldsymbol{K}(n+1)}\boldsymbol{w}(n) \quad (12)$$

$\tilde{\boldsymbol{w}}(n+1)$ is used in the next iteration $n+1$, instead of $\boldsymbol{w}(n+1)$, for generating $y(n+1)$ and $e(n+1)$. $\tilde{\boldsymbol{w}}(n+1)$ is updated to $\boldsymbol{w}(n+2)$ by using $\boldsymbol{K}(n+1)$.

4.2 Nonlinear Part of LP-AVF

Assuming the 2nd-order Volterra polynomial, the 2nd-order terms can be expressed by

$$\boldsymbol{b}(n) = [b(n), b(n-1), \cdots, b(n-N+1)]^T \quad (13)$$

$$\boldsymbol{b}(n) = \boldsymbol{K}^T(n)\boldsymbol{x}(n) \quad (14)$$

$$\boldsymbol{B}(n) = \boldsymbol{b}(n)\boldsymbol{b}^T(n) = \boldsymbol{K}^T(n)\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{K}(n) \quad (15)$$

$$= \boldsymbol{K}^T(n)\boldsymbol{R}(n)\boldsymbol{K}(n) \quad (16)$$

$$\boldsymbol{R}(n) = \boldsymbol{x}(n)\boldsymbol{x}^T(n) \quad (17)$$

$$\beta(n,i,j) = b(n-i)b(n-j), 0 \le i,j \le N-1 \quad (18)$$

The inputs of the Volterra polynomial in Fig.2 are denoted $b(n-i)$ for convenience. $\beta(n,i,j)$ is the (i,j)-th element of $\boldsymbol{B}(n)$, which is a symmetrical matrix, and includes all 2nd-order terms of the Volterra polynomial. The nonlinear part of the AVF output can be expressed by

$$y(n) = tr[\boldsymbol{W}(n)\boldsymbol{B}(n)] \quad (19)$$

$tr[\boldsymbol{A}]$ is a trace of a matrix $\boldsymbol{A}$. $\boldsymbol{W}(n)$ is a filter coefficient matrix. The i-th diagonal element of $\boldsymbol{W}(n)\boldsymbol{B}(n)$ is an inner product of the i-th row of $\boldsymbol{W}(n)$ and the i-th column of $\boldsymbol{B}(n)$. Thus, the elements of $\boldsymbol{W}(n)$ can express the filter coefficients for the nonlinear part.

By substituing Eq.(16), Eq.(19) can be rewritten as follows:

$$y(n) = tr[\boldsymbol{W}(n)\boldsymbol{K}^T(n)\boldsymbol{R}(n)\boldsymbol{K}(n)] \quad (20)$$

$$= tr[\boldsymbol{R}(n)\boldsymbol{K}(n)\boldsymbol{W}(n)\boldsymbol{K}^T(n)] \quad (21)$$

In the above equations, the property of the trace, that is $tr[\boldsymbol{AB}] = tr[\boldsymbol{BA}]$ is applied. From the above equation, $\boldsymbol{K}(n)\boldsymbol{W}(n)\boldsymbol{K}^T(n)$ can be regarded as an equivalent transfer function. A synchronized learning algorithm can be derived by modifying the filter coefficients $\boldsymbol{W}(n)$ so as to maintain the transfer function to be the same as before updating the reflection coefficients.

Let $\boldsymbol{W}(n)$ is updated to $\boldsymbol{W}(n+1)$ by using $\boldsymbol{K}(n)$, then the condition for the synchronized learning is expressed by

$$\boldsymbol{K}(n+1)\tilde{\boldsymbol{W}}(n+1)\boldsymbol{K}^T(n+1) = \boldsymbol{K}(n)\boldsymbol{W}(n+1)\boldsymbol{K}^T(n) \quad (22)$$

The modified filter coefficients are given by

$$\tilde{\boldsymbol{W}}(n+1) = \boldsymbol{K}^{-1}(n+1)\boldsymbol{K}(n)\boldsymbol{W}(n+1) \times\boldsymbol{K}^T(n)\boldsymbol{K}^T(n+1)^{-1} \quad (23)$$

$\tilde{\boldsymbol{W}}(n+1)$ is used at the next iteration instead of $\boldsymbol{W}(n+1)$ in the same way as the linear part.

5. SIMULATION AND DISCUSSIONS

5.1 Simulation Conditions

The AVF is applied to a system identification problem. Stationary and nonstationary colored signals, which are generated through a 2nd-order AR model, are used. A pole is given by $re^{\pm\theta}, r = 0.9, \theta = \pi/4$ for stationary signal, and $\theta(n) = (\pi/4)(1 + a\sin(2\pi n/500)), a = 0.2$ for nonstationary signal. Stepsize is optimized for each whitening methods and input signals by experience. The number of taps of the delay line is 50, and the 2nd-order Volterra polynomial is used. Therefore, 1325 terms, including 50 linear terms and 1275 nonlinear terms, are used. The unknown system is also realized using the FIR-Volterra filter shown in Fig.1 with fixed coefficients, which are determined as random numbers. The NLMS algorithm is employed for all cases.

