NORSIG
The 4th International Workshop on Acoustic Echo and Noise Control
Røros, Norway, 21 - 23 June 1995
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC NORWAY SECTION

Relationship between the "ES family" Algorithms and Conventional Adaptive Algorithms

Shoji Makino
NTT Human Interface Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo, 180 Japan
Tel: +81 422 59 3992, Fax: +81 422 60 7811
E-mail: makino@ac-lab.ntt.jp

ABSTRACT

We classify the conventional adaptive algorithms for acoustic echo cancellation on the basis of how well they whiten the received input signal. We also study the relationship between the conventional algorithms and the "ES family" algorithms, which were derived from the conventional ones by incorporating the knowledge that the expected variation in a room impulse response, which is normally treated as unknown in acoustic echo cancellation, decreases exponentially at the same exponential ratio that the impulse response does. We show that the NLMS, ES-NLMS, projection, ES-projection, and RLS algorithms are simplified versions of the ES-RLS algorithm.

1. INTRODUCTION

Acoustic echo cancellers are widely used for teleconferencing and hands-free telecommunication systems to overcome acoustic feedback, making conversation more comfortable.

The LMS algorithm and the NLMS (normalized LMS) algorithm require few computations, so they are widely applied to acoustic echo cancellation. However, their convergence speeds need to be increased. The projection algorithm [1] whitens the received input signal, *i.e.*, it removes the correlation between consecutive received input vectors. This process is especially effective for speech, which has a highly non-white spectrum. The RLS (recursive least-squares) algorithm, whose convergence does not depend on the input signal, is the fastest conventional adaptive algorithm.

Our previous study of the variation characteristics of a room impulse response, normally considered an "unknown system" for acoustic echo cancellation, showed that the expected variation in a room impulse response attenuates by the same exponential ratio that the impulse response does [2]. As a result, we proposed three adaptive algorithms: the ES-NLMS (exponentially weighted stepsize NLMS) algorithm [2], which reflects the variation characteristics of a room impulse response in the conventional NLMS algorithm; the ES-projection (exponentially weighted stepsize projection) algorithm [3], which reflects the variation characteristics of a room impulse response in the conventional projection algorithm; and the ES-RLS (exponentially weighted stepsize RLS) algorithm [4], which reflects the variation characteristics of a room impulse response in the conventional RLS algorithm.

The basic concept of these three algorithms is to adjust the coefficients with large errors in large steps and the coefficients with small errors in small steps. For this purpose, they use a stepsize matrix with a diagonal form. The diagonal components are time-invariant and weighted proportionally to the expected variation in the room impulse response. Consequently, they converge twice as fast as the corresponding conventional algorithm.

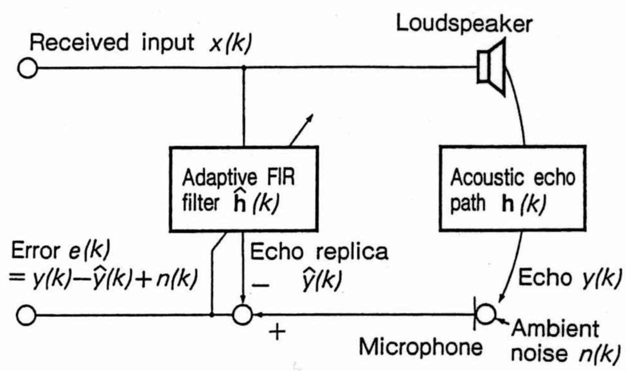

Figure 1: Configuration of an acoustic echo canceller.

In this paper, we classify the conventional adaptive algorithms on the basis of how well they whiten the received input signal and study the relationship between the "ES family" of algorithms and the conventional adaptive algorithms. We show that the NLMS, ES-NLMS, projection, ES-projection, and RLS algorithms are simplified versions of the ES-RLS algorithm.

