IEICE TRANS. FUNDAMENTALS, VOL. E75-A, NO. 11 NOVEMBER 1992

PAPER Special Section on Acoustic System Modeling and Signal Processing

Exponentially Weighted Step-Size Projection Algorithm for Acoustic Echo Cancellers

Shoji MAKINO† *and* Yutaka KANEDA†, *Members*

SUMMARY This paper proposes a new adaptive algorithm for acoustic echo cancellers with four times the convergence speed for a speech input, at almost the same computational load, of the normalized LMS (NLMS). This algorithm reflects both the statistics of the variation of a room impulse response and the whitening of the received input signal. This algorithm, called the ESP (exponentially weighted step-size projection) algorithm, uses a different step size for each coefficient of an adaptive transversal filter. These step sizes are time-invariant and weighted proportional to the expected variation of a room impulse response. As a result, the algorithm adjusts coefficients with large errors in large steps, and coefficients with small errors in small steps. The algorithm is based on the fact that the expected variation of a room impulse response becomes progressively smaller along the series by the same exponential ratio as the impulse response energy decay. This algorithm also reflects the whitening of the received input signal, *i.e.*, it removes the correlation between consecutive received input vectors. This process is effective for speech, which has a highly non-white spectrum. A geometric interpretation of the proposed algorithm is derived and the convergence condition is proved. A fast projection algorithm is introduced to reduce the computational complexity and modified for a practical multiple DSP structure so that it requires almost the same computational load, $2L$ multiply-add operations, as the conventional NLMS. The algorithm is implemented in an acoustic echo canceller constructed with multiple DSP chips, and its fast convergence is demonstrated.

key words: *digital signal processing, adaptive filter, adaptive algorithm, echo canceller, acoustics*

1. Introduction

An acoustic echo canceller can overcome the acoustic feedback that interferes with teleconferencing and hands-free telecommunication. It adaptively identifies the transfer function between a loudspeaker and a microphone, and then produces an echo replica which is subtracted from the real echo.

Various adaptive algorithms are applicable to acoustic echo cancellers. The least-mean-squares (LMS) algorithm[1] is the simplest one. The normalized LMS (NLMS) algorithm[2], whose convergence speed is independent of input signal power, is widely used in commercial acoustic echo cancellers[3],[4].

A step size parameter, used in many gradient-type adaptive algorithms, controls the convergence rate of the filter coefficients and determines the final excess mean-squared error. Therefore, a time-varying step size, and also a time-varying matrix-form step size[5] have been introduced to obtain fast convergence in the transient state and a small excess mean-squared error in the steady state. These time-varying step-size algorithms, however, require complicated control of the step size. Furthermore, the convergence speed of these algorithms has a maximum, attained when the step sizes are unity for white noise.

The authors have proposed the ES (exponentially weighted step-size NLMS) algorithm[6],[7] which converges twice as fast as the conventional NLMS. This algorithm uses a different step size for each coefficient of an adaptive transversal filter. These step sizes are time-invariant and weighted proportional to the expected variation of a room impulse response. This algorithm uses the knowledge of a room impulse response that the expected variation of a room impulse response becomes progressively smaller along the series by the same exponential ratio as the impulse response energy decay. Some other algorithms[8],[9] based on similar ideas have also been proposed recently.

The major drawback of the LMS, NLMS, and ES algorithms is their slow convergence with a colored input signal such as speech, where consecutive input vectors are highly correlated. For example, the mean-squared error in the NLMS takes 2 seconds to converge for a white noise input signal and 10 seconds for speech, for an 8-kHz sampling rate and a filter with an order of 4000. Therefore, there is a strong need to increase the convergence speed of the LMS, NLMS, and ES. For this purpose, the projection algorithm[10]–[12] was proposed which removes the correlation between consecutive input vectors. The projection algorithm doubles the convergence speed of the NLMS for a speech input.

This paper proposes the ESP (exponentially weighted step-size projection) algorithm, which combines the advantages of the ES algorithm and the projection algorithm. This algorithm converges about four times as fast as the conventional NLMS for a speech input. A fast projection algorithm is introduced to reduce the computational complexity and modified for a practical multiple DSP structure so that it requires almost the same computational load as the conventional NLMS.

Manuscript received April 30, 1992.
Manuscript revised July 6, 1992.

† The authors are with NTT Human Interface Laboratories, Musashino-shi, 180 Japan.

This paper is organized as follows. A brief review of acoustic echo cancellers and some conventional adaptive algorithms are given in Sect. 2. The new adaptive algorithm is derived in Sect. 3, followed by practical modifications. Convergence speeds are compared in Sect. 4, including real-time experimental results. Section 5 summarizes the paper.

2. Acoustic Echo Cancellers and Conventional Adaptive Algorithms

2. 1 Configuration of an Acoustic Echo Canceller

The configuration of an acoustic echo canceller is shown in Fig. 1. The echo canceller identifies the transfer function of the acoustic echo path, i.e., the impulse response $\boldsymbol{h}(k)$ between the loudspeaker and the microphone, where $\boldsymbol{h}(k)=[h_1(k), h_2(k), \cdots]^T$ and $h_1(k)$, $h_2(k)$, $\cdots$ represent coefficients of the impulse response at discrete time k, and the superscript ^{T} represents the transpose of a vector (matrix). Since the impulse response $\boldsymbol{h}(k)$ varies as a person moves and varies with the environment, an adaptive filter $\hat{\boldsymbol{h}}(k)$ is used to identify $\boldsymbol{h}(k)$. Usually, $\hat{\boldsymbol{h}}(k)$ is a finite impulse response (FIR) filter because a wide range of simple adaptive algorithms have been proposed for an FIR filter.

