From (25) and (26), it can be shown that

$$\mathbf{e}_l(0) = \mathbf{e}_l(1) = \mathbf{0} \tag{27}$$

$$\mathbf{e}_r^T(0)\mathbf{e}_r(0) = \mathbf{I}_{M-\beta},\ \mathbf{e}_d^T(0)\mathbf{e}_d(0) = \mathbf{I}_\beta. \tag{28}$$

On the other hand, the linear-phase property in (4) shows that $\mathbf{e}_d(0)$ and $\mathbf{e}_r(0)$ consist of symmetric (anti-symmetric) rows. Hence, the initial matrix $\mathbf{E}_1(z)$ can be expressed as

$$\mathbf{E}_1 = \begin{pmatrix} \mathbf{0} & \mathbf{S}_{M-\beta} \\ \mathbf{S}_\beta & \mathbf{0} \end{pmatrix} \tag{29}$$

where $\mathbf{S}_\beta$ and $\mathbf{S}_{M-\beta}$ are matrices with symmetric (anti-symmetric) rows in (2) and (3). From (27), it is noted that the first and last β taps of the filters in $\mathbf{S}_{M-\beta}$ in (29) that have length $M+\beta$ are zeros.

F. Algorithm and Discussions

Given a linear-phase paraunitary filter bank, the polyphase matrix $\mathbf{E}_K(z)$ can be gradually factored as the form $\mathbf{E}_K(z) = \mathbf{AB\Theta}(z)\mathbf{E}_{K-1}(z)$ in (22) and the initial matrix $\mathbf{E}_1$ in (29). Conversely, recursively cascading the lattice structures $\mathbf{AB\Theta}(z)$ in (22) to the initial matrix $\mathbf{E}_1$ in (29) spans the class of filter banks with lengths $N_i = k_iM + \beta$, $0 \le k_i \le K$, $0 \le \beta \le M-1$. [After each stage of the filter length changes in (22), the polyphase matrix needs to be rearranged in the form of (5) through row-wise permutations.] The permissible distributions of filter length and symmetry polarity among the channels can be determined in the design process. In the matrix $\mathbf{E}_1$ in (29), there are $\lceil \beta/2 \rceil$ symmetric and $\lfloor \beta/2 \rfloor$ antisymmetric filters in $\mathbf{S}_\beta$ that have length β, and $\lceil (M-\beta)/2 \rceil$ symmetric and $\lfloor (M-\beta)/2 \rfloor$ antisymmetric filters in $\mathbf{S}_{M-\beta}$ that have length $M+\beta$. As the factorizations in (14), (18), (20), and (21) do not change the numbers of symmetric and antisymmetric filters, it is shown that a higher order filter bank $\mathbf{E}_K(z)$ will have $\lceil \beta/2 \rceil + \lceil (M+\beta)/2 \rceil$ symmetric and $\lfloor \beta/2 \rfloor + \lfloor (M-\beta)/2 \rfloor$ antisymmetric filters, as the initial matrix $\mathbf{E}_1$. Moreover, as the number (2ς) of filters in the factorization in (18) is even and the changes of filter length in (20) and (21) is $2M$, the sum $\sum_{i=0}^{M-1} k_i$ will be even or odd as $\mathbf{E}_1$, i.e., $\sum_{i=0}^{M-1} k_i$ will be even or odd as $(M-\beta)$. Based on the new algorithm, a nine-channel filter bank with a length distribution (32, 32, 32, 32, 32, 32, 11, 11, 11) and alternative symmetric and antisymmetric property is shown in Fig. 3 and Table I.

IV. Conclusion

In this correspondence, we presented an algorithm for designing a class of linear-phase paraunitary filter banks with generalized lengths $k_iM + \beta$, $0 \le i \le M-1$, and $0 \le \beta \le M-1$. A number of properties in this class of filter banks were investigated, and the lattice factorizations were achieved through a succession of "length reduction" operations. The permissible distributions of filter length and symmetry polarity among the channels can be determined in the design process. The discussions for filter banks with even and odd number of channels are integrated in the generalized algorithm.

References

[1] D. Esteban and C. Galand, "Application of quadrature mirror filters to split-band voice coding schemes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.,* Hartford, CT, May 1977, pp. 191–195.

[2] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* Denver, CO, Apr. 1980, pp. 291–294.

[3] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[4] P. P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques," *IEEE Acoust., Speech, Signal Processing Mag.,* pp. 4–21, July 1987.

[5] M. Vetterli and D. Le Gall, "Perfect reconstruction FIR filter banks: Some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 1057–1071, July 1989.

[6] T. Q. Nguyen and P. P. Vaidyanathan, "Structures for M-channel perfect-reconstruction FIR QMF banks which yield linear-phase analysis filters," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 433–446, Mar. 1990.

[7] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[8] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 969–978, June 1990.

[9] A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen, "Linear phase paraunitary filter banks: Theory, factorizations and applications," *IEEE Trans. Signal Processing,* vol. 41, pp. 3480–3496, Dec. 1993.

[10] R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, "The GenLOT: Generalized linear phase lapped orthogonal transform," *IEEE Trans. Signal Processing,* vol. 44, pp. 497–507, Mar. 1996.

[11] C. W. Kok, M. Ikehara, and T. Q. Nguyen, "Structures and factorizations of linear-phase paraunitary filter banks," in *Proc. IEEE Int. Symp. Circuits Syst.,* June 1997, pp. I.365–I.368.

[12] T. Tran and T. Q. Nguyen, "On arbitrary-length M-channel linear-phase FIR filter banks," in *Proc. Asilomar Conf. Syst., Signals, Comput.* Monterey, CA, Nov. 1995.

Conditions for Convergence of a Delayless Subband Adaptive Filter and Its Efficient Implementation

Kiyoshi Nishikawa and Hitoshi Kiya

***Abstract*—This correspondence considers the convergence characteristics of the delayless subband adaptive digital filters (ADF's) proposed by Morgan and Thi. We derive a formula for the step-size parameter to ensure the convergence of ADF's and show that the derived formula enables self-adjusting of its value to ensure convergence regardless of the type of analysis filters employed or characteristics of the input signals. Consideration on the efficient implementation of the structure using short-length analysis filters is given through the results of simulations.**

***Index Terms*—Adaptive signal processing, identification.**

I. Introduction

In this correspondence, we consider the convergence characteristics of the subband adaptive digital filters (ADF's) implemented in the delayless architecture proposed by Morgan and Thi [1]. The conditions for convergence of ADF's are given, and a formula is derived that enables self-adjustment of its value to ensure convergence.

Subband ADF's are considered to be one of several possible solutions to improve the performance of adaptive signal processing [2]–[7]. However, there are problems with the conventional structure of the subband adaptive filters. One of the problems, and probably

Manuscript received February 15, 1997; revised November 30, 1997. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ahmed Tewfik.

The authors are with the Department of Electrical Engineering, Faculty of Engineering, Tokyo Metropolitan University, Tokyo, Japan 192-0397.

Publisher Item Identifier S 1053-587X(98)02693-2.

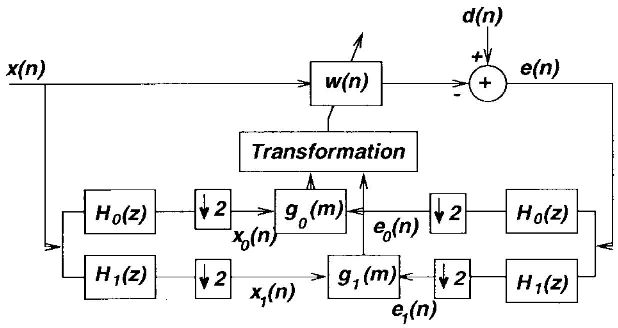

Fig. 1. Structure of the delayless subband adaptive filter.

the most important one, is the effects of aliasing components [3]. To reduce the effect of aliasing, several modifications to the structure of the subband ADF's are proposed [3], [5]–[7]. Using these modifications, we can reduce the effects to some level; however, they do not provide the perfect elimination of the aliasing effects.

To enable subband adaptive signal processing without the effects of aliasing, a delayless structure of subband ADF has been proposed [1], [8], [9]. In contrast to the conventional structure requiring a pair of analysis and synthesis filter banks, the delayless structure can be implemented with only analysis filter bank. Instead of synthesizing signals using synthesis filters, the structure uses two sets of adaptive filters, namely, a fullband one and subband ones. In the adaptation process, the subband ADF's are updated first, and then, the fullband ADF is obtained by transforming the updated subband ADF's. Morgan and Thi proposed [1] to use the fast Fourier transformation (FFT), or inverse FFT (IFFT), as the transformation from subband ADF's into the fullband one.

Recently, Hirayama and Sakai pointed out [8], [9] that if we employ the FFT as the transformation, the fullband filter cannot converge to the optimum Wiener filter. Besides, they show that by using the Hadamard transformation, the theoretical optimum filter of the fullband filter of the delayless structure would be identical to the Wiener filter of the original fullband structure. However, there are no theoretical considerations on the selection of the values of the step-size parameter in [9] and, in addition, no considerations on the selection of the analysis filters.