5.2 Comparison among Several Whitening Methods

The DCT and normalization method [6] and the linear prediction error filter [7] are used for comparison. The LP-AVF with asynchronous updating is used. The time constant γ is set to be 0.999999. Effects of γ will be discussed later. Figure3 shows learning curves for the nonstationary colored signal. The LP-AVF is superior to the other two methods.

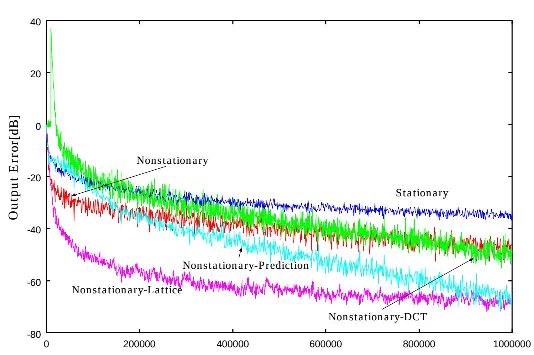

Figure 3: Learning curves for nonstationary colored signal.

5.3 Asynchronous Updaing Problem

As discussed in Sec.3.3, the LP-AVF has inherently the asynchronous updating problem. This property is analyzed here.

In the case of the stationary colored signal, the reflection coefficients $\kappa(n)$ are deviated from the ideal κ_o following

$$\kappa(n) = \kappa_o(1 + a) \quad (24)$$
$$\kappa(n) = \kappa_o(1 + a\sin(2\pi n/1000)) \quad (25)$$

Figure 4 shows the learning curves using the above reflection coefficients. When $\kappa(n)$ are deviated from the ideal and fixed, the asynchronous updating problem does not occur, and the output error can be well reduced. However, when $\kappa(n)$ change around the ideal, the deviation should be suppressed to 0.0001 in order to achieve the output error of $-60dB$.

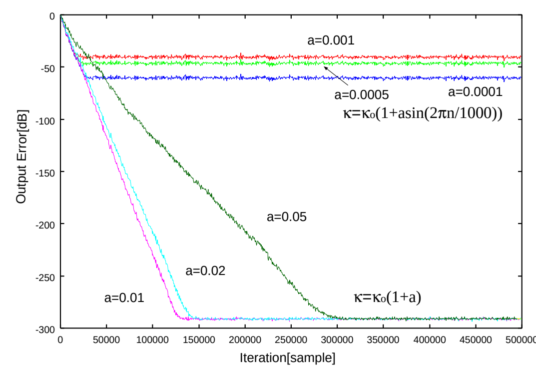

Figure 4: Learning curves for fixed and time varying $\kappa(n)$.

Another way to suppress change of $\kappa(n)$ in one iteration is to make the time constant γ in Eqs.(4) and (5) to be very close to unity. Figure 5 shows the learning curves using several values of γ in the case of the nonstationary colored signal.

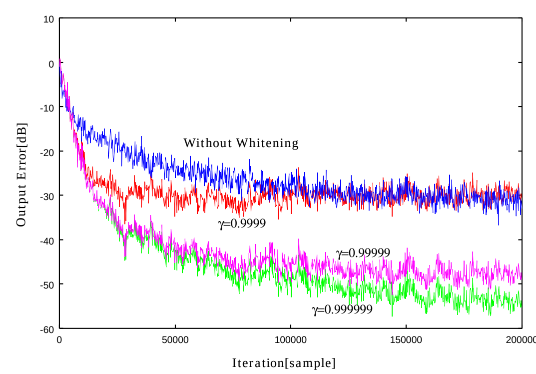

Figure 5: Learning curves for several time constant γ.

5.4 Synchronized Learning Algorithm

The synchronized learning algorithms for the linear part and the nonlinear part are simulated here. Figure6 shows the learning curves for the stationary colored signal. 'Asynchronous', 'Synchronized(L)' and 'Synchronized(L+NL)' mean the learning algorithm without synchronization of updating the reflection coefficients and the filter coefficients, the synchronized learning algorithm only for the linear part, and the synchronized learning algorithm for both the linear and nonlinear parts. The learning curve of 'Synchronized(L)' is almost the same as that of 'Asynchronous'. On the contrary, the synchronized learning algorithm for both the linear and nonlinear parts, shown with 'Synchronized(L+NL)', can improve the learning curve by 15dB.