An echo replica $\hat{y}(k)$ is created by convolving $\hat{\boldsymbol{h}}(k)$ with the received input vector $\boldsymbol{x}(k)$, where $\boldsymbol{x}(k)=[x(k), x(k-1), \cdots, x(k-L+1)]^T$ and L represents the number of taps. The echo replica $\hat{y}(k)$ is then subtracted from the real echo $y(k)$ to give the error $e(k)=y(k)-\hat{y}(k)+n(k)$, where $n(k)$ represents the ambient noise. In the double-talk situation where the near-end speech is added to the microphone, it is common to disable the adaptation in the echo canceller. Therefore, the near-end speech is disregarded in this paper. The adaptive FIR filter $\hat{\boldsymbol{h}}(k)$ is adjusted to decrease the error power in every sampling interval. The adaptive algorithm should provide real-time operation, fast convergence, and high echo return loss enhancement (ERLE, defined as the ratio of the real echo power to the error power excluding the ambient noise).

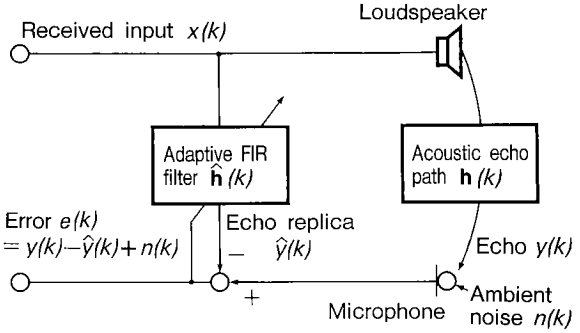


Fig. 1 Configuration of an acoustic echo canceller.

2. 2 Conventional Adaptive Algorithms

2. 2. 1 NLMS Algorithm

The normalized least-mean-squares (NLMS) algorithm[2] updates the filter coefficient vector $\hat{\boldsymbol{h}}(k)$ according to the following equations.

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\alpha\frac{e(k)}{\boldsymbol{x}(k)^T\boldsymbol{x}(k)}\boldsymbol{x}(k) \quad (1)$$

$$e(k)=y(k)-\hat{y}(k)+n(k) \quad (2)$$

$$\hat{y}(k)=\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k) \quad (3)$$

where

$\hat{\boldsymbol{h}}(k)=[\hat{h}_1(k), \hat{h}_2(k), \cdots, \hat{h}_L(k)]^T$,

$\hat{h}_i(k)$ $(i=1, \cdots, L)$: coefficients of an FIR filter,

$\boldsymbol{x}(k)=[x(k), x(k-1), \cdots, x(k-L+1)]^T$

: received input vector,

α: scalar step size $(0<\alpha<2)$.

The filter coefficient vector $\hat{\boldsymbol{h}}(k)$ is adjusted only in the direction of the received input vector $\boldsymbol{x}(k)$. Therefore, when the received input signal is a colored signal such as speech, where consecutive vectors $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ are highly correlated, $\hat{\boldsymbol{h}}(k)$ can hardly be adjusted, which results in slow convergence.

2. 2. 2 Exponentially Weighted Step-Size NLMS (ES) Algorithm

The ES (exponentially weighted step-size NLMS) algorithm[6],[7] is expressed by

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\boldsymbol{A}\frac{e(k)}{\boldsymbol{x}(k)^T\boldsymbol{x}(k)}\boldsymbol{x}(k) \quad (4)$$

where

$$\boldsymbol{A}=\begin{bmatrix}\alpha_1 & & & \\ & \alpha_2 & & 0 \\ & 0 & \ddots & \\ & & & \alpha_L\end{bmatrix} \quad (5)$$

and

$\alpha_i=\alpha_0\gamma^{i-1}$ $(i=1, \cdots, L)$,

γ: exponential attenuation ratio $(0<\gamma<1)$.

The ES algorithm uses the knowledge of a room impulse response, i.e., an impulse response attenuates exponentially and the expected variation of the impulse response also attenuates by the same exponential ratio. Incorporating this knowledge into the conventional NLMS, the ES algorithm adjusts coefficients with large errors in large steps and

coefficients with small errors in small steps. For this purpose, the ES algorithm uses a different step size for each coefficient of an adaptive transversal filter by introducing a step size matrix $\boldsymbol{A}$ with the diagonal form of Eq. (5). Elements α_i are time-invariant and decrease exponentially from α_1 to α_L with the same ratio γ as the impulse response $\boldsymbol{h}(k)$. As a results, this algorithm doubles the convergence speed of the NLMS.

The value of γ is common to all impulse responses in the same room. We can estimate the exponential attenuation ratio γ from the room conditions, or determine it by measuring one impulse response. This algorithm requires only $2L$ multiply-add operations (the same as the NLMS) after being modified for a practical multiple-DSP structure. Like the NLMS, the filter coefficient vector $\hat{\boldsymbol{h}}(k)$ is adjusted only in the direction of the received input vector $\boldsymbol{x}(k)$. Therefore, the ES algorithm converges very slowly when the received input signal is a colored signal such as speech.

2. 2. 3 Projection Algorithm

The projection algorithm[10]-[12] whitens the received input signal, i.e., it removes the correlation between consecutive received input vectors. As a result, it doubles the convergence speed of the NLMS for a speech input.

The second-order projection algorithm updates the filter coefficient vector $\hat{\boldsymbol{h}}(k)$ according to the following equation.

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\alpha[\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)] \quad (6)$$

where

α: scalar step size $(0<\alpha<2)$,

$\beta_1(k)$, $\beta_2(k)$: time-varying parameters.