In this correspondence, we derive the conditions for the convergence of ADF's in the delayless structure and propose a formula of the step-size parameter that ensures the convergence automatically. The validity of the formula is shown through the results of the computer simulations. We also consider the possibility of the efficient implementation of the structure using a short-length analysis filters based on the results of simulations.

II. Delayless Subband Adaptive Filters

In this section, we give a brief description of the delayless subband ADF's. We summarized the definitions and notations of the variables used in this paper in Table I. A structure of the delayless subband adaptive filter with two channel is depicted in Fig. 1. Sections II–IV deal only the case that the number of channels is two.

Note that in the following considerations, we assume that all signals are stationary and that the desired signal $d(n)$ is given as [10]

$$d(n) = \boldsymbol{x}_{fb}^T(n)\boldsymbol{w}_{opt} + q(n) \tag{1}$$

where $\boldsymbol{w}_{opt}$ shows the optimum filter, and $q(n)$ is additive noise. However, for simplicity of notation, we set $q(n) = 0$ in this paper.

TABLE I
Summary of Variables. The Superscript ^{T} Indicates the Transpose Operation

n	:	Time. We assume adaptive filters are updated when n is even and, when n is odd, all variables keeps their states at $n-1$.
$x(n)$	:	Input signal at time n
$d(n)$	:	Desired signal at time n
$e(n)$	:	Fullband error signal at time n
$\boldsymbol{w}(n)$	:	Coefficient vector of fullband filter. $[w_0(n)\ w_1(n)\ \ldots\ w_{S-1}(n)]^T$
S	:	Length of $\boldsymbol{w}(n)$.
M	:	The number of subband channels.
i	:	Suffix used to indicate subband channel. $0 \le i < M$
$\boldsymbol{g}_i(n)$	:	Coefficient vector of adaptive filter in ith channel $[g_{i,0}(n)\ g_{i,1}(n)\ \ldots\ g_{i,N-1}(n)]^T$
N	:	Length of $\boldsymbol{g}_i(n)$. Note that the relation $S = MN$ always holds.
$\boldsymbol{h}_i$	:	Coefficient of analysis filter $[h_{i,0}\ h_{i,1}\ \ldots\ h_{i,L-1}]^T$
L	:	Length of $\boldsymbol{h}_i$
$\boldsymbol{x}_{fb}(n)$	:	Input vector to $\boldsymbol{w}(n)$ $[x(n)\ x(n-1)\ \ldots\ x(n-S+1)]^T$
$\boldsymbol{x}_{sb}(n)$	:	Input vector to $\boldsymbol{h}_i$ $[x(n)\ x(n-1)\ \ldots\ x(n-L+1)]^T$
$x_{ds}(n)$	:	The output of $\boldsymbol{h}_i$ convolved with $\boldsymbol{x}_{sb}(n)$ $x_{ds}(n) = \boldsymbol{h}_i^T\boldsymbol{x}_{sb}(n)$
$\boldsymbol{x}_i(n)$	:	Input vector to $\boldsymbol{g}_i(n)$ $[x_{ds}(n)\ x_{ds}(n-M)\ \ldots\ x_{ds}(n-NM+1)]^T$
$e_i(n)$	:	Error signal in i-th channel at time n.

A. Aspects for Consideration

As shown in Fig. 1, a delayless subband ADF is made up of analysis filters $\boldsymbol{h}_i$, a downsampler indicated by $\downarrow 2$, and two types of adaptive filters, namely, $\boldsymbol{w}(n)$ and $\boldsymbol{g}_i(n)$.

The adaptive processing using the structure shown in Fig. 1 is explained as follows. First, the fullband error $e(n)$ is calculated using

$$\boldsymbol{e}(n) = d(n) - \boldsymbol{x}_{fb}^T(n)\boldsymbol{w}(n) \tag{2}$$

where ^{T} indicates the transpose operation. Then, two signals $\boldsymbol{x}(n)$ and $\boldsymbol{e}(n)$ are analyzed into sub-channels using $\boldsymbol{h}_i(z)$. Adaptive filters in each channel $\boldsymbol{g}_i(n)$ are updated using the LMS algorithm [11]

$$\boldsymbol{g}_i(n+1) = \boldsymbol{g}_i(n) + \mu_i\boldsymbol{x}_i(n)e_i(n) \tag{3}$$

where μ_i is the step-size parameter of the algorithm in each channel. At this point, we should note about the notation of time variable n. We use only one variable n to indicate time instant, and n indicates the highest clock rate in the structure, i.e., the clock rate of input signal $x(n)$. We will see in the following sections that this selection makes it easier to develop the consideration. Note that it is assumed that adaptive filters are updated when n is even, and when n is odd, their states are kept as the same as those at $n-1$.

Adaptive filters in each channel are then transformed into the fullband adaptive filter $\boldsymbol{w}(n)$ [1], [8], [9]. This may symbolically be written as

$$\boldsymbol{w}(n) = \mathbf{T}[\boldsymbol{g}_0(n), \boldsymbol{g}_1(n)] \tag{4}$$

where $\mathbf{T}[\cdot]$ shows the transformation used to transform $\boldsymbol{g}_i(n)$ into $\boldsymbol{w}(n)$.

The performance of the fullband adaptive filter $\boldsymbol{w}(n)$ is determined, therefore, by the employed transformation $\mathbf{T}[\cdot]$, $\boldsymbol{h}_i$ and the characteristics of the adaptive algorithm, namely, the step-size parameter of the LMS algorithm [11]. This implies that we must consider the following four aspects:

1) Selection of the transformation that transforms $\boldsymbol{g}_i(n)$ into $\boldsymbol{w}(n)$;
2) selection of the analysis filter bank $\boldsymbol{h}_i$;
3) conditions for the step-size parameter μ to ensure the convergence of $\boldsymbol{w}(n)$;
4) method for automatically assigning the value for step-size parameter μ.

The first aspect, namely, the selection of the transformation $\mathbf{T}[\cdot]$, is considered in [1], [8], and [9]. Morgan and Thi [1] propose to use the fast Fourier transformation (FFT) as a transformation $\mathbf{T}[\cdot]$. However, it is shown in [8] and [9] that when we use the FFT as the transformation, we cannot uniquely determine $\boldsymbol{g}_0(n)$ and $\boldsymbol{g}_1(n)$ from $\boldsymbol{w}(n)$.

Instead of the FFT, Hirayama and Sakai [8], [9] propose a simpler transformation, and the transformation is given in the z domain in the two-channel case as

$$W(z) = \tfrac{1}{2}[(1+z^{-1})G_0(z^2) + (1-z^{-1})G_1(z^2)] \tag{5}$$

where $W(z)$, $G_0(z)$, and $G_1(z)$ are the z transformations of $\boldsymbol{w}(n)$, $\boldsymbol{g}_0(n)$, and $\boldsymbol{g}_1(n)$, respectively. Note that (5) is identical to the Hadamard transform. In the time domain, (5) is given as

$$\begin{aligned} w_j(n) &= \tfrac{1}{2}[g_{0,m}(n) + (-1)^k g_{1,m}(n)] \\ m &= \left\lceil \frac{j}{2} \right\rceil, \; 0 < j < S-1 \end{aligned} \tag{6}$$

where $\lceil x \rceil$ denotes the largest integer less than x. Using the transformation (5) or (6), we can uniquely determine $\boldsymbol{g}_i(n)$ from $\boldsymbol{w}(n)$, and the optimum solution of $\boldsymbol{w}(n)$ is shown to be the Wiener filter [8], [9] if $\boldsymbol{w}(n)$ convergences.

Although the selection of Hadamard transformation ensures $\boldsymbol{w}(n)$ to have the Wiener filter solution, there are no considerations on the convergence of $\boldsymbol{w}(n)$ thus far. In other words, only aspect 1 has been considered, but aspects 2–4 remain unconsidered. We will consider, therefore, the convergence characteristics of $\boldsymbol{w}(n)$ under the transformation shown in (5) in the following.

B. Expression of Downsampling and Upsampling Using Matrixes

Before considering the convergence characteristics in detail, here, we introduce the notation we will use in the following sections. Note that the description here is restricted on the case that the number of channel M is $M=2$, and the extensions of the expression for the case $M>2$ are given in Section V.

When we considering the convergence characteristics of the delayless subband ADF's, the forms of (5) or (6) are not suitable forms because we cannot express the update recursion of $\boldsymbol{w}(n)$ in a time-domain equation. Here, we develop matrix expressions for the downsampling and transformation processes to obtain the time-domain update recursion of $\boldsymbol{w}(n)$.