Figure7 shows the learning curves for the nonstationary colored signal, which is generated through the 2nd-order AR model, whose pole oscillates around $\pi/4$ by $\pm 20\%$ with a period of 500 samples. In this case, also the synchronized learning algorithm for both the linear and nonlinear parts is superior to the others. In this figure, 'Synchronized(L)' is not shown, because it is almost the same as 'Asynchronous'.

Furthermore, the learning curves for another nonstationary colored signal are shown in Fig.8. The pole of the 2nd-order AR model oscillates around $\pi/4$ by $\pm 100\%$ with a period of 500 samples. In this case, the reflection coefficients change more rapidly compared to the previous example. For this reason, the convergence becomes a little slower than the previous. Still, the synchronized learning algorithm can improve the convergence property.

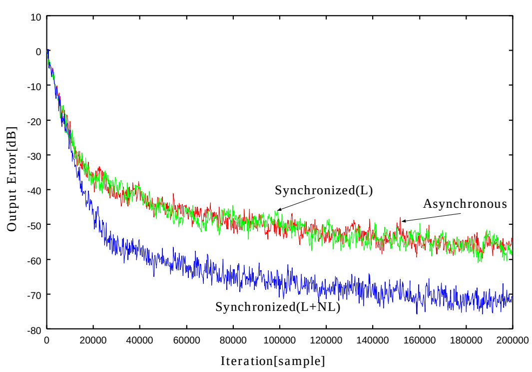

Figure 6: Learning curves for LP-AVF using stationary colored signal. Synchronized(L) and Synchronized(L+NL) means synchronized learning algorithms for linear part and linear and nonlinear parts, respectively, are used.

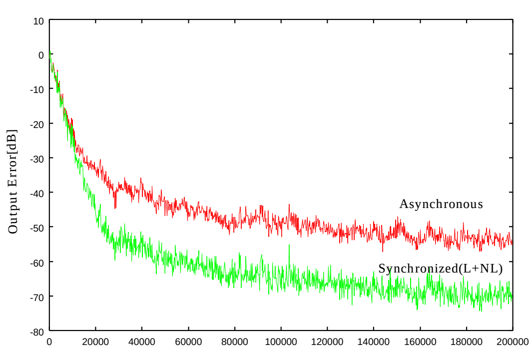

Figure 7: Learning curves for LP-AVF using nonstationary colored signal. Pole of 2nd-order AR model oscillates around $\pi/4$ by 20% with period 500 samples.

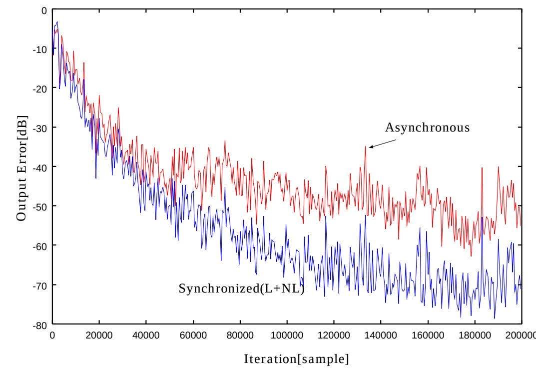

Figure 8: Learning curves for LP-AVF using nonstationary colored signal. Pole of 2nd-order AR model oscillates around $\pi/4$ by 100% with period of 500 samples.

6. CONCLUSIONS

Several kinds of whitening methods for the adaptive Volterra filter have been compared. The lattice predictor based AVF is superior to the others. Convergence property of the asynchronous LP-AVF has been analyzed. Furthermore, the synchronized learning algorithm for the nonlinear part of the LP-AVF has been proposed, and its usefulness has been confirmed through several simulations using stationary and nonstationary colored signals.

REFERENCES

[1] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.*, pp. 10-26, July 1991.

[2] 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.

[3] B.S.Nollett and D.L.Jones, "Nonlinear echo cancellation for hands-free speakerphones," *1997 IEEE Workshop on Nonlinear Signal and Image Proc*

[4] 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.

[5] 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.

[6] X.Li and W.K.Jenkins, "Computationally efficient algorithms for third order adaptive Volterra filters," *IEEE Proc. ICASSP'98*, DSP5.3, 1998.

[7] S.Haykin, *Adaptive Filter Theory*, 3rd Ed., Prentice Hall Inc., New York, 1996.

[8] N.Tokui, K.Nakayama and A.Hirano, "A synchronized learning algorithm for reflection coeffcients and tap weights in a joint lattice predictor and transversal filter", *IEEE Proc. ICASSP'2001*, Salt Lake City, pp.1472-1475, May 2001.

[9] K.Nakayama, A.Hirano and H.Kashimoto,"A lattice predictor based adaptive Volterra filter and a synchronized learning algorithm", *Proc., XII Europian Signal Processing Conference*, Vienna, Austria, pp.1585-1588, Sept. 2004.