2. ACOUSTIC ECHO CANCELLER AND CONVENTIONAL ADAPTIVE ALGORITHMS

2.1 Acoustic echo canceller

The configuration of an acoustic echo canceller is shown in Fig. 1. An echo canceller adaptively identifies impulse response $\mathbf{h}(k)$ between the loudspeaker and the microphone at discrete time k. FIR filter coefficient $\hat{\mathbf{h}}(k)$ should be a copy of $\mathbf{h}(k)$. Echo replica $\hat{y}(k)$ is created by convolving $\hat{\mathbf{h}}(k)$ with received input vector $\mathbf{x}(k)$; then $\hat{y}(k)$ is subtracted from actual echo $y(k)$ to give error $e(k)$. Adaptive FIR filter $\hat{\mathbf{h}}(k)$ is adjusted to decrease the error power at every sampling interval. The adaptive algorithm should provide fast convergence and high echo return loss enhancement (*ERLE*).

2.2 Classification of conventional adaptive algorithms

The conventional adaptive algorithms can be classified according to the extent to which the old input-output relationship is used to calculate filter coefficient vector $\hat{\mathbf{h}}(k+1)$.

$$\hat{\mathbf{h}}(k+1)^T\mathbf{x}(k) = y(k) \tag{1}$$

$$\hat{h}(k+1)^T x(k-1) = y(k-1) \tag{2}$$

$$\cdots$$

$$\hat{h}(k+1)^T x(k-p+1) = y(k-p+1) \tag{3}$$

$$\cdots$$

$$\hat{h}(k+1)^T x(0) = y(0). \tag{4}$$

Equation (1) means that if $x(k)$ is input, then filter $\hat{h}(k+1)$ outputs correct value $y(k)$ and so on for (2)-(4).

(a) NLMS algorithm

The NLMS (normalized least-mean-squares) algorithm updates filter coefficient $\hat{h}(k+1)$, which satisfies only (1). However, the number of unknowns L is larger than the number of equations, which is one here, *i.e.*, the set of equation is underdetermined. Therefore, the NLMS algorithm chooses the minimum-norm solution.

The NLMS algorithm updates filter coefficient vector $\hat{h}(k)$ according to the following equations:

$$\hat{h}(k+1) = \hat{h}(k) + \alpha \frac{e(k)}{\delta + x(k)^T x(k)} x(k) \tag{5}$$

$$e(k) = y(k) - \hat{h}(k)^T x(k) + n(k), \tag{6}$$

where

$\hat{h}(k) = [\hat{h}_1(k), \hat{h}_2(k), \cdots, \hat{h}_L(k)]^T$,
$\hat{h}_i(k)(i = 1, \cdots, L)$: coefficients of an FIR filter,
$x(k) = [x(k), x(k-1), \cdots, x(k-L+1)]^T$
: received input vector,
α : scalar stepsize ($0 < \alpha < 2$),
δ : small positive constant.

Filter coefficient vector $\hat{h}(k)$ is adjusted only in the direction of received input vector $x(k)$. Therefore, when the received input signal is a colored signal, such as speech, where consecutive vectors $x(k)$ and $x(k-1)$ are highly correlated, $\hat{h}(k)$ is hard to adjust, which results in slow convergence. With NLMS, convergence for speech input is about five times slower than that for white noise [2].

(b) Projection algorithm

The p-th order projection algorithm updates filter coefficient $\hat{h}(k+1)$, which satisfies (1)-(3) ($p < L$). Again the number of unknowns L is larger than the number of equations p, *i.e.*, the set of equations is underdetermined. Therefore, the projection algorithm also chooses the minimum-norm solution.

The *p-th* order projection algorithm updates filter coefficient vector $\hat{h}(k)$ as follows.

$$\begin{aligned}\hat{h}(k+1) &= \hat{h}(k) + \alpha X(k)[X(k)^T X(k) + \delta I]^{-1} e(k) \\ &= \hat{h}(k) + \alpha[\beta_1(k)x(k) + \beta_2(k)x(k-1) \\ &\quad + \cdots + \beta_p(k)x(k-p+1)]\end{aligned} \tag{7}$$