Here we define $\tilde{h}(k+1)$ as

$$\tilde{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1). \quad (7)$$

Comparing Eqs. (6) and (7), the equality $\tilde{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k+1)$ holds when $\alpha=1$ in Eq. (6). The parameters $\beta_1(k)$ and $\beta_2(k)$ are determined so that $\tilde{\boldsymbol{h}}(k+1)$ satisfies both the following equations.

$$\boldsymbol{x}(k)^T\tilde{\boldsymbol{h}}(k+1)=y(k) \quad (8)$$

$$\boldsymbol{x}(k-1)^T\tilde{\boldsymbol{h}}(k+1)=y(k-1). \quad (9)$$

Combining Eqs. (6), (7), (8), and (9) gives

$$\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k)^T(k-1)=e(k) \quad (10)$$

$$\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{x}(k-1)=(1-\alpha)e(k-1). \quad (11)$$

The parameters $\beta_1(k)$ and $\beta_2(k)$ are obtained by solving simultaneous Eqs. (10) and (11). Thus, $\hat{\boldsymbol{h}}(k+1)$ is obtained by using $\beta_1(k)$, $\beta_2(k)$, and Eq. (6). The filter coefficient vector $\hat{\boldsymbol{h}}(k)$ is adjusted in an arbitrary direction on the plane produced by $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$. Therefore, convergence can be improved for a colored input signal, where consecutive input vectors $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ are highly correlated.

In particular, when $\alpha=1$, we solve the simultaneous Eqs. (10) and (11), and define $c(k)$ as

$$c(k)=-\frac{\beta_2(k)}{\beta_1(k)}=\frac{\boldsymbol{x}(k-1)^T\boldsymbol{x}(k)}{\boldsymbol{x}(k-1)^T\boldsymbol{x}(k-1)}. \quad (12)$$

We define $\boldsymbol{u}(k)$ as

$$\boldsymbol{u}(k)=\boldsymbol{x}(k)-c(k)\boldsymbol{x}(k-1)=\boldsymbol{x}(k)-\frac{\boldsymbol{x}(k-1)\boldsymbol{x}(k-1)^T}{\boldsymbol{x}(k-1)^T\boldsymbol{x}(k-1)}\boldsymbol{x}(k). \quad (13)$$

It can be understood that the correlated components of $\boldsymbol{x}(k-1)$ are subtracted from $\boldsymbol{x}(k)$, and/or that $\boldsymbol{u}(k)$ is orthogonal to $\boldsymbol{x}(k-1)$.

Combining Eqs. (6), (12), (13), and using $\boldsymbol{x}(k)^T\boldsymbol{u}(k)=\boldsymbol{u}(k)^T\boldsymbol{u}(k)$, the second-order projection algorithm is expressed as

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\frac{e(k)}{\boldsymbol{u}(k)^T\boldsymbol{u}(k)}\boldsymbol{u}(k). \quad (14)$$

Equation (14) is a special formula for $\alpha=1$ of the projection algorithm (6), (10), and (11). The filter coefficient vector $\hat{\boldsymbol{h}}(k)$ is adjusted in the direction of the "decorrelated" or "whitened" vector $\boldsymbol{u}(k)$.

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

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\alpha[\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)+\cdots+\beta_p(k)\boldsymbol{x}(k-p+1)]. \quad (15)$$

Again, we define $\tilde{\boldsymbol{h}}(k+1)$ by setting $\alpha=1$ in Eq. (15):

$$\tilde{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)+\cdots+\beta_p(k)\boldsymbol{x}(k-p+1). \quad (16)$$

The parameters $\beta_1(k)$, $\beta_2(k)$, $\cdots$, $\beta_p(k)$ are determined so that $\tilde{\boldsymbol{h}}(k+1)$ satisfies the following equations.

$$\boldsymbol{x}(k)^T\tilde{\boldsymbol{h}}(k+1)=y(k) \quad (17)$$

$$\boldsymbol{x}(k-1)^T\tilde{\boldsymbol{h}}(k+1)=y(k-1) \quad (18)$$

$$\cdots$$

$$\boldsymbol{x}(k-p+1)^T\tilde{\boldsymbol{h}}(k+1)=y(k-p+1). \quad (19)$$

The higher order projection algorithm achieves faster convergence, but it requires a larger computational cost. Here, the second-order projection algorithm is considered for practical usage.

3. New Adaptive Algorithm

3.1 Exponentially Weighted Step-Size Projection (ESP) Algorithm

The ES algorithm only reflects the statistics of the variation of a room impulse response. On the other hand, the projection algorithm only reflects a whitening of the received input signal. These are independent of each other, so we aim to reflect both of them. To incorporate the advantages of both the ES algorithm and the projection algorithm, we replace the scalar step size α in Eq. (6) by the step size matrix $\boldsymbol{A}$ of Eq. (5) with a step size matrix scaling factor μ. Then Eq. (6) becomes

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\mu\boldsymbol{A}[\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)]. \tag{20}$$

Again, by setting $\mu=1$, we define $\tilde{\boldsymbol{h}}(k+1)$ as

$$\tilde{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\boldsymbol{A}[\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)]. \tag{21}$$

The parameters $\beta_1(k)$ and $\beta_2(k)$ are determined so that $\tilde{\boldsymbol{h}}(k+1)$ satisfies Eqs. (8) and (9).

Combining Eqs. (8) and (21) yields

$$\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k)+\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k-1)=y(k). \tag{22}$$

Since

$$\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k)=\hat{y}(k), \tag{23}$$

Eq. (22) becomes

$$\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)=y(k)-\hat{y}(k)=e(k). \tag{24}$$

Similarly, combining Eqs. (9) and (21) gives

$$\boldsymbol{x}(k-1)^T\hat{\boldsymbol{h}}(k)+\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)=y(k-1). \tag{25}$$

On the other hand, combining Eqs. (20) and (21) yields

$$\hat{\boldsymbol{h}}(k+1)=\mu\tilde{\boldsymbol{h}}(k+1)+(1-\mu)\hat{\boldsymbol{h}}(k). \tag{26}$$

Since Eq. (26) also holds when k is replaced by $k-1$,

$$\hat{\boldsymbol{h}}(k)=\mu\tilde{\boldsymbol{h}}(k)+(1-\mu)\hat{\boldsymbol{h}}(k-1). \tag{27}$$

