A SYNCHRONIZED LEARNING ALGORITHM FOR REFLECTION COEFFICIENTS AND TAP WEIGHTS IN A JOINT LATTICE PREDICTOR AND TRANSVERSAL FILTER

Naoki Tokui, Kenji Nakayama and † Akihiro Hirano

Graduate School of Natural Science and Technology Kanazawa University
† Faculty of Engineering, Kanazawa University
2-40-20, Kodatsuno, Kanazawa, 920-8667, Japan
e-mail:tokui@leo.ec.t.kanazawa-u.ac.jp

ABSTRACT

In order to achieve fast convergence and less computation for adaptive filters, a joint method combining a whitening process and the NLMS algorithm is a hopeful approach. One of them is to combine a lattice predictor and a transversal filter supervised by the NLMS algorithm. However, the filter coefficient adaptation is very sensitive to the reflection coefficient fluctuation. In this paper, the reason of this instability is analyzed. The filter coefficients are updated one sample behind the reflection coefficient update. This causes large error, in other words, sensitivity of their mismatch is very high on filter characteristics. An improved learning method is proposed in order to compensate for this mismatch. The convergence property is close to that of the RLS algorithm. Computational complexity can be well reduced from that of the RLS algorithm. Simulation results using real voices demonstrate usefulness of the proposed method.

1. INTRODUCTION

As VLSI technology has been developed, adaptive filters have been applied to audio acoustic processing, control systems, telecommunication systems, and others. Among them, acoustic echo cancellation and noise cancelation are very important.

When very high-order adaptive filters are required, fast convergence and less computation for real signals are very important. The normalized LMS (NLMS) algorithm can be implemented with less computation. However, a very long time is required for convergence. On the contrary, the recursive least squires (RLS) algorithm can converge fast, at the expense of computational complexity.

One method to overcome this problem is to join a whitening process and the NLMS algorithm. The whitening process includes orthogonal transform and linear prediction [1–6]. The former method requires many frequency bands in order to realize good orthogonalization [1–3]. A lattice predictor is used in the latter method [1,4–5]. Order of the predictor is determined by that of an AR model generating the input signal, which is not so high. Therefore, this method is hopeful. However, filter coefficient adaptation is unstable due to the reflection coefficient fluctuation. Even though the lattice predictor works well, the output error cannot be well reduced. Any methods to stabilize convergence for reflection coefficient fluctuation have not been proposed.

In this paper, the learning process of the lattice predictor based NLMS algorithm is analyzed based on a relation between the reflection and filter coefficient update. A reason of the unstable adaptation is clarified. Based on this analysis, a synchronized algorithm for reflection coefficients and tap weights is proposed. This method can achieve fast convergence like the RLS algorithm with a moderate number of computations. Computer simulation using real voice will be demonstrated to confirm usefulness of the proposed learning method.

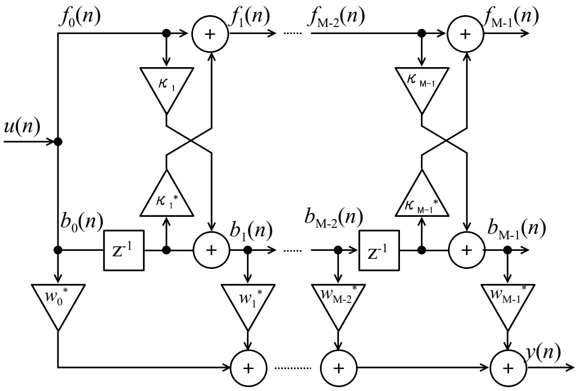


Fig. 1. Lattice-based structure for joint-process filter of order M

2. A JOINT LATTICE PREDICTOR AND NLMS TRANSVERSAL ADAPTIVE FILTER

Figure 1 shows a block diagram. The 1st-stage is the lattice predictor and the 2nd-stage is the transversal filter.

$f_m(n)$ and $b_m(n)$ are the forward and the backward prediction errors, respectively, at the mth-stage and the nth-sample. They are calculated by the following recursive formulas.

$$f_m(n) = f_{m-1}(n) + \kappa_m^*(n) b_{m-1}(n-1) \tag{1}$$

$$b_m(n) = b_{m-1}(n-1) + \kappa_m(n) f_{m-1}(n) \tag{2}$$

$$m = 1, 2, \ldots, M-1$$

$$f_0(n) = b_0(n) = u(n). \tag{3}$$

$\kappa_m(n)$ is the reflection coefficient at the mth-stage and the nth-sample.