First, let us consider the expression of the downsampling process. As can be seen from Fig. 1, $x_i(n)$ is obtained by decimating the output of $\boldsymbol{h}_i$. We want to express the relationship between $\boldsymbol{x}_i(n)$ and $\boldsymbol{x}_{fb}(n)$ in matrix form. For that purpose, we introduce a matrix $\boldsymbol{D}_N$ defined as

$$\boldsymbol{D}_N = \left[\boldsymbol{\phi}_{N,0}, \mathbf{0}_N, \boldsymbol{\phi}_{N,1}, \mathbf{0}_N, \ldots, \boldsymbol{\phi}_{N,N-1}, \mathbf{0}_N\right] \tag{7}$$

where $\boldsymbol{\phi}_{N,k}$ and $\mathbf{0}_N$ are given as

$$\boldsymbol{\phi}_{N,k} = [\overbrace{0, \ldots, 0}^{k}, 1, \overbrace{0, \ldots, 0}^{N-1-k}]^T \tag{8a}$$

$$\mathbf{0}_N = [\overbrace{0 \quad 0 \quad \ldots \quad 0}^{N}]^T \tag{8b}$$

and the size of $\boldsymbol{D}_N$ is $N \times S$. Note that the relation $S = MN$ and that $M = 2$. For example, when $S=4$ and $N=2$, $\boldsymbol{D}_2$ will be given as

$$\boldsymbol{D}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{9}$$

By applying $\boldsymbol{D}_N$ from the left side, we can express the downsampling by a factor of two. As an example, the signal $\boldsymbol{x}_i(m)$ can be expressed as

$$\boldsymbol{x}_i(n) = \boldsymbol{D}_N \boldsymbol{X}_{sb}(n) \boldsymbol{h}_i \tag{10}$$

where $\boldsymbol{X}_{sb}(n)$ is

$$\boldsymbol{X}_{sb}(n) = \begin{bmatrix} \boldsymbol{x}_{sb}(n)^T \\ \boldsymbol{x}_{sb}(n-1)^T \\ \vdots \\ \boldsymbol{x}_{sb}(n-S+1)^T \end{bmatrix}. \tag{11}$$

Next, let us consider expressing the upsampling process using matrixes. This is because the transformation (5) can be regarded as an upsampling process. In this case, we introduce two matrixes $\boldsymbol{U}_{N,0}$ and $\boldsymbol{U}_{N,1}$

$$\boldsymbol{U}_{N,0} = \begin{bmatrix} \boldsymbol{\phi}_{N,0}^T \\ \mathbf{0}_N^T \\ \boldsymbol{\phi}_{N,1}^T \\ \mathbf{0}_N^T \\ \vdots \\ \boldsymbol{\phi}_{N,S-1}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \boldsymbol{U}_{N,1} = \begin{bmatrix} \mathbf{0}_N^T \\ \boldsymbol{\phi}_{N,0}^T \\ \mathbf{0}_N^T \\ \boldsymbol{\phi}_{N,1}^T \\ \vdots \\ \mathbf{0}_N^T \\ \boldsymbol{\phi}_{N,S-1}^T \end{bmatrix} \tag{12}$$

and the size of $\boldsymbol{U}_{N,i}$ is $S \times N$. For example, $\boldsymbol{U}_{2,0}$ and $\boldsymbol{U}_{2,1}$ are given as

$$\boldsymbol{U}_{2,0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{13a}$$

$$\boldsymbol{U}_{2,1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{13b}$$

when $S=4$ and $N=2$. Using $\boldsymbol{U}_{N,0}$ and $\boldsymbol{U}_{N,1}$, we can express the transformation (5) in a time-domain equation as

$$\boldsymbol{w}(n) = \tfrac{1}{2}[\boldsymbol{U}_{N,0}\{\boldsymbol{g}_0(n) + \boldsymbol{g}_1(n)\} + \boldsymbol{U}_{N,1}\{\boldsymbol{g}_0(n) - \boldsymbol{g}_1(n)\}] \tag{14}$$

and is a suitable expression for the following consideration than (6).

C. Expression of the Signals

Here, we consider expressing signals in Fig. 1 using the notation introduced in the above. As shown in the following section, to derive the conditions for convergence, we need to express (3) using $\boldsymbol{x}_{fb}(n)$.

Let us begin with the expression of $e_i(n)$. The error signal in ith channel $e_i(n)$ is given by the inner product

$$e_i(n) = \boldsymbol{h}_i^T \boldsymbol{e}(n) \tag{15}$$

where $\boldsymbol{e}(n)$ is defined as

$$\boldsymbol{e}(n) = [e(n),\, e(n-1),\, \cdots,\, e(n-L+1)]^T. \tag{16}$$

Using the assumption (1), $e(n)$ is expressed as

$$\begin{aligned} e(n) &= d(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n) \\ &= \boldsymbol{x}_{fb}^T(n)\boldsymbol{w}_{opt} - \boldsymbol{x}_{fb}^T(n)\boldsymbol{w}(n). \end{aligned} \tag{17}$$

Hence, (15) can be written as

$$\begin{aligned} e_i(n) &= \sum_{j=0}^{L-1} h_{i,j} e(n-j) \\ &= \sum_{j=0}^{L-1} h_{i,j} \boldsymbol{x}_{fb}^T(n-j)[\boldsymbol{w}_{opt} - \boldsymbol{w}(n-j)]. \end{aligned} \tag{18}$$

Next, let us consider the expression of $\boldsymbol{x}_i(n)$. From (11), we notice

$$\begin{aligned} \boldsymbol{X}_{sb}(n) &= \begin{bmatrix} \boldsymbol{x}_{sb}(n)^T \\ \boldsymbol{x}_{sb}(n-1)^T \\ \vdots \\ \boldsymbol{x}_{sb}(n-S+1)^T \end{bmatrix} \\ &= [\boldsymbol{x}_{fb}(n) \quad \boldsymbol{x}_{fb}(n-1) \quad \cdots \quad \boldsymbol{x}_{fb}(n-L+1)] \end{aligned} \tag{19}$$

and, therefore, in matrix form, we can express $\boldsymbol{x}_i(n)$ as

$$\boldsymbol{x}_i(n) = \boldsymbol{D}_N \left\{ \sum_{j=0}^{L-1} h_{i,j} \boldsymbol{x}_{fb}(n-j) \right\}. \tag{20}$$

Equations (18) and (20) are the forms on which we will derive an analysis method in the following consideration.

III. Derivation of the Formula of the Weight Error

In this section, we derive the recursive formula of the weight-error vector to develop an analysis method of convergence characteristics that is similar to that of the LMS algorithm. Our final results derived in this section are shown in (34). Those not having interest in the detailed derivation may skip this section.

A. Preparation

By substituting (3) into (14), we have

$$\begin{aligned} \boldsymbol{w}(n+2) &= \boldsymbol{w}(n+1) \\ &= \boldsymbol{w}(n) + \tfrac{1}{2}[\boldsymbol{U}_{N,0}\{\mu_0 \boldsymbol{\xi}_0(n) + \mu_1 \boldsymbol{\xi}_1(n)\} \\ &\qquad + \boldsymbol{U}_{N,1}\{\mu_0 \boldsymbol{\xi}_0(n) - \mu_1 \boldsymbol{\xi}_1(n)\}] \end{aligned} \tag{21}$$

where we have used the notations

$$\boldsymbol{\xi}_0(n) = \boldsymbol{e}_0(n)\boldsymbol{x}_0(n) \tag{22a}$$

$$\boldsymbol{\xi}_1(n) = \boldsymbol{e}_1(n)\boldsymbol{x}_1(n). \tag{22b}$$

Note that the relation $\boldsymbol{w}(n+2) = \boldsymbol{w}(n+1)$. For the following description, we define $\Delta\boldsymbol{w}(n)$ as

$$\begin{aligned} \Delta\boldsymbol{w}(n) = \tfrac{1}{2}[&\boldsymbol{U}_{N,0}\{\mu_0 \boldsymbol{\xi}_0(n) + \mu_1 \boldsymbol{\xi}_1(n)\} \\ &+ \boldsymbol{U}_{N,1}\{\mu_0 \boldsymbol{\xi}_0(n) - \mu_1 \boldsymbol{\xi}_1(n)\}]. \end{aligned} \tag{23}$$

In [1], [8], and [9], it is suggested that the step-size parameters in each channel use different values for each channel. This correspondence uses a single step-size parameter μ, and this is shown as

$$\tfrac{1}{2}\mu_0 = \tfrac{1}{2}\mu_1 \stackrel{\Delta}{=} \mu. \tag{24}$$

The validity of this selection is shown in the following consideration. Using (24), (23) is written as

$$\Delta\boldsymbol{w}(n) = \mu[\boldsymbol{U}_{N,0}\{\boldsymbol{\xi}_0(n) + \boldsymbol{\xi}_1(n)\} + \boldsymbol{U}_{N,1}\{\boldsymbol{\xi}_0(n) - \boldsymbol{\xi}_1(n)\}]. \tag{25}$$

B. Derivation of the Update Formula of Weight-Error Vector

We introduce the weight-error vector $\boldsymbol{\epsilon}(n)$ defined as [12]

$$\boldsymbol{\epsilon}(n) = \boldsymbol{w}_{opt} - \boldsymbol{w}(n) \tag{26}$$

to analyze the convergence characteristics of $\boldsymbol{w}(n)$. Using this definition, (21) is written as

$$\boldsymbol{\epsilon}(n+1) = \boldsymbol{\epsilon}(n) - \Delta\boldsymbol{w}(n). \tag{27}$$

To develop the analysis method of the convergence characteristic that is similar to that of the LMS algorithm, we consider expressing $\Delta\boldsymbol{w}(n)$ in terms of $\boldsymbol{\epsilon}(n)$.