Substituting Eq. (27) for Eq. (25), we have

$$\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)=y(k-1)-\mu\boldsymbol{x}(k-1)^T\tilde{\boldsymbol{h}}(k)-(1-\mu)\boldsymbol{x}(k-1)^T\hat{\boldsymbol{h}}(k-1). \tag{28}$$

Since Eqs. (8) and (23) also hold when k is replaced by $k-1$,

$$\boldsymbol{x}(k-1)^T\tilde{\boldsymbol{h}}(k)=y(k-1) \tag{29}$$

$$\boldsymbol{x}(k-1)^T\hat{\boldsymbol{h}}(k-1)=\hat{y}(k-1). \tag{30}$$

Substituting Eqs. (29) and (30) for Eq. (28), we get

$$\begin{aligned}&\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)\\&=(1-\mu)y(k-1)-(1-\mu)\hat{y}(k-1)\\&=(1-\mu)e(k-1).\end{aligned} \tag{31}$$

The parameters $\beta_1(k)$ and $\beta_2(k)$ are obtained by solving simultaneous Eqs. (24) and (31). Thus, $\hat{\boldsymbol{h}}(k+1)$ is obtained by using $\beta_1(k)$, $\beta_2(k)$, and Eq. (20).

Accordingly, the proposed ESP (expenentially weighted step-size projection) algorithm is expressed as

$$\hat{\boldsymbol{h}}(k+1)=\hat{\boldsymbol{h}}(k)+\mu\boldsymbol{A}[\beta_1(k)\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)] \tag{32}$$

$$\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k-1)=e(k) \tag{33}$$

$$\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)=(1-\mu)e(k-1) \tag{34}$$

$$e(k)=y(k)-\hat{y}(k)+n(k) \tag{35}$$

$$\hat{y}(k)=\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k) \tag{36}$$

where

μ: step size matrix scaling factor.

Comparing the proposed ESP algorithm (32)-(34) to the conventional projection algorithm (6), (10), (11), the scalar step size α in Eq. (6) is replaced by the step size matrix $\boldsymbol{A}$ with a step size matrix scaling factor μ in Eq. (32). Then $\boldsymbol{x}(k)^T\boldsymbol{x}(k)$ and $\boldsymbol{x}(k)^T\boldsymbol{x}(k-1)$ in Eq. (10) are replaced by $\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)$ and $\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k-1)$ in Eq. (33). And $\boldsymbol{x}(k-1)^T\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)^T\boldsymbol{x}(k-1)$ in Eq. (11) are replaced by $\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)$ in Eq. (34). The scalar step size α in Eq. (11) is replaced by the step size matrix scaling factor μ in Eq. (34). Equations (32)-(34) are also derived geometrically in the Appendix. This algorithm converges when μ satisfies

$$0<\mu<2, \tag{37}$$

which is geometrically proved in the Appendix.

3.2 Reduction of Computational Complexity

The computational complexity of the ESP algo-

rithm can be reduced by introducing an intermediate variable[13], $\boldsymbol{z}(k)$, as follows.

$$\boldsymbol{z}(k+1)=\boldsymbol{z}(k)+\mu\boldsymbol{A}[\beta_1(k-1)+\beta_2(k)]\boldsymbol{x}(k-1) \tag{38}$$

$$\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k-1)=e(k) \tag{39}$$

$$\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)+\beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)=(1-\mu)e(k-1) \tag{40}$$

$$e(k)=y(k)-\hat{y}(k)+n(k) \tag{41}$$

$$\hat{y}(k)=\boldsymbol{z}(k)^T\boldsymbol{x}(k)+\mu\beta_1(k-1)\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k). \tag{42}$$

Equation (32) costs $3L$ multiplications and $2L$ additions since each element α_i is multiplied by $\mu\beta_1(k)\boldsymbol{x}(k)+\mu\beta_2(k)\boldsymbol{x}(k-1)$. On the other hand, Eq. (38) costs $2L$ multiplications and L additions. Since $\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k)$ is calculated in Eq. (39), the additional cost of Eq. (42) compared to Eq. (36) is negligible.

Incidentally, the intermediate variable $\boldsymbol{z}(k)$ is related to the impulse response replica $\hat{\boldsymbol{h}}(k)$ as

$$\boldsymbol{z}(k)=\hat{\boldsymbol{h}}(k)-\mu\boldsymbol{A}\beta_1(k-1)\boldsymbol{x}(k-1). \tag{43}$$

3. 3 Practical Modification for a Multiple DSP Structure

In a practical acoustic echo canceller constructed with multiple digital signal processor (DSP) chips, the exponential decay curve is approximated stepwise, and step size α_i is set in discrete steps with one constant value per DSP chip, as shown in Fig. 2.

Since α_i is constant in each DSP chip, Eq. (38) costs only L multiply-add operations. $\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k)$ is calculated as the sum of the value in each DSP chip, which can be derived from $\boldsymbol{x}(k-1)^T\boldsymbol{A}\boldsymbol{x}(k-1)$ by adding the new value and subtracting the oldest value in each DSP chip. The same scheme can be used for $\boldsymbol{x}(k)^T\boldsymbol{A}\boldsymbol{x}(k-1)$. Therefore, the simultaneous Eqs. (39) and (40) can be solved with small computational cost. Equation (42) costs only L multiply-add operations. Thus, the practical modification allows the proposed algorithm to have almost the same computational load, $2L$ multiply-add operations, as the conventional NLMS.