$$\beta(k) = [X(k)^T X(k) + \delta I]^{-1} e(k) \tag{8}$$

$$X(k) = [x(k), x(k-1), \cdots, x(k-p+1)] \tag{9}$$

$$\begin{aligned} e(k) &= y(k) - X(k)^T \hat{h}(k) + n(k) \\ &\cong [e(k), (1-\alpha)e(k-1), \cdots, (1-\alpha)^{p-1} e(k-p+1)]^T \end{aligned} \tag{10}$$

$$\beta(k) = [\beta_1(k), \beta_2(k), \cdots, \beta_p(k)]^T \tag{11}$$

$$y(k) = [y(k), y(k-1), \cdots, y(k-p+1)]^T \tag{12}$$

$$n(k) = [n(k), n(k-1), \cdots, n(k-p+1)]^T. \tag{13}$$

In the projection algorithm, filter coefficient vector $\hat{h}(k)$ is adjusted in the direction of the plane produced by $x(k)$, $x(k-1)$, $\cdots$, $x(k-p+1)$. In other words, the projection algorithm whitens the received input signal according to projection order p. Therefore, convergence can be improved for a colored input signal, where consecutive input vectors $x(k)$, $x(k-1)$, $\cdots$, $x(k-p+1)$ are highly correlated.

The second-order projection algorithm doubles the convergence speed of the NLMS for speech input. Higher-order projection algorithms achieve even faster convergence. By introducing an intermediate variable and by using the sliding windowed FTF (fast transversal filter), the computational complexity can be reduced to $2L + 20p$ multiply-add operations [5][6], where L is the number of taps and p is the projection order.

(c) RLS algorithm

The RLS (recursive least-squares) algorithm updates filter coefficient $\hat{h}(k+1)$, which satisfies all the input-output relations, (1)-(4). In the RLS case, when $L \leq k$, the number of unknowns L is smaller than the number of equations ($k+1$), *i.e.*, the set of equations is overdetermined. Therefore, the RLS algorithm chooses the least squares solution.

The RLS algorithm updates filter coefficient vector $\hat{h}(k)$ according to the following equations.

$$\hat{h}(k+1) = \hat{h}(k) + k(k)e(k) \tag{14}$$

$$k(k) = \frac{\nu^{-1} P(k) x(k)}{1 + \nu^{-1} x(k)^T P(k) x(k)} \tag{15}$$

$$P(k+1) = \nu^{-1} P(k) - \nu^{-1} k(k) x(k)^T P(k) \tag{16}$$

$$e(k) = y(k) - \hat{h}(k)^T x(k) + n(k), \tag{17}$$

where

$k(k)$: L-th order gain vector,
$P(k)$: $L \times L$ matrix,
ν : forgetting factor ($0 < \nu < 1$).

In the RLS algorithm, the received input is fully whitened, so convergence is independent of the input signal, resulting in fast convergence for all input signals.

3. "ES FAMILY" ALGORITHMS

3.1 Variation in room impulse response

An adaptive algorithm with suitable special assumptions about the characteristics of the "unknown system" to be identified can improve convergence. Although the detailed waveform is complicated, the envelope of a room impulse response (our "unknown system") attenuates exponentially, and more importantly, the expected variation in the room impulse response also attenuates by the same exponential ratio when a person moves or the environment changes [2]. Exponential attenuation ratio is common to all impulse responses in the same room. It can be derived from the reverberation time, which is determined by the acoustical conditions of the room, *i.e.*, size and absorption coefficient. We can thus estimate the exponential attenuation ratio from the room conditions, or determine it by measuring one impulse response.

3.2 Exponentially weighted stepsize NLMS (ES-NLMS) algorithm

Because of the variation in room impulse response, the expected error in each coefficient becomes progressively smaller at the same exponential ratio that the impulse response does. We incorporated this relationship into the conventional NLMS algorithm by adjusting the coefficients with large errors in large steps and the coefficients with small errors in small steps by using a stepsize matrix with a diagonal form.