Using (18) and (20), $\boldsymbol{\xi}_i(n)$ is expressed as

$$\begin{aligned} \boldsymbol{\xi}_i(n) &= \boldsymbol{D}_N \left[\sum_{j=0}^{L-1} e_i(n) h_{i,j} \boldsymbol{x}_{fb}(n-j) \right] \\ &= \boldsymbol{D}_N \left[\sum_{j=0}^{L-1} \sum_{k=0}^{L-1} h_{i,k} h_{i,j} \boldsymbol{R}(j,k) \boldsymbol{\epsilon}(n-k) \right] \end{aligned} \tag{28}$$

where $\boldsymbol{R}(j,k)$ is

$$\boldsymbol{R}(j,k) = \boldsymbol{x}_{fb}(n-j)\boldsymbol{x}_{fb}^T(n-k). \tag{29}$$

By introducing the notation

$$\beta_0(k,j) = h_{0,k}h_{0,j} + h_{1,k}h_{1,j} \tag{30a}$$

$$\beta_1(k,j) = h_{0,k}h_{0,j} - h_{1,k}h_{1,j} \tag{30b}$$

we can thus express $\Delta\boldsymbol{w}(n)$ as a function of $\boldsymbol{\epsilon}(n)$

$$\begin{aligned} \Delta\boldsymbol{w}(n) = &\mu \boldsymbol{U}_{N,0}\boldsymbol{D}_N \left[\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} \beta_0(k,j)\boldsymbol{R}(j,k)\boldsymbol{\epsilon}(n-k) \right] \\ &+ \mu \boldsymbol{U}_{N,1}\boldsymbol{D}_N \left[\sum_{j=0}^{L-1}\sum_{k=0}^{L-1} \beta_1(k,j)\boldsymbol{R}(j,k)\boldsymbol{\epsilon}(n-k) \right]. \end{aligned} \tag{31}$$

By noting the relation $\boldsymbol{\epsilon}(n) = \boldsymbol{\epsilon}(n-1)$ at even n, (31) can be expressed as

$$\Delta\boldsymbol{w}(n) = \mu \sum_{\ell=0}^{L/2-1} \boldsymbol{\Gamma}_\ell \boldsymbol{\epsilon}(n-2\ell) \tag{32}$$

where $\boldsymbol{\Gamma}_\ell$ is

$$\begin{aligned} \boldsymbol{\Gamma}_\ell = &\boldsymbol{U}_{N,0}\boldsymbol{D}_N \left[\sum_{k=2\ell}^{2\ell+1}\sum_{j=0}^{L-1} \beta_0(k,j)\boldsymbol{R}(j,k) \right] \\ &+ \boldsymbol{U}_{N,1}\boldsymbol{D}_N \left[\sum_{k=2\ell}^{2\ell+1}\sum_{j=0}^{L-1} \beta_1(k,j)\boldsymbol{R}(j,k) \right]. \end{aligned} \tag{33}$$

Therefore, (27) becomes

$$\begin{aligned} \boldsymbol{\epsilon}(n+1) &= \boldsymbol{\epsilon}(n) - \mu \sum_{\ell=0}^{L/2-1} \boldsymbol{\Gamma}_\ell \boldsymbol{\epsilon}(n-2\ell) \\ &= [\mathrm{I} - \mu\boldsymbol{\Gamma}_0]\boldsymbol{\epsilon}(n) - \mu \sum_{\ell=1}^{L/2-1} \boldsymbol{\Gamma}_\ell \boldsymbol{\epsilon}(n-2\ell). \end{aligned} \tag{34}$$

In the next section, we will develop an analysis method of convergence characteristics using (34).

IV. The Conditions for Convergence

In this section, we consider the conditions for the convergence based on the formulation derived in the preceding section. The considerations are separated into two cases, namely, the cases of $L = 2$ and $L > 2$.

For the case of $L = 2$, we show that we can derive the condition for the convergence under one assumption (Assumption 1). Then, we give a formula of the step-size parameter, which satisfies the condition, as Theorem 1. By approximating Theorem 1, a simpler formula that can be implemented with less complexity is given as Corollary 1. The extension of Theorem 1 for the case of $L > 2$ is given by Theorem 2 based on Assumptions 1 and 2.

Note that the conditions we derive in the following are to ensure the convergence of $\boldsymbol{w}(n)$ in mean-square [12].

A. Case of $L = 2$

First, let us consider a special case of $L = 2$. Then, (34) becomes

$$\boldsymbol{\epsilon}(n+2) = [\mathrm{I} - \mu\boldsymbol{\Gamma}_0]\boldsymbol{\epsilon}(n) \tag{35}$$

and this equation resembles the recursion of the weight-error vector used in the convergence analysis of the LMS algorithm [11]. This fact enables us to develop an analysis method that is similar to that of the LMS algorithm in (35). However, we cannot directly apply the analysis of the LMS algorithm. This is because it requires introduction of several assumptions on the statistical characteristics of the input signal, e.g., i.i.d. assumption [12]. Obviously, from the deterministic forms (see [13] and [14]) of (33) and (35), it seems hard to analyze the convergence characteristics under those assumptions. Instead of using the analysis of the LMS algorithm, therefore, we use the analysis method developed for the least squares gradient (LSG) algorithm [15], [16]. As shown in [15], [17], and [18], the LMS algorithm can be interpreted to be an optimization algorithm for one-point least square problem. Using this interpretation, we can analyze the behavior of (35).

Our task, in the following, is to derive the value of μ to ensure the convergence of $\boldsymbol{\epsilon}(n+2)$. For $\boldsymbol{w}(n)$ to converge, we require $\boldsymbol{\epsilon}(n+2)$ to satisfy

$$\mathrm{E}[||\boldsymbol{\epsilon}(n+2)||^2] < \mathrm{E}[||\boldsymbol{\epsilon}(n)||^2] \tag{36}$$

or equivalently

$$\mathrm{E}[\boldsymbol{\epsilon}^T(n)\boldsymbol{\epsilon}(n)] - \mathrm{E}[\boldsymbol{\epsilon}^T(n+2)\boldsymbol{\epsilon}(n+2)] > 0. \tag{37}$$

The condition of μ to ensure the relation (36) is expressed using the largest eigenvalue of $\boldsymbol{\Gamma}_0$ as

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{38}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{\Gamma}_0$. A brief derivation of (38) is given in the Appendix. This condition is valid, provided all the eigenvalues of $\boldsymbol{\Gamma}_0$ are real and positive. Therefore, we introduce the following assumption.

Assumption 1: All the eigenvalues of $\boldsymbol{\Gamma}_0$ are assumed to be real and positive. ◇

Under this assumption, we derive the following theorem.

Theorem 1: The fullband filter $\boldsymbol{w}(n)$ converges if the step-size parameter μ is set as

$$\mu = \frac{\alpha}{\alpha_0(n) + \alpha_1(n)} \tag{39}$$

where

$$0 < \alpha < 2 \tag{40}$$

and

$$\alpha_0(n) = \boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n) + \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n) \tag{41a}$$

$$\alpha_1(n) = \boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n-1) - \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n-1) \tag{41b}$$

when Assumption 1 holds. Note that α corresponds to the parameter of the normalized LMS algorithm [12], [19] and that we omit the notation on α in the following for simplicity of notation. ◇

Let us derive Theorem 1. Under Assumption 1, it is obvious that the sum of all the eigenvalues are bigger than the maximum one

$$\sum \lambda > \lambda_{\max} \tag{42}$$

where the left-hand side of (42) shows the sum of all the eigenvalues of $\boldsymbol{\Gamma}_0$. This equation implies that

$$\frac{2}{\sum \lambda} < \frac{2}{\lambda_{\max}}. \tag{43}$$

Therefore, we can automatically assure the convergence of $\boldsymbol{w}(n)$ at each time n by setting

$$\mu = \frac{\alpha}{\sum \lambda} \qquad 0 < \alpha < 2. \tag{44}$$

We know the sum of all the eigenvalues of a matrix can easily be computed by the trace of a matrix; therefore, $\sum \lambda$ is expressed as

$$\sum \lambda = \mathrm{Tr}[\boldsymbol{\Gamma}_0] \tag{45}$$

where Tr denotes the trace of a matrix.

Let us consider the trace of $\boldsymbol{\Gamma}_0$. From (33) and the property of the trace operation, we obtain

$$\begin{aligned}\mathrm{Tr}[\boldsymbol{\Gamma}_0] = &\sum_{j=0}^{1}\sum_{k=0}^{1} \beta_0(j, k)\mathrm{Tr}[\boldsymbol{U}_{N,0}\boldsymbol{D}_N\boldsymbol{R}(j, k)] \\ &+ \sum_{j=0}^{1}\sum_{k=0}^{1} \beta_1(j, k)\mathrm{Tr}[\boldsymbol{U}_{N,1}\boldsymbol{D}_N\boldsymbol{R}(j, k)].\end{aligned} \tag{46}$$

However, direct calculation of (46) requires huge amounts of calculation. We therefore consider an alternative method to calculate the trace of $\boldsymbol{\Gamma}_0$ in the following.