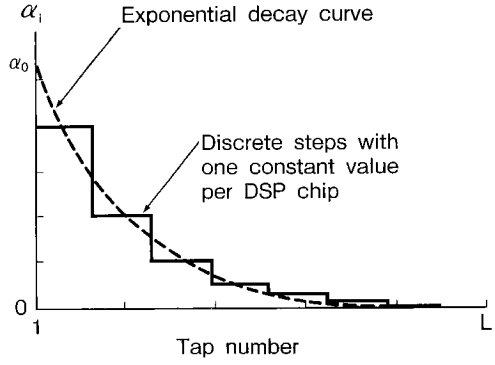
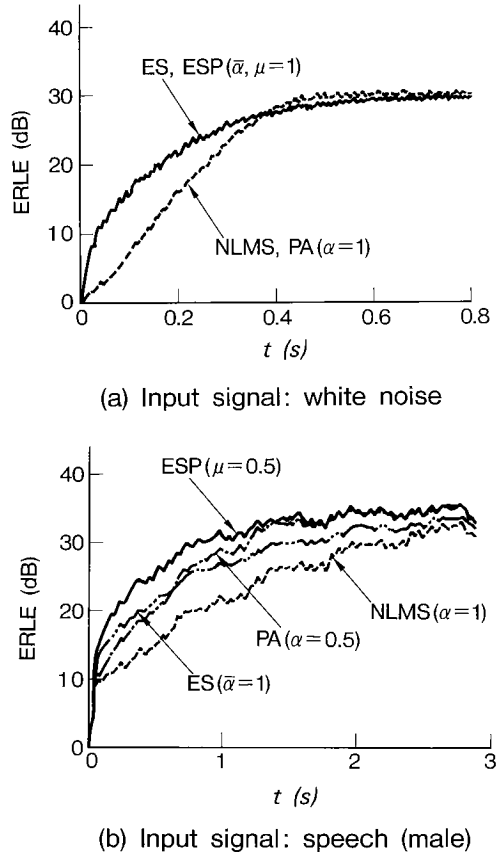


Fig. 2 Step size α_i of matrix $\boldsymbol{A}$ when α_i is set in discrete steps with one constant value per DSP chip.

4. Convergence Comparison

4. 1 Computer Simulations

Computer simulation results on the convergence of the ERLE in ESP algorithm are shown in Fig. 3. The impulse response used in the computer simulation was measured in a room with a reverberation time of 150 ms. The speaker-microphone distance was 1 m. The number of taps was 512 and the sampling frequency was 8 kHz. The filter coefficients were initially set to zero. The received inputs were (a) white noise



(b) Input signal: speech (male)

Fig. 3 Computer simulation results on ERLE convergence. The number of taps is 512 and sampling frequency is 8 kHz. Ambient noise with a fixed SNR of 30 dB for white noise and 35 dB for speech is added. The speaker-microphone distance is 1 m. Room reverberation time at 500 Hz is 150 ms.

and (b) speech. Ambient noise with a fixed SNR of 30 dB for white noise and 35 dB for speech was added. Each curve is the average of 50 independent results.

ERLE has been widely adopted and is defined by

$$ERLE = 10 \log_{10} \frac{p_y}{p_e} \quad \text{(dB)} \tag{44}$$

where

$p_y = E[y(k)^2]$,

$p_e = E[\tilde{e}(k)^2] = E[(y(k) - \bar{y}(k))^2]$,

$E[\cdot]$: statistical expectation.

The signal powers p_y and p_e were estimated from 10 data samples for white noise and 256 for speech in the computer simulations.

For speech, the scalar step size was 1.0 for the NLMS and 0.5 for the projection algorithm, and the mean step size was 1.0 for the ES and the step size matrix scaling factor was 0.5 for the ESP. For white noise, they all were 1.0. They were set so that the steady-state ERLEs were almost equal.

With a white noise input [Fig. 3(a)], the convergence speed of the projection algorithm (PA) was almost equal to that of the NLMS, and that of the ESP was almost equal to that of the ES. Evidently, the whitening effect of the projection algorithm produces no benefit for the white noise input signal. The ES and ESP algorithms both reach an ERLE of 20 dB twice as fast as the NLMS and projection algorithms. This is due to the echo path knowledge of the ES and the ESP algorithms. With a speech input [Fig. 3(b)], the ES and projection algorithms both reach an ERLE of 20 dB twice as fast as the NLMS. This is due to the echo path knowledge of the ES algorithm and the whitening effect of the projection algorithm, respectively. The ESP algorithm reaches an ERLE of 20 dB four times as fast as the NLMS. This is due to the combination of the independent effects of the ES and the projection algorithms.

4. 2 Real-Time Experiments

Real-time experiments were performed with the ESP algorithm implemented in an acoustic echo canceller constructed with multiple DSP chips[14]. The subband technique was used to separate the 7-kHz frequency range into two bands, each sampled at 8 kHz. The number of taps was 3072 in each band. The received inputs were (a) white noise and (b) speech. For speech, the scalar step size was 1.0 for the NLMS and 0.5 for the projection algorithm, and the mean step size was 1.0 for the ES and the step size matrix scaling factor was 0.5 for the ESP. For white noise, they all were 1.0. They were set so that steady-state ERLEs were almost equal. The speaker-microphone distance was 2.5 m and the reverberation time of the room was 300 ms.

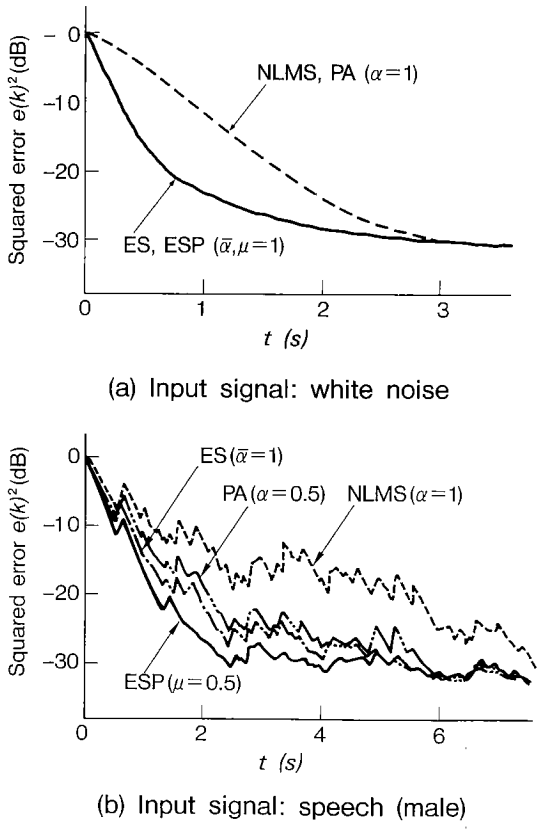


Fig. 4 Real-time experimental results on squared error level convergence using a two-subband acoustic echo canceller constructed with multiple DSP chips. The number of taps is 3072 and sampling frequency is 8 kHz in each band. The speaker-microphone distance is 2.5 m. Room reverberation time at 500 Hz is 300 ms.