The input signal for the transversal filter is the backward prediction error $b_m(n)$. Letting $\boldsymbol{b}(n)$, $\boldsymbol{w}(n)$ and $y(n)$ be the backward prediction error, the filter coefficients and the output, respectively, they are related by

$$\boldsymbol{b}(n) = [b_0(n), \ldots, b_{M-1}(n)]^T \tag{4}$$

$$\boldsymbol{w}(n) = [w_0(n), \ldots, w_{M-1}(n)]^T \tag{5}$$

$$y(n) = \boldsymbol{w}^H(n)\boldsymbol{b}(n). \tag{6}$$

3. ANALYSIS OF LEARNING PROCESS

3.1. Update of Reflection Coefficients

The reflection coefficient $\kappa_m(n)$ is determined so as to minimize the following prediction error.

$$J_m = E\left[|f_m(n)|^2\right] + E\left[|b_m(n)|^2\right]. \tag{7}$$

From the condition,

$$\frac{\partial J_m}{\partial \kappa_m(n)} = 0 \tag{8}$$

$\kappa_m(n)$ is given by

$$\kappa_m(n) = -\frac{2E[b_{m-1}(n-1)f^*_{m-1}(n)]}{E[|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2]}. \tag{9}$$

Furthermore, letting the numerator and the denominator be $-2\kappa_{N,m}(n)$ and $\kappa_{D,m}(n)$, respectively, they are approximately updated by

$$\begin{aligned}\kappa_{N,m}(n) \;&=\; (1-\gamma)\left(\gamma\kappa_{N,m}(n-1)\right.\\ &\qquad \left.+\, b_{m-1}(n-1)f^*_{m-1}(n)\right)\end{aligned} \tag{10}$$

$$\begin{aligned}\kappa_{D,m}(n) \;&=\; (1-\gamma)\left(\gamma\kappa_{D,m}(n-1)\right.\\ &\qquad \left.+\left(|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2\right)\right)\end{aligned} \tag{11}$$

$$1 > \gamma > 0 \tag{12}$$

3.2. Update of Filter Coefficients

The filter coefficients are updated by the NLMS algorithm as shown in

$$e(n) = d(n) - y(n) \tag{13}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\alpha}{\|\boldsymbol{b}(n)\|^2 + \delta}\boldsymbol{b}(n)e(n). \tag{14}$$

α is a step size and δ is a small positive number. The other algorithms can be also employed.

3.3. Relation between Reflection and Filter Coefficient Update

$\kappa_m(n)$ is updated at the nth-sample, and will be used at the $(n+1)$th-sample. $\boldsymbol{b}(n)$ is obtained by using $\kappa_m(n-1)$, that is the previous values. The filter coefficients are updated at the nth-sample using $\boldsymbol{b}(n)$ resulting $\boldsymbol{w}(n+1)$, which will be used at the $(n+1)$th-sample.

In Eq.(14), $e(n)$ and $\boldsymbol{b}(n)$ are obtained using $\kappa_m(n-1)$, not $\kappa_m(n)$. This means the filter coefficients $\boldsymbol{w}(n+1)$ can reduce the cost function in collaboration with $\kappa_m(n-1)$, not with $\kappa_m(n)$. However, at the $(n+1)$th-sample, $\boldsymbol{w}(n+1)$ is combined with $\kappa_m(n)$ to generate the output $y(n+1)$. This means the filter coefficient update is always one sample behind the reflection coefficient update.

This relation is held when $\kappa_m(n)$ is used to generate $\boldsymbol{b}(n)$ and $e(n)$. The index on the time axis is only shifted. An essential point is the fact that the reflection coefficients used in updating $\boldsymbol{w}(n+1)$ and in generating the filter output $y(n+1)$ together with $\boldsymbol{w}(n+1)$ are different.

These relations are shown in Fig.2.

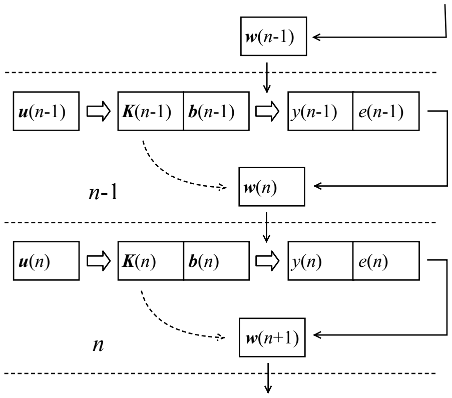


Fig. 2. Flow diagram of updating reflection coefficients and tap coefficients.