We consider utilizing $\boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n)$ here. By simple calculation, we find that

$$\begin{aligned}\boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n) &= \left[\sum_{j=0}^{1} h_{i,j}\boldsymbol{x}_{fb}^T(n-j)\boldsymbol{D}_N^T\right]\left[\sum_{j=0}^{1} h_{i,j}\boldsymbol{D}_N\boldsymbol{x}_{fb}(n-j)\right] \\ &= \sum_{j=0}^{1}\sum_{k=0}^{1} h_{i,j}h_{i,k}\boldsymbol{x}_{fb}^T(n-j)\boldsymbol{D}_N^T\boldsymbol{D}_N\boldsymbol{x}_{fb}(n-k).\end{aligned} \tag{47}$$

From this, it is obvious that

$$\begin{aligned}\boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n) + \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n) = \sum_{j=0}^{1}\sum_{k=0}^{1} \beta_0(j, k)\boldsymbol{x}_{fb}^T(n-j)\boldsymbol{D}_N^T\boldsymbol{D}_N\boldsymbol{x}_{fb}(n-k)\end{aligned} \tag{48}$$

and we can easily show the relation

$$\boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n) + \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n) = \sum_{j=0}^{1}\sum_{k=0}^{1} \beta_0(j, k)\mathrm{Tr}[\boldsymbol{U}_{N,0}\boldsymbol{D}_N\boldsymbol{R}(j, k)] \tag{49}$$

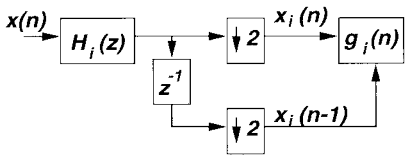

Fig. 2. Possible structure for generating $\boldsymbol{x}(n-1)$.

after some calculation. From the similar steps, we can derive

$$\boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n-1) - \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n-1) = \sum_{j=0}^{1}\sum_{k=0}^{1} \beta_1(j,k)\mathrm{Tr}[\boldsymbol{U}_{N,1}\boldsymbol{D}_N\boldsymbol{R}(j,k)] \tag{50}$$

where we used the input signal at $n-1$, that is, $\boldsymbol{x}(n-1)$, and in Fig. 2, a possible structure for generating $\boldsymbol{x}(n-1)$ is depicted.

By comparing (46) with (49) and (50), we can see that (44) can be satisfied by (39). Hence, we have derived Theorem 1.

B. Consideration of Assumption 1

For Theorem 1 to be valid, the Assumption 1 must hold at each time n; otherwise, (43) is not guaranteed. At the worst case, the algorithm might diverge. Let us consider the validity of the Assumption 1.

We know from the definition of $\boldsymbol{\Gamma}_0$ in (33) that the eigenvalues and eigenvectors of $\boldsymbol{\Gamma}_0$ are adjustable by the coefficients of analysis filters $\boldsymbol{h}_i$. The effect of difference of the analysis filters can be calculated by using $\beta_i(k,j)$ using (30) and (33).

For example, let us consider using the analysis filters

$$\boldsymbol{h}_0 = [1 \quad 1],\ \boldsymbol{h}_1 = [1 \quad -1]. \tag{51}$$

Then, $\beta_i(k,j)$ are calculated as

$$\beta_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},\ \beta_1 = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}. \tag{52}$$

Consequently, $\boldsymbol{\Gamma}_0$ becomes

$$\begin{aligned}\boldsymbol{\Gamma}_0 &= \boldsymbol{U}_{N,0}\boldsymbol{D}_N[2\boldsymbol{R}(0,0) + 2\boldsymbol{R}(1,1)] \\ &\quad + \boldsymbol{U}_{N,1}\boldsymbol{D}_N[2\boldsymbol{R}(0,1) + 2\boldsymbol{R}(1,0)] \\ &= 2\boldsymbol{x}_u(n)\boldsymbol{x}_{fb}(n) + 2\boldsymbol{x}_u(n-1)\boldsymbol{x}_{fb}(n-1) \\ &\quad + 2\boldsymbol{x}_d(n)\boldsymbol{x}_{fb}(n-1) + 2\boldsymbol{x}_d(n-1)\boldsymbol{x}_{fb}(n)\end{aligned} \tag{53}$$

where $\boldsymbol{x}_u$ and $\boldsymbol{x}_d$ are

$$\boldsymbol{x}_u(n) = \boldsymbol{U}_{N,0}\boldsymbol{D}_N\boldsymbol{x}_{fb}(n) \tag{54a}$$
$$\boldsymbol{x}_d(n) = \boldsymbol{U}_{N,1}\boldsymbol{D}_N\boldsymbol{x}_{fb}(n). \tag{54b}$$

Using tapped delay characteristics of $\boldsymbol{x}_{fb}(n)$, (53) is equivalently written as

$$\begin{aligned}\boldsymbol{\Gamma}_0 &= 2\boldsymbol{x}_u(n)\boldsymbol{x}_u(n) + 2\boldsymbol{x}_u(n-1)\boldsymbol{x}_u(n-1) \\ &\quad + 2\boldsymbol{x}_d(n)\boldsymbol{x}_d(n-2) + 2\boldsymbol{x}_d(n-1)\boldsymbol{x}_d(n-1).\end{aligned} \tag{55}$$

On the left-hand side of this equation, sum of all terms; however, the third one can be seen as an deterministic auto-correlation of $\boldsymbol{x}_u(n)$, and hence, the eigenvalues are expected to be positive. However, the third term might violate this positiveness of eigenvalues.

By generalizing this example, the following requirements are posed on βi or the coefficients of analysis filters $\boldsymbol{h}_i$:

- β_0 $\cdots$ The values of the diagonal elements $\beta_0(k,k)$ must be larger than that of other elements.
- β_1 $\cdots$ The values of the elements $\beta_1(k,k-1)$ must be larger than that of other elements.

Thus, we can say that Assumption 1 can be justified by selecting the analysis filters according to the above requirements.

C. Approximation of (39) for Implementation with Less Complexity

As shown in Fig. 2, to implement (50) is somewhat complicated. Here, let us consider the approximation of (39) without using $\boldsymbol{x}(n-1)$.

In actual applications, we can say that

$$\alpha_0(n) \geq \alpha_1(n) \tag{56}$$

can be assumed as long as $\boldsymbol{x}_{fb}(n)$ has the tapped delay-line characteristic. Let us show the reasoning for this assumption. $\boldsymbol{x}_0(n-1)$ and $\boldsymbol{x}_1(n-1)$ can be divided as

$$\boldsymbol{x}_0(n-1) = \kappa_0\boldsymbol{x}_0(n) + \kappa_{0\perp}\boldsymbol{x}_{0\perp}(n) \tag{57a}$$
$$\boldsymbol{x}_1(n-1) = \kappa_1\boldsymbol{x}_1(n) + \kappa_{1\perp}\boldsymbol{x}_{1\perp}(n) \tag{57b}$$

where $\{\boldsymbol{x}_{i\perp}(n),\ i=0,1\}$ are the component that is orthogonal to $\boldsymbol{x}_i(n)$

$$\boldsymbol{x}_{i\perp}^T(n)\boldsymbol{x}_i(n) = 0 \tag{58}$$

and κ_i and $\kappa_{i\perp}$ are defined as

$$\kappa_i = \boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n-1)/\|\boldsymbol{x}_i(n)\| \tag{59a}$$
$$\kappa_{i\perp} = \boldsymbol{x}_{i\perp}^T(n)\boldsymbol{x}_i(n-1)/\|\boldsymbol{x}_i(n)\| \qquad i=0,1 \tag{59b}$$

respectively. Using (57), $\alpha_1(n)$ can be expressed as

$$\alpha_1(n) = \kappa_0\boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n) - \kappa_1\boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n). \tag{60}$$

Therefore, the relation shown in (56) will be valid, provided $|\kappa_0| < 1$ and $|\kappa_1| < 1$ and these inequalities are likely to be satisfied under the tapped delay-line input signal $\boldsymbol{x}_{fb}(n)$.

Using the relation (56), (39) can be approximated by

$$\mu = \frac{\alpha}{2\alpha_0(n)} = \frac{\alpha}{2[\boldsymbol{x}_0^T(n)\boldsymbol{x}_0(n) + \boldsymbol{x}_1^T(n)\boldsymbol{x}_1(n)]}. \tag{61}$$

Equation (61) is simpler and easier to implement than (39), and the validity of this approximation is shown by the computer simulations in Section VI.

Thus, we obtain the following Corollary.

Corollary 1: Under (56), we can approximate (39) as (61). ◇

D. Case of $L > 2$

When $L > 2$, we cannot directly apply the same analysis method in the case of $M = 2$. This is because (34) contains $\{\boldsymbol{\epsilon}(n-2\ell)\colon \ell > 1\}$, and the equation is too complicated to derive the condition for convergence. Instead of deriving the valid condition in theoretically, we consider deriving an approximating condition here. For that purpose, we introduce the next assumption.