Figure 4 shows the real-time measurements of squared error level convergence. The results were almost the same as the computer simulations in Sect. 4. 1. With a white noise input [Fig. 4(a)], the squared error levels of the ES and ESP decayed to -20 dB twice as fast as the NLMS and the projection algorithm (PA). With a speech input [Fig. 4(b)], the squared error levels of the ES and projection algorithms both decayed to -20 dB twice as fast as the NLMS. The squared error level of the ESP algorithm decayed to -20 dB four times as fast as the NLMS. The steady-state ERLEs of all algorithms were over 30 dB in the 7-kHz frequency range.

Thus, the ESP algorithm is superior to the NLMS algorithm and improves the convergence of practical acoustic echo cancellers with a speech input.

5. Conclusions

A new adaptive algorithm has been developed for use in acoustic echo cancellers. This algorithm uses the knowledge of a room impulse response and whitens a

colored input signal like speech, where consecutive input vectors are highly correlated. This algorithm adjusts each filter coefficient by a different step size. These step sizes α_i are time-invariant and are set proportional to the expected variation of a room impulse response. This algorithm removes the correlation between consecutive received input vectors and speeds up the convergence compared to the NLMS for a speech input. By introducing a fast algorithm and modifying α_i in a practical multiple-DSP structure, we reduce the amount of computation of the proposed algorithm to almost $2L$ multiply-add operations, which is the same as the conventional NLMS. A geometric interpretation of the proposed algorithm is derived and the convergence condition is proved. The algorithm has been implemented in an acoustic echo canceller constructed with multiple DSP chips. Real-time experiments in a room showed that this algorithm converges faster than the conventional NLMS: about two times as fast for a white noise input signal, and four times as fast for speech.

References

(1) Widrow B. and Stearns S.: "Adaptive Signal Processing", Englewood Cliffs, NJ: Prentice-Hall(1985).
(2) Nagumo J. and Noda A.: "A learning method for system identification", IEEE Trans. Automat. Contr., **AC-12**, 3, pp. 282-287(June 1967).
(3) Itoh Y., Maruyama Y., Furuya N., and Araseki T.: "An acoustic echo canceller for teleconference", Proc. ICC85, pp. 1498-1502(June 1985).
(4) Oikawa H., Minami S., and Saeki T.: "A new echo canceller realized by high performance digital signal processor", Proc. ISCAS88, pp. 1329-1332(June 1988).
(5) Harris R., Chabries D., and Bishop F.: "A variable step (VS) adaptive filter algorithm", IEEE Trans. Acoust., Speech, Signal Processing, **ASSP-34**, 2, pp. 309-316 (April 1986).
(6) Makino S. and Kaneda Y.: "Acoustic echo canceller algorithm based on the variation characteristics of a room impulse response", Proc. ICASSP90, pp. 1133-1136(April 1990).
(7) Makino S., Kaneda Y., and Koizumi N.: "Exponentially weighted step-size NLMS adaptive filter based on the statistics of a room impulse response", to be published in IEEE Trans. Speech and Audio, **1**, 1(Jan. 1993).
(8) Maruyama Y.: "An echo canceller using weighting adaptation", Proc. 1989 IEICE Spring Conf., B-557.
(9) Furukawa H., Kanamori T., Ibaraki S., Naono H. and Tanaka K.: "Handsfree telephone with echo cancellers which require a little computation", Proc. IEICE Spring Conf. '92, SA-7-8.
(10) Ozeki K. and Umeda T.: "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties", Trans. IEICE, **J67-A**, pp. 126-132(Feb. 1984).
(11) Hinamoto T. and Maekawa S.: "Extended theory of learning identification", Trans. IEE Japan, **95**, pp. 227-234(Oct. 1975).
(12) Yasukawa H., Furukawa I., and Ishiyama Y.: "Acoustic echo control for high quality audio teleconferencing", Proc. ICASSP89, pp. 2041-2044(May 1989).
(13) Maruyama Y.: "A fast method of projection algorithm", Proc. 1990 IEICE Spring Conf., B-744.
(14) Kaneko T., Yamauchi H. and Iwata A.: "A 50 ns floating-point signal processor VLSI", Proc. ICASSP86, pp. 401-404(April 1986).

Appendix: Geometric Interpretation of the Proposed Algorithm

Let us assume that the room impulse response $\boldsymbol{h}(k)$ takes a constant value $\boldsymbol{h}_0$ for times $k-1$, k, and $k+1$, and that the noise is disregarded. The second-order ESP algorithm updates the filter coefficient vector $\hat{\boldsymbol{h}}(k)$ in such a way that $\hat{\boldsymbol{h}}(k+1)$ satisfies the following equations.

$$\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k+1)=y(k)=\boldsymbol{x}(k)^T\boldsymbol{h}_0 \quad (A\cdot1)$$

$$\boldsymbol{x}(k-1)^T\hat{\boldsymbol{h}}(k+1)=y(k-1)=\boldsymbol{x}(h-1)^T\boldsymbol{h}_0. \quad (A\cdot2)$$

Using a diagonal matrix $\boldsymbol{B}$, which satisfies

$$\boldsymbol{BB}=\boldsymbol{A}, \quad (A\cdot3)$$

Eqs. (A·1) and (A·2) can be rewritten as

$$[\boldsymbol{Bx}(k)]^T\boldsymbol{B}^{-1}[\boldsymbol{h}_0-\hat{\boldsymbol{h}}(k+1)]=0 \quad (A\cdot4)$$

$$[\boldsymbol{Bx}(k-1)]^T\boldsymbol{B}^{-1}[\boldsymbol{h}_0-\hat{\boldsymbol{h}}(k+1)]=0. \quad (A\cdot5)$$

Equations (A·4) and (A·5) mean that $\boldsymbol{B}^{-1}[\boldsymbol{h}_0-\hat{\boldsymbol{h}}(k+1)]$ is orthogonal to both $\boldsymbol{Bx}(k)$ and $\boldsymbol{Bx}(k-1)$. This can be understood geometrically using Fig. A·1.