The ES-NLMS algorithm is expressed as

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \mu \mathbf{A} \frac{e(k)}{\delta + \mathbf{x}(k)^T \mathbf{A}\, \mathbf{x}(k)} \mathbf{x}(k), \qquad (18)$$

where

$$\mathbf{A} = \begin{pmatrix} \alpha_1 & & & 0 \\ & \alpha_2 & & \\ & & \ddots & \\ 0 & & & \alpha_L \end{pmatrix} \qquad (19)$$

and

μ : scalar stepsize $(0 < \mu < 2)$,
$\alpha_i = \alpha_0 \gamma^{i-1} \quad (i = 1, \cdots, L)$,
γ : exponential attenuation ratio of room impulse responses $(0 < \gamma \leq 1)$.

Elements α_i are time-invariant and decrease exponentially from α_1 to α_L at the same ratio γ that impulse response $\mathbf{h}(k)$ does. This algorithm requires only $2L$ multiply-add operations (the same as the NLMS) after being modified into a practical multiple-DSP structure.

This algorithm doubles the convergence speed of the NLMS. Like the NLMS, filter coefficient vector $\hat{\mathbf{h}}(k)$ is adjusted only in the direction of received input vector $\mathbf{x}(k)$. Therefore, the ES-NLMS algorithm converges slowly when the received input signal is a colored signal, such as speech.

3.3 Exponentially weighted stepsize projection (ES-projection) algorithm

The ES-projection algorithm reflects the exponentially decaying characteristics of the room impulse response and whitens the received input signal. The second-order ES-projection algorithm converges about four times as fast as the NLMS for speech input. It is expressed as follows.

$$\begin{aligned} \hat{\mathbf{h}}(k+1) &= \hat{\mathbf{h}}(k) + \mu \mathbf{A} \mathbf{X}(k)[\mathbf{X}(k)^T \mathbf{A} \mathbf{X}(k) + \delta \mathbf{I}]^{-1} \mathbf{e}(k) \\ &= \hat{\mathbf{h}}(k) + \mu \mathbf{A}[\beta_1(k)\mathbf{x}(k) + \beta_2(k)\mathbf{x}(k-1) \\ &\quad + \cdots + \beta_p(k)\mathbf{x}(k-p+1)] \qquad (20) \end{aligned}$$

$$\beta(k) = [\mathbf{X}(k)^T \mathbf{A} \mathbf{X}(k) + \delta \mathbf{I}]^{-1} \mathbf{e}(k) \qquad (21)$$

$$\begin{aligned} \mathbf{e}(k) &= \mathbf{y}(k) - \mathbf{X}(k)^T \hat{\mathbf{h}}(k) + \mathbf{n}(k) \\ &\cong [e(k), (1-\mu)e(k-1), \cdots, (1-\mu)^{p-1} e(k-p+1)]^T. \qquad (22) \end{aligned}$$

The computational complexity of the ES-projection algorithm is reduced by introducing an intermediate variable, $\mathbf{z}(k)$:

$$\mathbf{z}(k+1) = \mathbf{z}(k) + \mu \mathbf{A}[\beta_1(k-p+1) + \cdots + \beta_p(k)]\mathbf{x}(k-p+1) \qquad (23)$$

$$\beta(k) = [\mathbf{X}(k)^T \mathbf{A} \mathbf{X}(k) + \delta \mathbf{I}]^{-1} \mathbf{e}(k) \qquad (24)$$

$$\mathbf{e}(k) \cong [e(k), (1-\mu)e(k-1), \cdots, (1-\mu)^{p-1} e(k-p+1)]^T \qquad (25)$$

$$\hat{y}(k) = \mathbf{z}(k)^T \mathbf{x}(k) + \mu \mathbf{r}(k)^T \mathbf{s}(k-1) \qquad (26)$$

$$\begin{aligned} \mathbf{r}(k) &= [\mathbf{x}(k)^T \mathbf{A} \mathbf{x}(k-1), \mathbf{x}(k)^T \mathbf{A} \mathbf{x}(k-2), \\ &\quad \cdots, \mathbf{x}(k)^T \mathbf{A} \mathbf{x}(k-p+1)]^T \qquad (27) \end{aligned}$$

$$\mathbf{s}(k-1) = \begin{pmatrix} \beta_1(k-1) \\ \beta_2(k-1) + \beta_1(k-2) \\ \cdots \\ \beta_{p-1}(k-1) + \beta_{p-2}(k-2) + \cdots + \beta_1(k-p+1) \end{pmatrix} \qquad (28)$$

Intermediate variable $\mathbf{z}(k)$ is related to impulse response replica $\hat{\mathbf{h}}(k)$:

$$\mathbf{z}(k) = \hat{\mathbf{h}}(k) - \mu \mathbf{A}[\mathbf{x}(k-1), \mathbf{x}(k-2), \cdots, \mathbf{x}(k-p+1)]\mathbf{s}(k-1). \qquad (29)$$

Furthermore, the same scheme as in [5] and [6] can be used to reduce the computational complexity.