Assumption 2: We assume that the weight errors $\{\boldsymbol{\epsilon}(n-\ell): 2 < \ell < L\}$ are similar to $\boldsymbol{\epsilon}(n)$, that is

$$\boldsymbol{\epsilon}(n-\ell) \simeq \boldsymbol{\epsilon}(n) \qquad 2 < \ell < L. \tag{62}$$

◇

Under this assumption, (34) becomes

$$\boldsymbol{\epsilon}(n+2) = \left[\mathrm{I} - \mu \sum_{\ell=0}^{L/2-1} \boldsymbol{\Gamma}_{2\ell}\right]\boldsymbol{\epsilon}(n) \tag{63}$$

and we can use the same procedure as the case of $M = 2$ to derive the condition for μ to ensure convergence. This is summarized in the following Theorem 2.

Theorem 2: Under the Assumptions 1 and 2, we can assure the convergence of $\boldsymbol{w}(n)$ by selecting the step-size parameter as shown in (39). ◇

We can also apply the approximation (56) in this case so that Corollary 1 holds.

E. Consideration of Assumption 2

As shown in the above, for Theorem 2 to be valid, Assumption 2 has to be satisfied. As will be shown in Section VI, the rate of convergence of $\boldsymbol{w}(n)$ will become small compared with the standard LMS algorithm when the delayless structure of the subband filtering is used. Therefore, Assumption 2 is likely to be satisfied in actual applications.

However, development of a general analysis method that does not require Assumptions 1 and 2 is considered as a future work.

V. Conditions for the Case of $M > 2$

Thus far, we have considered the conditions for the convergence under the condition that the number of channel M is $M = 2$. The above analysis is, however, easily extend to the case $M = 2^m$, where m is an integer greater than 1. Here, we give the outline of the extension of the above described analysis method to the case $M = 2^m$.

A. Extension Using the Tree Structure

As shown in [8], delayless subband ADF's can be extended using the tree structure to 2^m channels, where m is an integer greater than 1. For example, when $M = 2^2 = 4$, then the idea of the structure is given as shown in Fig. 3. In this structure, we first update intermediate filters $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ using the same transformation as in the case of $M = 2$ so that they are given as

$$\begin{aligned}\boldsymbol{w}_i(n) = \tfrac{1}{2}[&\boldsymbol{U}_{N,0}\boldsymbol{g}_{2i}(n) + \boldsymbol{U}_{N,0}\boldsymbol{g}_{2i+1}(n)\\ &+ \boldsymbol{U}_{N,1}\boldsymbol{g}_{2i}(n) - \boldsymbol{U}_{N,1}\boldsymbol{g}_{2i+1}(n)].\end{aligned} \tag{64}$$

The fullband filter $\boldsymbol{w}(n)$ is then obtained by transforming $\boldsymbol{w}_i(n)$

$$\begin{aligned}\boldsymbol{w}(n) = \tfrac{1}{2}[&\boldsymbol{U}_{2N,0}\boldsymbol{w}_0(n) + \boldsymbol{U}_{2N,0}\boldsymbol{w}_1(n)\\ &+ \boldsymbol{U}_{2N,1}\boldsymbol{w}_0(n) - \boldsymbol{U}_{2N,1}\boldsymbol{w}_1(n)].\end{aligned} \tag{65}$$

By introducing the notation

$$\boldsymbol{U}^{(4)}_{N,\,i+2j} = \boldsymbol{U}_{2N,\,i}\boldsymbol{U}_{N,\,j} \qquad 0 \le i,\, j < 2 \tag{66}$$

we can directly express $\boldsymbol{w}(n)$ using $\boldsymbol{g}_i(n)$ as

$$\boldsymbol{w}(n) = \sum_{j=0}^{4-1}\sum_{i=0}^{4-1}\psi(i,\,j)\boldsymbol{U}^{(4)}_j\boldsymbol{g}_i(n) \tag{67}$$

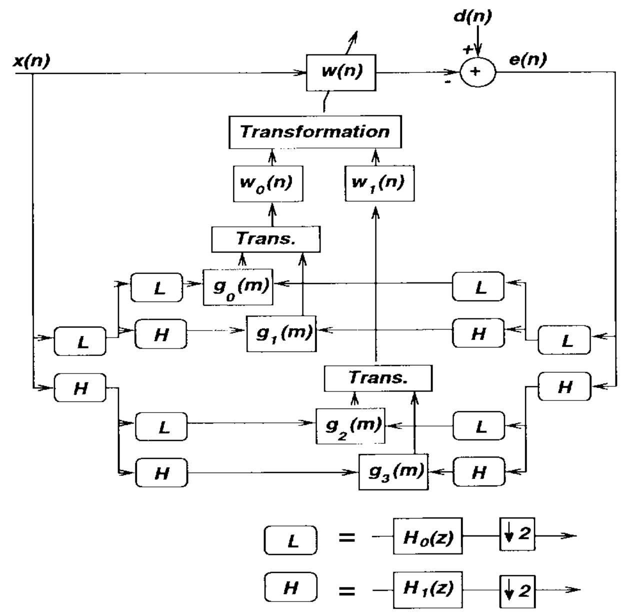

Fig. 3. Tree structure of the delayless subband adaptive filter when the number of the channel is 4.

where $\psi(j,\,i)$ are determined by (64) and (65), and we construct a matrix whose elements are $\psi(i,\,j)$ as

$$\boldsymbol{\psi} = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}. \tag{68}$$

Note that the matrix $\boldsymbol{\psi}$ is a Hadamard matrix.

We can extend the structure to the case $m > 3$ by repeating this procedure.

B. Generalization of Elementary Matrices

Here, we describe the generalization of matrices for expressing upsampling and downsampling shown in Section II-B.

First, let us consider the generalization of the matrices $\boldsymbol{U}_{N,\,i}$ and $\boldsymbol{D}_N$. When the number of the channel is M, the matrix for expressing the decimation is

$$\boldsymbol{D}^{(M)}_N = [\boldsymbol{D}^{(M)}_{N,\,0} \quad \boldsymbol{D}^{(M)}_{N,\,1} \quad \cdots \quad \boldsymbol{D}^{(M)}_{N,\,S-1}] \tag{69}$$

where the superscript $^{(M)}$ indicates that the number of the channel is M, and the size of $\boldsymbol{D}^{(M)}_N$ is $N \times S$. Submatrices $\boldsymbol{D}^{(M)}_{N,\,k}$ are given by

$$\boldsymbol{D}^{(M)}_{N,\,k} = [\boldsymbol{\phi}_{N,\,k} \quad \overbrace{\mathbf{0}_N \quad \cdots \quad \mathbf{0}_N}^{M-1}]. \tag{70}$$

For interpolation, we have M matrices defined as

$$\boldsymbol{U}^{(M)}_{N,\,i} = \begin{bmatrix}\boldsymbol{U}^{(M)}_{N,\,i,\,0}\\ \boldsymbol{U}^{(M)}_{N,\,i,\,1}\\ \vdots\\ \boldsymbol{U}^{(M)}_{N,\,i,\,N-1}\end{bmatrix} \tag{71}$$

where $\boldsymbol{U}^{(M)}_{N,\,i,\,\ell}$ is

$$\boldsymbol{U}^{(M)}_{N,\,i,\,\ell} = [\overbrace{\mathbf{0}_N \quad \cdots \quad \mathbf{0}_N}^{i} \quad \boldsymbol{\phi}_{N,\,\ell} \quad \overbrace{\mathbf{0}_N \quad \cdots \quad \mathbf{0}_N}^{M-i-1}]^T. \tag{72}$$

C. The Condition for the Convergence

Using the matrices (69) and (71) together with the idea of tree structure, we can derive the condition to ensure the convergence in the 2^m-channel case. The derivation of the condition can be done by similar steps shown in Section IV, and here, we only show the result of the derivation. We obtain the following Theorem.

Theorem 3: When 2^m-channel delayless subband ADF's are constructed and the number of channels equals to the length of $\boldsymbol{h}_i$, i.e., $M = L$, then the condition for fullband filter $\boldsymbol{w}(n)$ to converge is given as the condition for the step-size parameter

$$\mu = \frac{\alpha}{\sum_{j=0}^{M-1}\sum_{i=0}^{M-1} \psi(i, j)\boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n-j)} \tag{73}$$

where $\psi(j, i)$ can be calculated by the same step for obtaining (68).◇

Note that Theorem 3 includes Theorem 1 as a special case of $M = 2$. Under the condition $M < L$, we have the following Theorem.

Theorem 4: When we apply 2^m-channel delayless subband ADF's for the structure with the analysis filters whose length L is longer than M, we can assure the convergence of $\boldsymbol{w}(n)$ by selecting the step-size parameter as shown in (73) under Assumptions 1 and 2. ◇

By assuming (56), we obtain the following Corollary.

Corollary 2: Under Assumption (56), we can approximate Theorem 3 or 4 as

$$\mu = \frac{\alpha}{M\sum_{i=0}^{M-1} \psi(i, i)\boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n)}. \tag{74}$$

◇

VI. Consideration on Efficient Implementations Through Computer Simulations

Thus far, we have considered conditions for convergence of ADF's and derived theorems and corollaries. We can say that the considerations on aspects 3 and 4 mentioned in Section II-A are done. Therefore, aspect 2 is left to consider. In this section, we provide some results of computer simulations that show the validity of the above consideration and, through the results, we consider the difference of the convergence characteristics of the ADF's depending on the selection of the analysis filters $\boldsymbol{h}_i$. From that, we show the possibility of efficient implementation of delayless subband ADF's using short-length analysis filters.