In Fig. A·1,

$$\boldsymbol{B}^{-1}[\boldsymbol{h}_0-\hat{\boldsymbol{h}}(k)]=\overrightarrow{OC} \quad (A\cdot6)$$

$$\boldsymbol{Bx}(k)=\overrightarrow{OD} \quad (A\cdot7)$$

$$\boldsymbol{Bx}(k-1)=\overrightarrow{OE} \quad (A\cdot8)$$

$$\boldsymbol{B}^{-1}[\boldsymbol{h}_0-\hat{\boldsymbol{h}}(k+1)]=\overrightarrow{FC} \quad (A\cdot9)$$

where point C is orthogonally projected to point F on

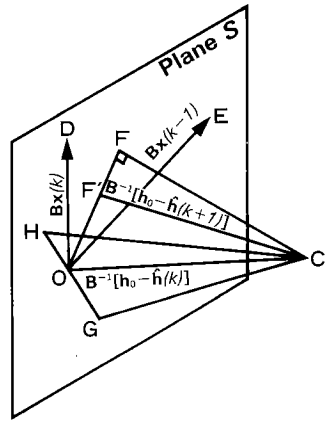


Fig. A·1 Geometric interpretation of the proposed algorithm.

the plane S produced by $\boldsymbol{Bx}(k)$ and $\boldsymbol{Bx}(k-1)$. Then

$$\begin{aligned}\overrightarrow{OF} &= \beta_1(k)\overrightarrow{OD} + \beta_2(k)\overrightarrow{OE} \\ &= \beta_1(k)\boldsymbol{Bx}(k) + \beta_2(k)\boldsymbol{Bx}(k-1). \quad (A\cdot 10)\end{aligned}$$

On the other hand, the second-order ESP algorithm updates the filter coefficient vector $\hat{\boldsymbol{h}}(k)$ according to the following equation.

$$\begin{aligned}\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu[&\beta_1(k)\boldsymbol{BBx}(k) \\ &+ \beta_2(k)\boldsymbol{BBx}(k-1)]. \quad (A\cdot 11)\end{aligned}$$

This can be rewritten as

$$\begin{aligned}\boldsymbol{B}^{-1}\hat{\boldsymbol{h}}(k+1) &= \boldsymbol{B}^{-1}\hat{\boldsymbol{h}}(k) + \mu[\beta_1(k)\boldsymbol{Bx}(k) \\ &\qquad + \beta_2(k)\boldsymbol{Bx}(k-1)] \\ &= \boldsymbol{B}^{-1}\hat{\boldsymbol{h}}(k) + \mu\overrightarrow{OF} \\ &= \boldsymbol{B}^{-1}\hat{\boldsymbol{h}}(k) + \overrightarrow{OF'}, \quad (A\cdot 12)\end{aligned}$$

where $\overrightarrow{OF'} = \mu\overrightarrow{OF}$. Using $\boldsymbol{h}_0$,

$$\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k+1)] = \boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k)] - \overrightarrow{OF'} \quad (A\cdot 13)$$

$$\overrightarrow{F'C} = \overrightarrow{OC} - \overrightarrow{OF'}. \quad (A\cdot 14)$$

Obviously, $\overrightarrow{F'C}$ is smaller than OC when $0<\mu<2$, and is smallest when $\mu=1$. Since $\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k)]$ is monotonically non-increasing when

$$0<\mu<2, \quad (A\cdot 15)$$

it converges. Since $\boldsymbol{B}^{-1}$ has full rank, the coefficient error vector $\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k)$ also converges.

Subtracting $\boldsymbol{x}(k)^T\hat{\boldsymbol{h}}(k) = \hat{y}(k)$ from both sides of Eq. (A·1) and using $\boldsymbol{B}$ and $\boldsymbol{B}^{-1}$, we get

$$\begin{aligned}&[\boldsymbol{Bx}(k)]^T\boldsymbol{B}^{-1}[\tilde{\boldsymbol{h}}(k+1) - \hat{\boldsymbol{h}}(k)] \\ &= [\boldsymbol{Bx}(k)]^T\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k)] \\ &= y(k) - \hat{y}(k) \quad (A\cdot 16)\end{aligned}$$

$$(\overrightarrow{OD}, \overrightarrow{OF}) = (\overrightarrow{OD}, \overrightarrow{OC}) = e(k). \quad (A\cdot 17)$$

Multiplying both sides of (A·11) by $\boldsymbol{x}(k)^T$ gives

$$\begin{aligned}&\mu[\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{BBx}(k) \\ &\quad + \beta_2(k)\boldsymbol{x}(k)^T\boldsymbol{BBx}(k-1)] \\ &= \boldsymbol{x}(k)^T[\hat{\boldsymbol{h}}(k+1) - \hat{\boldsymbol{h}}(k)] \\ &= [\boldsymbol{Bx}(k)]^T\boldsymbol{B}^{-1}[\hat{\boldsymbol{h}}(k+1) - \hat{\boldsymbol{h}}(k)] \\ &= (\overrightarrow{OD}, \overrightarrow{OF'}) \\ &= (\overrightarrow{OD}, \mu\overrightarrow{OF}) \\ &= \mu e(k). \quad (A\cdot 18)\end{aligned}$$