4. AN IMPROVED LEARNING METHOD

4.1. Transfer Function Representation

The transfer function of the joint adaptive filter shown in Fig.1 consists of the reflection coefficients and the filter coefficients. In this section, an equivalent transfer function in the time domain is obtained.

First, $\boldsymbol{b}(n)$ is expressed by

$$\boldsymbol{b}(n) = \boldsymbol{K}^H(n)\boldsymbol{u}(n) \tag{15}$$

$$\boldsymbol{u}(n) = [u(n), \ldots, u(n-M+1)]^T. \tag{16}$$

Here, $\boldsymbol{K}(n)$ has the following structure.

$$\boldsymbol{K}(n) = \begin{bmatrix} 1 & K_{1,2}(n) & K_{1,3}(n) & \cdots & K_{1,M}(n) \\ 0 & 1 & K_{2,3}(n) & \cdots & K_{2,M}(n) \\ \vdots & \ddots & 1 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & K_{M-1,M}(n) \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{17}$$

$$K_{1,2}(n) = \kappa^*_1(n) \tag{18}$$

$$K_{1,3}(n) = \kappa^*_2(n) \tag{19}$$

$$K_{2,3}(n) = \kappa^*_2(n)\kappa_1(n) + \kappa^*_1(n-1) \tag{20}$$

$$K_{1,4}(n) = \kappa^*_3(n) \tag{21}$$

$$K_{2,4}(n) = \kappa^*_3(n)(\kappa_1(n) + \kappa_2(n)\kappa^*_1(n-1)) + \kappa^*_2(n-1) \tag{22}$$

$$K_{3,4}(n) = \kappa^*_3(n)\kappa_2(n) + \kappa^*_2(n-1)\kappa_1(n-1) + \kappa^*_1(n-2). \tag{23}$$

Using the above expression, the filter output is given by

$$y(n) = \boldsymbol{w}^H(n)\boldsymbol{K}^H(n)\boldsymbol{u}(n). \tag{24}$$

In this expression, $\boldsymbol{w}^H(n)\boldsymbol{K}^H(n)$ represents the equivalent transfer function in the time domain. $*$ and H indicate complex conjugate and Hermitian transposition, respectively.

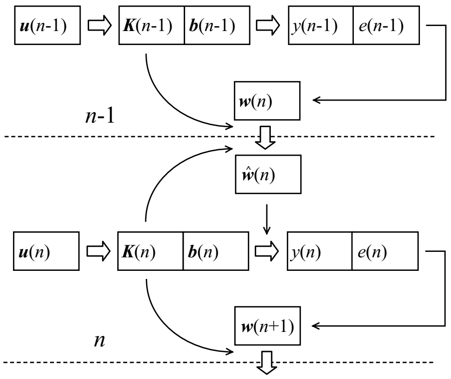


Fig. 3. Flow diagram of the modified filter coefficients

4.2. Compensation of Filter Coefficients

From the discussions in the previous section, $\boldsymbol{w}(n+1)$ is updated using $\boldsymbol{K}(n)$, therefore, the following output at the next sample can reduce the cost function.

$$\hat{y}(n+1) = \boldsymbol{w}^H(n+1)\boldsymbol{K}^H(n)\boldsymbol{u}(n+1). \qquad (25)$$

However, $\boldsymbol{K}(n)$ is updated at the $(n+1)$th-sample, then the actual output becomes

$$y(n+1) = \boldsymbol{w}^H(n+1)\boldsymbol{K}^H(n+1)\boldsymbol{u}(n+1). \qquad (26)$$

$y(n+1)$ cannot reduce the error well. In order to overcome this mismatch, the filter coefficients are modified so that the equivalent transfer function satisfies Eg.(25). That is,

$$\boldsymbol{K}(n+1)\hat{\boldsymbol{w}}(n+1) = \boldsymbol{K}(n)\boldsymbol{w}(n+1). \qquad (27)$$

From this condition, we obtain

$$\hat{\boldsymbol{w}}(n+1) = \boldsymbol{K}^{-1}(n+1)\boldsymbol{K}(n)\boldsymbol{w}(n+1). \qquad (28)$$