3.4 Exponentially weighted stepsize RLS (ES-RLS) algorithm

In the ES-RLS algorithm, stepsize matrix $\mathbf{A}$ is added to matrix $\mathbf{P}_{ES}(k)$.

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \mathbf{k}(k) e(k) \qquad (30)$$

$$\mathbf{k}(k) = \frac{\mathbf{P}_{ES}(k)\mathbf{x}(k)}{1 + \mathbf{x}(k)^T \mathbf{P}_{ES}(k) \mathbf{x}(k)} \qquad (31)$$

$$\mathbf{P}_{ES}(k+1) = \mathbf{P}_{ES}(k) - \mathbf{k}(k)\mathbf{x}(k)^T \mathbf{P}_{ES}(k) + \frac{\mathbf{A}}{R} \qquad (32)$$

$$e(k) = y(k) - \hat{\mathbf{h}}(k)^T \mathbf{x}(k) + n(k), \qquad (33)$$

where

$\mathbf{P}_{ES}(k)$: $L \times L$ matrix,
R : variance of $n(k)$.

Elements $[\alpha_1, \alpha_2, ..., \alpha_L]$ of stepsize matrix $\mathbf{A}$ are not really "stepsizes" as in the ES-NLMS and ES-projection algorithms. However, as described below, these elements function as if they were stepsizes, and from the relationship between the ES-NLMS and ES-projection algorithms, we call matrix $\mathbf{A}$ a stepsize matrix.

The stepsize is known to be related to forgetting factor ν of the RLS algorithm. In fact, according to (32), when $\mathbf{A}/R$ is large compared to $\mathbf{P}_{ES}(k)$, the proportion of $\mathbf{P}_{ES}(k)$ in $\mathbf{P}_{ES}(k+1)$ becomes small. In other words, old information is forgotten quickly. Thus, the mean stepsize, $\bar{\alpha} = \frac{1}{L}\sum_{i=1}^{L} \alpha_i$, has the same role as the forgetting factor in the RLS algorithm [4].

Time-invariant $\mathbf{A}/R$ is always added in (32). In other words, an exponentially attenuating bias is always added to the diagonal elements of matrix $\mathbf{P}_{ES}(k)$. As a result, gain vector $\mathbf{k}(k)$ attenuates exponentially in (31), so filter coefficient vector $\hat{\mathbf{h}}(k)$ is adjusted by the exponentially attenuating adjustment vector in (30). Accordingly, this algorithm adjusts the coefficients with large errors in large steps and the coefficients with small errors in small steps.

4. RELATIONSHIP AMONG THE ALGORITHMS

The NLMS, projection, and RLS algorithms can be generalized when $\mu = 1$ as:

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + [\mathbf{X}(k)^T]^+ \mathbf{e}(k), \qquad (34)$$

where

$^+$: generalized inverse.

The only difference among these three algorithms is the number of columns in matrix $\mathbf{X}(k)$ and the number of elements in vector $\mathbf{e}(k)$.

When $\mathbf{A} = 0$, the ES-RLS algorithm in (30)-(33) can be regarded as an RLS algorithm (14)-(17) with a forgetting factor of 1, *i.e.*, the growing windowed RLS.