Thus

$$\begin{aligned}&\beta_1(k)\boldsymbol{x}(k)^T\boldsymbol{BBx}(k) \\ &\quad + \beta_2(k)\boldsymbol{x}(k)^T\boldsymbol{BBx}(k-1) = e(k). \quad (A\cdot 19)\end{aligned}$$

Again, in Fig. A·1,

$$\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k-1)] = \overrightarrow{GC} \quad (A\cdot 20)$$

$$\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \tilde{\boldsymbol{h}}(k)] = \overrightarrow{HC} \quad (A\cdot 21)$$

where point C is orthogonally projected to point H on the plane produced by $\boldsymbol{Bx}(k-1)$ and $\boldsymbol{Bx}(k-2)$. Since Eq. (A·1) also holds when k is replaced by $k-1$,

$$\begin{aligned}\boldsymbol{x}(k-1)^T\tilde{\boldsymbol{h}}(k) &= y(k-1) \\ &= \boldsymbol{x}(k-1)^T\boldsymbol{h}_0. \quad (A\cdot 22)\end{aligned}$$

Subtracting $\boldsymbol{x}(k-1)^T\hat{\boldsymbol{h}}(k-1) = \hat{y}(k-1)$ from both sides of Eq. (A·22) and using $\boldsymbol{B}$ and $\boldsymbol{B}^{-1}$, we get

$$\begin{aligned}&[\boldsymbol{Bx}(k-1)]^T\boldsymbol{B}^{-1}[\tilde{\boldsymbol{h}}(k) - \hat{\boldsymbol{h}}(k-1)] \\ &= [\boldsymbol{Bx}(k-1)]^T\boldsymbol{B}^{-1}[\boldsymbol{h}_0 - \hat{\boldsymbol{h}}(k-1)] \\ &= y(k-1) - \hat{y}(k-1) \quad (A\cdot 23)\end{aligned}$$

$$(\overrightarrow{OE}, \overrightarrow{GH}) = (\overrightarrow{OE}, \overrightarrow{GC}) = e(k-1). \quad (A\cdot 24)$$

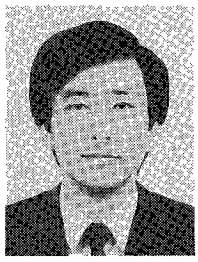
Multiplying both sides of Eq. (A·11) by $\boldsymbol{x}(k-1)^T$ yields

$$\begin{aligned}&\mu[\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{BBx}(k) \\ &\quad + \beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{BBx}(k-1)] \\ &= \boldsymbol{x}(k-1)^T[\hat{\boldsymbol{h}}(k+1) - \hat{\boldsymbol{h}}(k)] \\ &= [\boldsymbol{Bx}(k-1)]^T\boldsymbol{B}^{-1}[\hat{\boldsymbol{h}}(k+1) - \hat{\boldsymbol{h}}(k)] \\ &= (\overrightarrow{OE}, \overrightarrow{OF'}) \\ &= (\overrightarrow{OE}, \mu\overrightarrow{OF}) \\ &= (\overrightarrow{OE}, \mu[\overrightarrow{OH} + \overrightarrow{HC} + \overrightarrow{CF}]) \\ &= (\overrightarrow{OE}, \mu[(1-\mu)\overrightarrow{GH} + \overrightarrow{HC} + \overrightarrow{CF}]) \\ &= \mu(1-\mu)e(k-1). \quad (A\cdot 25)\end{aligned}$$

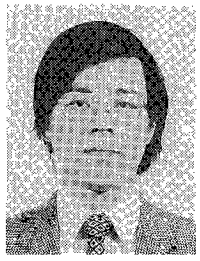
Thus

$$\begin{aligned}&\beta_1(k)\boldsymbol{x}(k-1)^T\boldsymbol{BBx}(k) \\ &\quad + \beta_2(k)\boldsymbol{x}(k-1)^T\boldsymbol{BBx}(k-1) \\ &= (1-\mu)e(k-1). \quad (A\cdot 26)\end{aligned}$$

Equations (A·11), (A·19) and (A·26) give the ESP algorithm. Incidentally, the geometric interpretation of the conventional projection algorithm is obtained by setting $\boldsymbol{B}=\boldsymbol{I}$, the unit matrix.



Shoji Makino was born in Nikko, Japan, on June 4, 1956. He received the B.E. and M.E. degrees in mechanical engineering from Tohoku University, Sendai, Japan, in 1979 and 1981. He joined the Electrical Communication Laboratory of Nippon Telegraph and Telephone Corporation (NTT) in 1981. Since then, he has been engaged in research on electroacoustic transducers and acoustic echo cancellers. He is now a Senior Research Engineer at the Speech and Acoustics Laboratory of the NTT Human Interface Laboratories. His research interests include acoustic signal processing, and adaptive filtering and its applications. He is a member of the IEEE, the Acoustical Society of Japan.



Yutaka Kaneda was born in Osaka, Japan, on February 20, 1951. He received the B.E., M.E. and Doctor of Engineering degrees from Nagoya University, Nagoya, Japan, in 1975, 1977 and 1990. In 1977, he joined the Electrical Communication Loboratory of Nippon Telegraph and Telephone Corporation (NTT), Musashino, Tokyo, Japan. He has since been engaged in research on acoustic signal processing. He is now a Senior Research Engineer at the Speech and Acoustics Laboratory of the NTT Human Interface Laboratories. His recent research interests include microphone array processing, adaptive filtering and sound field control. He received the IEEE ASSP Senior Award in 1990 for an article on inverse filtering of room acoustics, and paper awards from the Acoustical Society of Japan in 1990 and 1992 for articles on adaptive microphone arrays and active noise control. Dr. Kaneda is a member of the Acoustical Society of Japan, the Acoustical Society of America.