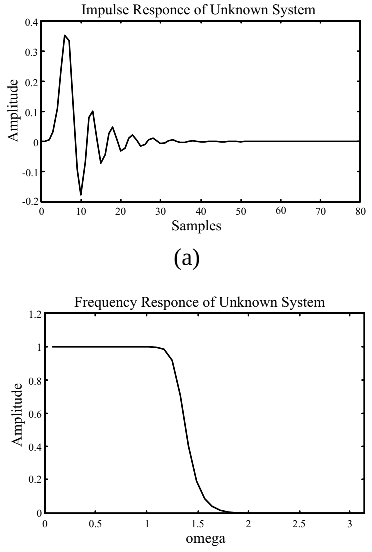
This modified filter coefficients will be used at the $(n+1)$th-sample to generate $\boldsymbol{b}(n+1)$ and $y(n+1)$. The coefficients update and modification processes are shown in Fig.3.

4.3. Computational Complexity

Equation (28) requires inverse matrix operation. However, the matrix $\boldsymbol{K}(n)$ is an upper triangle matrix, in which the diagonal elements are unity. In addition, the order of the lattice predictor depends on the input signal model, not on the unknown system. For example, in the case of voice, the signal generation process can be modeled by using approximately a 20th-order AR model. This means $\boldsymbol{K}(n)$ is also a band matrix. Taking these properties into account, the computational complexity can be reduced into $O(LM)$, where L is the order of the AR model for the input signal. In the case of acoustic echo canceller, usually $M = 1000 \sim 4000$ and $L = 20$, then the computational load is well reduced compared with that of the RLS algorithm. Table 1 lists the number of computations of the proposed and the conventional algorithms. “Conventional Lattice” indicates the joint adaptive filter, which has the same structure as the proposed method. However, the filter coefficients are updated one-sample behind the reflection coefficients.

Table 1. Comparison of LMS Algorithm with Preceding Lattice Predictor

	Multiplier	Adder
Proposed	$2ML + 5M + 9L$	$2ML + 4M + 5L$
Con. Lattice	$5M + 9L$	$4M + 5L$
NLMS	$5M$	$4M$
RLS	$3M^2 + 4M$	$2M^2 + 3M$



(a)

(b)

Fig. 4. (a)Impulse response of unknown system (b)Frequency response of unknown system.

5. SIMULATION AND DISCUSSIONS

5.1. Simulation Problem

Simulation was carried out based on system identification. An unknown system is the 12th-order IIR lowpass filter. The amplitude response and the impulse response are shown in Figs.4(a) and 4(b), respectively. The impulse response spreads over 50 samples. Therefore, the adaptive filter length M is set to 50 taps. The NLMS algorithm and the RLS algorithm are used for comparison.

5.2. Colored Input Signal

Colored signal is generated through a 2nd-order AR model with the white noise input. The amplitude response of the AR model is shown in Fig.5. In the joint adaptive filter and the NLMS algorithm, $\alpha = 1$, $\delta = 0.001$, $\gamma = 0.999$, and $\lambda = 0.95$ in the RLS algorithm.

The learning curves are shown in Fig.6. The “Conventional Lattice” cannot converge. The error is saturated at -40dB due to the one-sample delay mismatch. The learning curve of the proposed method is close to that of the RLS algorithm. This means the one-sample delay can be compensated for, at the same time, the input signal whitening by the lattice predictor is successful.

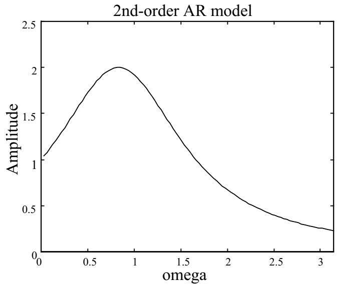


Fig. 5. Amplitude response of 2nd-order AR model.

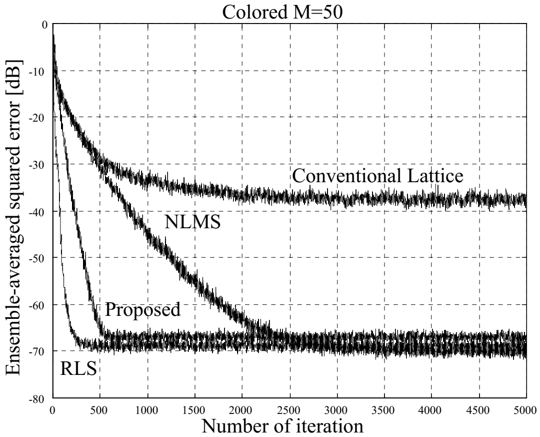


Fig. 6. Learning curves for colored noise.