Next, for simplicity, when $p = 3$, $\mu = 1$, and $\delta = 0$ in the ES-projection algorithm (20)-(22), solving (21) we have

$$\beta_1(k) = \frac{e(k)}{\det[\mathbf{X}(k)^T\mathbf{A}\mathbf{X}(k)]}(r_{11}r_{22} - r_{12}r_{12}) \qquad (35)$$

$$\beta_2(k) = \frac{e(k)}{\det[\mathbf{X}(k)^T\mathbf{A}\mathbf{X}(k)]}(r_{12}r_{20} - r_{22}r_{10}) \qquad (36)$$

$$\beta_3(k) = \frac{e(k)}{\det[\mathbf{X}(k)^T\mathbf{A}\mathbf{X}(k)]}(r_{12}r_{10} - r_{11}r_{20}), \qquad (37)$$

where

$$\det[\mathbf{X}(k)^T\mathbf{A}\mathbf{X}(k)] = r_{00}r_{11}r_{22} + r_{01}r_{02}r_{12} + r_{01}r_{02}r_{12} - r_{00}r_{12}r_{12} - r_{01}r_{01}r_{22} - r_{02}r_{02}r_{11} \qquad (38)$$

$$r_{ij} = \mathbf{x}(k-i)^T\mathbf{A}\mathbf{x}(k-j). \qquad (39)$$

We define $c_1(k)$, $c_2(k)$, and $\mathbf{u}(k)$ as

$$c_1(k) \equiv -\frac{\beta_2}{\beta_1} = \frac{r_{22}r_{10} - r_{12}r_{20}}{r_{11}r_{22} - r_{12}r_{12}} \qquad (40)$$

$$c_2(k) \equiv -\frac{\beta_3}{\beta_1} = \frac{r_{11}r_{20} - r_{12}r_{10}}{r_{11}r_{22} - r_{12}r_{12}} \qquad (41)$$

$$\mathbf{u}(k) \equiv \mathbf{x}(k) - c_1(k)\mathbf{x}(k-1) - c_2(k)\mathbf{x}(k-2)$$
$$= [\mathbf{I} - \frac{r_{22}Q_{11} - r_{12}Q_{12}}{r_{11}r_{22} - r_{12}r_{12}} - \frac{r_{11}Q_{22} - r_{12}Q_{21}}{r_{11}r_{22} - r_{12}r_{12}}]\mathbf{x}(k), \qquad (42)$$

where

$$Q_{ij} = \mathbf{x}(k-i)\{\mathbf{A}\mathbf{x}(k-j)\}^T \qquad (43)$$

and $\mathbf{I}$ is the unit matrix. When

$$\mathbf{P}_{ES}(k) = \frac{\mathbf{A}}{R}[\mathbf{I} - \frac{r_{22}Q_{11} - r_{12}Q_{12}}{r_{11}r_{22} - r_{12}r_{12}} - \frac{r_{11}Q_{22} - r_{12}Q_{21}}{r_{11}r_{22} - r_{12}r_{12}}], \qquad (44)$$

then by substituting (31), (42), and (44) into (30), we get

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \frac{\mathbf{A}\mathbf{u}(k)}{R + \mathbf{x}(k)^T\mathbf{A}\mathbf{u}(k)}e(k). \qquad (45)$$

Equation (45) can be regarded as setting $\mu = 1$ and adding R to the denominator of the second term on the left side in the special formula for $\mu = 1$ of the ES-projection algorithm.

Furthermore, by setting $\mathbf{A} = \mathbf{I}$, we get the conventional projection algorithm. Equation (42) can be understood from the Gram-Schmidt process as

$$\mathbf{v}_1(k) \equiv \mathbf{x}(k-2) \qquad (46)$$

$$\mathbf{v}_2(k) \equiv \mathbf{x}(k-1) - \frac{\mathbf{v}_1(k)^T\mathbf{x}(k-1)}{\mathbf{v}_1(k)^T\mathbf{v}_1(k)}\mathbf{v}_1(k) \qquad (47)$$

$$\mathbf{v}_3(k) \equiv \mathbf{x}(k) - \frac{\mathbf{v}_1(k)^T\mathbf{x}(k)}{\mathbf{v}_1(k)^T\mathbf{v}_1(k)}\mathbf{v}_1(k) - \frac{\mathbf{v}_2(k)^T\mathbf{x}(k)}{\mathbf{v}_2(k)^T\mathbf{v}_2(k)}\mathbf{v}_2(k) = \mathbf{u}(k). \qquad (48)$$