In the simulations, we used the following three different analysis filters and compare:

- Filter 1: Length $L = 2$

$$\boldsymbol{h}_0 = [1 \quad 1]^T,\ \boldsymbol{h}_1 = [1 \quad -1]^T. \tag{75}$$

- Filter 2: Length $L = 3$

$$\boldsymbol{h}_0 = [1 \quad 2 \quad 1]^T, \quad \boldsymbol{h}_1 = [1 \quad -2 \quad 1]^T. \tag{76}$$

- Filter 3: Length $L = 8$
  Conjugate quadrature filter (CQF) bank [20].

The conditions for the simulations follow. Input signal $\boldsymbol{x}(n)$ is an AR(1) process given as

$$\boldsymbol{x}(n) = a_0\boldsymbol{x}(n-1) + w(n) \tag{77}$$

where a_0 is the AR parameter, and $w(n)$ is a white Gaussian process with mean 0 and variance 1. Two different values were used for a_0, i.e., $a_0 = 0.0$ and 0.95 in order to simulate the white and the colored processes. The number of channels M is fixed as 2. The optimum filter is an FIR filter with order 9 or length 10, and length of the adaptive filters are selected as $S = 10$ and $N = 5$. The value of α is set as $\alpha = 1$. We did not simulate additive noises to demonstrate the alias-free characteristics of the delayless subband ADF architecture. Comparisons are made in $e^2(n)$, and results shown in this section are averages of 200 independent processes.

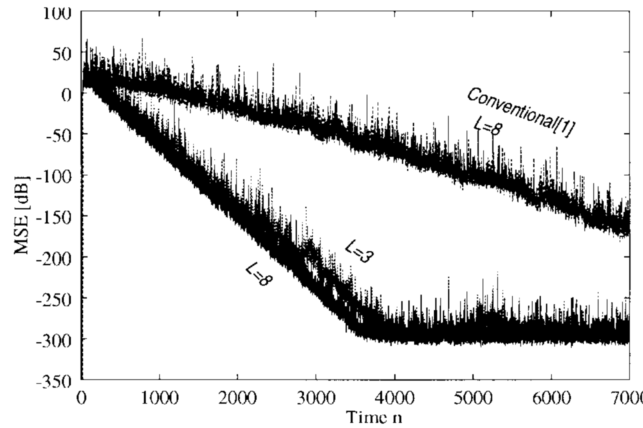

Fig. 4. Comparison of the analysis filter banks. Input signal is a white Gaussian process. L denotes the length of $\boldsymbol{h}_i$. The *conventional* is the result of using (78) under $\mu_s = 0.6$ with $\boldsymbol{h}_i$ of $L = 8$. The theoretical equation (39) was used to update $\boldsymbol{w}(n)$ for the proposed method.

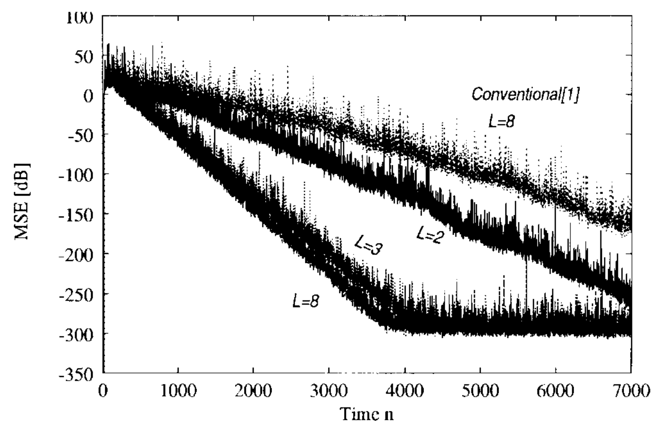

Fig. 5. Comparison of the analysis filter banks. Input signal is a white Gaussian process. The approximated equation (61) was used to update $\boldsymbol{w}(n)$ for the proposed method. L denotes the length of $\boldsymbol{h}_i$. The *conventional* is the result of using (78) under $\mu_s = 0.6$ with $\boldsymbol{h}_i$ of $L = 8$.

For comparison, we show the results of using

$$\mu_i = \frac{\mu_s}{\boldsymbol{x}_i^T(n)\boldsymbol{x}_i(n)} \tag{78}$$

which is suggested in [1] and [8]. The value of μ_s was selected as the best value obtained after simulations with several different values.

The results of simulations are shown in Figs. 4–6. Figs. 4 and 5 show those of the white input case, i.e., $a_0 = 0.0$. On the other hand, Fig. 6 shows those of the colored input case, i.e., $a_0 = 0.95$. In Fig. 4, the results of using (39) are shown and, in Figs. 5 and 6, those of using (61). Note that the lines marked with "conventional" are the results of using (78) and the analysis filter of $L = 8$.

From the results, we can verify the following points.

1) In all the cases, $\boldsymbol{w}(n)$ converges to the optimum point the ordinary fullband ADF's give, i.e., -300 dB in the figures, regardless of the type of analysis filters used and characteristics of the input signal.

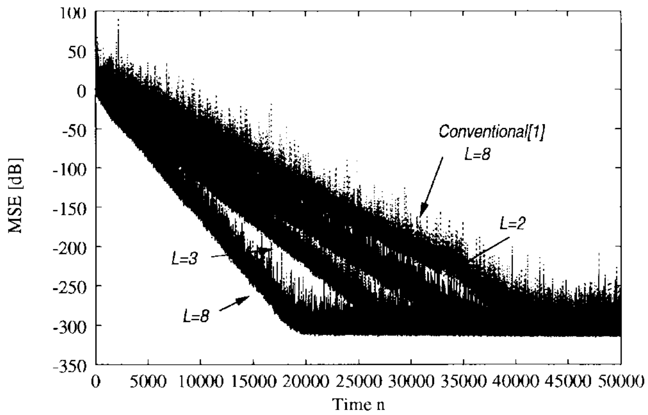

Fig. 6. Comparison of the analysis filter banks. Input signal is an AR(1) process with the AR parameter 0.95. The approximated equation (61) was used to update $\boldsymbol{w}(n)$ for the proposed method. $\boldsymbol{L}$ denotes the length of $\boldsymbol{h}_i$. The *conventional* is the result of using (78) under $\mu_s = 0.5$ with h_i of $L = 8$.

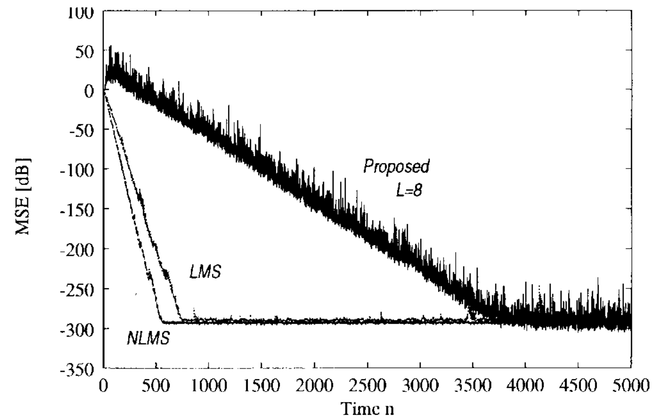

Fig. 7. Comparison with the standard LMS and NLMS algorithms. An AR(1) process with the AR parameter 0.0 was used as the input process. The approximated equation (61) was used to update $\boldsymbol{w}(n)$ for the proposed method. L denotes the length of $\boldsymbol{h}_i$.

2) Equations (39) and (61) give almost identical convergence characteristics, although (61) is an approximation of (39) (Figs. 4 and 5).
3) The proposed formulas provide faster convergence than the conventional method (78) in all the cases.
4) The proposed formulas ensure the convergence of $\boldsymbol{w}(n)$ without adjusting the value of α (α was fixed as 1). However, divergence of $\boldsymbol{w}(n)$ occurred if we selecteded an improper value for μ_s of (78). There is no consideration on the value for μ_s in [1] and [8].
5) Even the analysis filters of $L = 2$ can realize faster convergence than the conventional method, regardless of the characteristics of the input signal (Figs. 5 and 6).

Thus, we have verified the validity of the analysis developed in the above sections. Besides, we have shown the possibility of low computational complexity implementations of the alias-free subband ADF's using short-length analysis filters. This is because (75) or (76) can be implemented using only a few of additions and shift operations. This possibility is not mentioned in [1] [8], or [9]. We also verified the convergence of $\boldsymbol{w}(n)$ under the condition $M > 2$ using (74).

Here, we note the selection of the single step-size parameter of the proposed method. As we mentioned in (24) of Section III, the proposed method uses a single step-size for each subband. Although this selection of single step size seems to conflict with the philosophy of the subband adaptive filtering, it provides the faster rate of convergence than the conventional method (78), which assigns step-size parameters according to the powers of subband signals. However, these results do not assure that the proposed method is the optimum selection. The energy of the signal in each channel is not equally distributed in general, and there is possibility of improving the convergence rate if we utilize these properties of the subband signals.

A. Comparison with the Standard LMS and NLMS Algorithms

Here, we show a comparison of the convergence characteristics with the standard LMS and the normalized LMS (NLMS) algorithms.