5.3. Voice Input Signal

The voice signal used in the simulation is shown in Fig.7. A sampling frequency is 8kHz, then $20,000$ samples mean 2.5 seconds. Figure 8 shows the learning curves. The proposed method can catch up with the RLS at 2000 iterations. The number of iterations is the same as that of the signal samples. On the contrary, the NLMS algorithm requires 10000 samples until converge.

From these simulation results, the proposed method is useful for both stationary and nonstationary processes. Its convergence speed is close to that of the RLS.

6. CONCLUSIONS

Joint adaptive filters combining a whitening process and the NLMS transversal filter is useful to make convergence speed fast with less computation. However, the lattice predictor based adaptive filter has the instability problem. In this paper, the reason of the instability has been clarified. In the conventional method, the filter coefficients are updated one-sample behind the reflection coefficients. Sensitivity of this one-sample delay mismatch is very high. The synchronized learning algorithm has been proposed to compensate for this mismatch. The computer simulation using the colored signal and the voice signal demonstrate that the proposed method can converge fast like the RLS with less computations.

7. REFERENCES

[1] S. Haykin, “Adaptive filter theory,” PRENTICE-HALL, 3rd Edition, 1996.

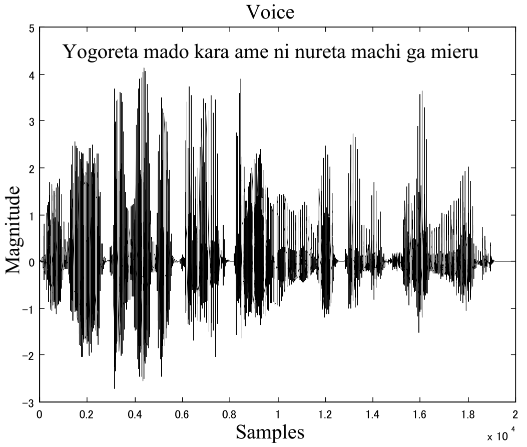


Fig. 7. Input signal of voice.

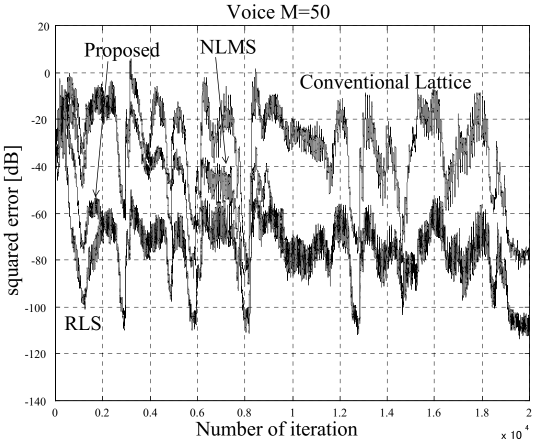


Fig. 8. Learning curves for voice.

[2] J.J. Shynk, “Frequency–domain and multirate adaptive filtering,” IEEE SP MAGAZINE, pp.14–37, Jan., 1992.

[3] F. Beaufays, “Transform–domain adaptive filters: an analytical approach, ” IEEE Trans. Signal Process., Vol. 43, No. 2, pp.422–431, Feb., 1995.

[4] J.H. Yoo, S.H. Cho and D.H. Youn, “A lattice/transversal joint(LTJ) structure for an acoustic echo canceller, ” 1995 IEEE Symposium on Circuits and Systems, Vol. 2, pp.1090–1093, 1995.

[5] S.H. Leung and C.C. Chu, “Adaptive LMS filter with lattice prefilter, ” Electron. Lett., Vol. 33, Iss. 1, pp.34–35, 2nd Jan., 1997.

[6] G. Mandyam and N. Ahmed, “The discrete laguerre transform: derivation and applications,” IEEE Trans. Signal Process., Vol. 44, No. 12, pp.2925–2931, Dec., 1996.

[7] A. Fertner, “Frequency–domain echo canceller with phase adjustment,” IEEE Trans. Circuits and Systems — II, Vol. 44, No. 10, pp.835–841, Oct., 1997.

[8] V.N. Parikh and A.Z. Baraniecki, “The use of the modified escalator algorithm to improve the performance of transform-domain LMS adaptive filters,” IEEE Trans. Signal Process., Vol. 46, No. 3, pp.625–635, Mar., 1998.