The second and third terms of (48) are understood as the projection of $\mathbf{x}(k)$ onto the orthogonal vectors $\mathbf{v}_1(k)$ and $\mathbf{v}_2(k)$, respectively. Therefore, $\mathbf{u}(k)[= \mathbf{v}_3(k)]$ is orthogonal to the plane produced by $\mathbf{v}_1(k)$ and $\mathbf{v}_2(k)$, which is equivalent to the plane produced by $\mathbf{x}(k-1)$ and $\mathbf{x}(k-2)$. Thus, the correlated components of $\mathbf{x}(k-1)$ and $\mathbf{x}(k-2)$ are subtracted from $\mathbf{x}(k)$, and/or $\mathbf{u}(k)$ is orthogonal to $\mathbf{x}(k-1)$ and $\mathbf{x}(k-2)$. From (42), $c_1(k)$ and $c_2(k)$ are also understood as linear prediction filter coefficients and $\mathbf{u}(k)$ as a prediction error vector. Filter coefficient vector $\hat{\mathbf{h}}(k)$ is adjusted in the direction of the "decorrelated" or "whitened" vector $\mathbf{u}(k)$.

Next, when projection order p equals 1, we get

$$\mathbf{u}(k) = \mathbf{x}(k), \qquad (49)$$

and (45) becomes

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \frac{\mathbf{A}\mathbf{x}(k)}{R + \mathbf{x}(k)^T\mathbf{A}\mathbf{x}(k)}e(k). \qquad (50)$$

Equation (50) can be regarded as the ES-NLMS algorithm with $\mu = 1$ and $\delta = R$. Also, (50) can be introduced by setting $\mathbf{P}_{ES}(k) = \mathbf{A}/R$ in (30) and (31). Furthermore, by setting $\mathbf{A}(k) = \mathbf{I}$, we get the conventional NLMS algorithm.

5. CONCLUSIONS

Our previously proposed ES-NLMS, ES-projection, and ES-RLS algorithms for acoustic echo cancellers incorporate the fact that the impulse response attenuates exponentially at the same exponential ratio that the impulse response does. These algorithms double the convergence speed of the corresponding conventional algorithm.

We classified the conventional adaptive algorithms according to the extent to which the old input-output relationship is used to calculate the filter coefficient vector and studied the relationship between the "ES family" of algorithms and the conventional adaptive algorithms. We showed that the NLMS, projection, and RLS algorithms are one "family" of algorithms, and that the ES-NLMS, ES-projection, and ES-RLS algorithms are an extended one. Finally, the NLMS, ES-NLMS, projection, ES-projection, and RLS algorithms were shown to be simplified versions of the ES-RLS algorithm.

ACKNOWLEDGEMENTS

The author would like to thank Dr. N. Kitawaki, Mr. J. Kojima, Dr. Y. Kaneda, Mr. M. Tanaka, Mr. Y. Haneda, and Mr. S. Shimauchi for many fruitful discussions.

REFERENCES

[1] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Trans. IEICE Japan*, vol. J67-A, pp. 126-132, Feb. 1984 (in Japanese).

[2] S. Makino, Y. Kaneda, and N. Koizumi, "Exponentially weighted stepsize NLMS adaptive filter based on the statistics of a room impulse response," *IEEE Trans. Speech and Audio*, vol. 1, no. 1, pp. 101-108, Jan. 1993.

[3] S. Makino and Y. Kaneda, "Exponentially weighted stepsize projection algorithm for acoustic echo cancellers," *Trans. IEICE Japan*, vol. E75-A, no. 11, pp. 1500-1508, Nov. 1992.

[4] S. Makino and Y. Kaneda, "A new RLS algorithm based on the variation characteristics of a room impulse response," *Proc. ICASSP94*, pp. III373-III376, Apr. 1994.

[5] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "Fast projection algorithm and its step size control," *Proc. ICASSP95*, TA08.6, May 1995.

[6] S. Gay and S. Tavathia, "The fast affine projection algorithm," *Proc. ICASSP95*, RA04.1, May 1995.