The conditions of the simulations for the proposed method are the following; a simplified formula (61) was used, and the analysis filter with its length $L = 8$ was selected. The value of the step size for the LMS algorithm was set as $\mu = 0.1$, and for the NLMS, μ is set as $\mu = 1/\boldsymbol{x}^T(n)\boldsymbol{x}(n)$. As the input signal, a white Gaussian process was used, and results of 200 independent processes were averaged. The results of the simulations are shown in Fig. 7.

From the figure, we can see that the rates of the convergence of the proposed method are slower than those of the LMS and NLMS algorithms, although the proposed method provides faster convergence than the conventional ones, as shown above. The improvement of the convergence characteristics of the delayless structure of subband ADF is considered to be one of the most important aspects of future work.

VII. Conclusion

In this correspondence, we considered conditions for convergence of the delayless subband ADF's proposed by Morgan and Thi [1]. We proposed a formula for the step-size parameter to ensure the convergence of ADF's, regardless of the employed analysis filter bank and characteristics of input signals. The exact formula was given in (39) for the case that the number of channels is two. By approximating (39), we obtained the simpler equation (61) for implementation with low complexity. Then, the formula was extended for the case $M > 2$, and (73) was derived. Through the results of the computer simulations, it was shown that by using the proposed formulas, faster convergence of the ADF's than the conventional method can be achieved, regardless of characteristics of input signal. Besides, the results show the possibility of low computational complexity implementations of the alias-free subband ADF's using short-length analysis filters.

As a future work, a method to improve the rate of convergence has to be developed. For that purpose, a method that uses nonsingle step-size could possibly be an efficient one.

Appendix
Derivation of (38)

From the relation (35), $\boldsymbol{\epsilon}^T(n+2)\boldsymbol{\epsilon}(n+2)$ is expressed as

$$\begin{aligned}\boldsymbol{\epsilon}^T&(n+2)\boldsymbol{\epsilon}(n+2)\\ &= \boldsymbol{\epsilon}^T(n)\boldsymbol{\epsilon}(n) - \mu\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\epsilon}^T(n)\\ &\quad -\mu\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n) + \mu^2\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n).\end{aligned} \tag{79}$$

Substituting (79) into (37), we have

$$\begin{aligned}\mathrm{E}[\boldsymbol{\epsilon}^T&(n)\boldsymbol{\epsilon}(n)] - \mathrm{E}[\boldsymbol{\epsilon}^T(n+2)\boldsymbol{\epsilon}(n+2)]\\ &= \mathrm{E}[\mu\boldsymbol{\epsilon}^T(n)[\boldsymbol{\Gamma}_0^T + \boldsymbol{\Gamma}_0]\boldsymbol{\epsilon}(n)]\\ &\quad - \mathrm{E}[\mu^2\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n)] > 0.\end{aligned} \tag{80}$$

By restricting μ to be positive, or $\mu > 0$, we have

$$\mathrm{E}[\boldsymbol{\epsilon}^T(n)[\boldsymbol{\Gamma}_0^T + \boldsymbol{\Gamma}_0]\boldsymbol{\epsilon}(n)] > \mu\mathrm{E}[\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n)]. \tag{81}$$

From this, we have the upper bound for μ as

$$0 < \mu < \frac{\mathrm{E}[\boldsymbol{\epsilon}^T(n)[\boldsymbol{\Gamma}_0^T + \boldsymbol{\Gamma}_0]\boldsymbol{\epsilon}(n)]}{\mathrm{E}[\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n)]}. \tag{82}$$

Let us consider expressing (82) using the eigenvalues and eigenvectors of $\boldsymbol{\Gamma}_0$. By denoting the ith eigenvector of $\boldsymbol{\Gamma}_0$ by $\boldsymbol{\phi}_i$, $\boldsymbol{\epsilon}(n)$ is expressed as

$$\boldsymbol{\epsilon}(n) = \sum_{i=0}^{N_\lambda - 1} a_i\boldsymbol{\phi}_i + \boldsymbol{\omega} \tag{83}$$

where N_λ is the rank of $\boldsymbol{\Gamma}_0$, $\boldsymbol{\omega}$ is the component of $\boldsymbol{\epsilon}(n)$ orthogonal to the space spanned by eigenvectors $\boldsymbol{\phi}_i$, i.e.,

$$\boldsymbol{\omega}^T\boldsymbol{\phi}_i = 0 \tag{84}$$

and a_i are determined by

$$a_i = \boldsymbol{\epsilon}^T(n)\boldsymbol{\phi}_i. \tag{85}$$

Using (83), $\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\epsilon}(n)$ is written as

$$\begin{aligned}\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\epsilon}(n) &= \left[\sum_{i=0}^{N_\lambda-1}\{a_i\boldsymbol{\phi}_i^T + \boldsymbol{\omega}\}\right]\boldsymbol{\Gamma}_0^T \times \sum_{i=0}^{N_\lambda-1}\{a_i\boldsymbol{\phi}_i + \boldsymbol{\omega}\}\\ &= \left[\sum_{i=0}^{N_\lambda-1} a_i\lambda_i\boldsymbol{\phi}_i^T\right] \times \sum_{i=0}^{N_\lambda-1}\{a_i\boldsymbol{\phi}_i + \boldsymbol{\omega}\}\\ &= \sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i.\end{aligned} \tag{86}$$

Similarly, $\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n)$ and $\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T(n)\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n)$ are expressed as

$$\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n) = \sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i. \tag{87}$$

$$\boldsymbol{\epsilon}^T(n)\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0\boldsymbol{\epsilon}(n) = \sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i^2\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i \tag{88}$$

respectively. Using (86)–(88), the upper bound for μ is expressed as

$$\begin{aligned}\mu &< \frac{2\mathrm{E}\left[\sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right]}{\mathrm{E}\left[\sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i^2\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right]}\\ &< \frac{2\mathrm{E}\left[\sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right]}{\lambda_{\max}\mathrm{E}\left[\sum_{i=0}^{N_\lambda-1} a_i^2\lambda_i\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right]} = \frac{2}{\lambda_{\max}}\end{aligned} \tag{89}$$

where we have assumed eigenvalues $\{\lambda_i\colon\ i = 0, \cdots, N_\lambda\}$ are positive, and we treated $\boldsymbol{\Gamma}_0$ and $\boldsymbol{\Gamma}_1$ as deterministic quantities to derive the above condition.

Therefore, (38) is derived.

References

[1] D. R. Morgan and J. C. Thi, "A delayless subband adaptive filter architecture," *IEEE Trans. Signal Processing,* vol. 43, pp. 1819–1830, Aug. 1995.

[2] H. Yasukawa, S. Shimada, and I. Furukawa, "Acoustic echo canceller with high speech quality," in *Proc. IEEE ICASSP,* 1987, pp. 2125–2128.

[3] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 40, pp. 1862–1875, Aug. 1992.

[4] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Trans. Signal Processing Mag.,* vol. 9, pp. 14–37, Jan. 1992.

[5] H. Kiya and S. Yamaguchi, "FSF (frequency sampling filter) bank for adaptive system identification," in *Proc. IEEE ICASSP,* 1992, pp. 261–264.

[6] Y. Yamada, H. Ochi, and H. Kiya, "A subband adaptive filter allowing maximally decimation," *IEEE J. Select. Areas Commun.,* vol. 12, pp. 1548–1552, Dec. 1994.

[7] F. Amano, P. Meana, A. de Luca, and G. Duchen, "A multirate acoustic echo canceler structure," *IEEE Trans. Commun.,* vol. 43, pp. 2172–2176, July 1995.

[8] N. Hirayama and H. Sakai, "Analysis of a delayless subband adaptive filter," in *Proc. 11th DSP Symp.,* Tottori, Japan, Nov. 1996, pp. A5-2-1–A5-2-4.

[9] ——, "Analysis of a delayless subband adaptive filter," in *Proc. IEEE ICASSP,* Munich, Germany, Apr. 1997, pp. 2333–2336.

[10] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[11] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[12] S. Haykin, *Adaptive Filter Theory,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[13] V. Solo, "The limiting behavior of LMS," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 1909–1922, Dec. 1989.

[14] ——, "The error variance of LMS with time-varying weights," *IEEE Trans. Signal Processing,* vol. 40, pp. 803–813, Apr. 1992.

[15] K. Nishikawa and H. Kiya, "An LS based new gradient type adaptive algorithm—Least square gradient," *Trans. IEICE,* vol. E77-A, pp. 1417–1425, Sept. 1994.

[16] ——, "An LS based new gradient type adaptive algorithm—Least squares gradient," in *Proc. Int. Conf. Acoust., Speech, Signal Process.,* Adelaide, Australia, Apr. 1994.

[17] D. T. M. Slock, "On the convergence speed of the LMS and the normalized LMS algorithms," *IEEE Trans. Signal Processing,* vol. 41, pp. 2811–2825, Sept. 1993.

[18] K. Nishikawa and H. Kiya, "A representation method of the convergence characteristic of the LMS algorithm using tap-input vectors," *Trans. IEICE,* vol. E78-A, 1995.

[19] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.,* vol. 12, pp. 282–287, June 1967.

[20] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 434–441, June 1986.