PAPER *Special Section on Papers Selected from ITC-CSCC 2001*

Analysis of the Convergence Condition of LMS Adaptive Digital Filter Using Distributed Arithmetic

Kyo TAKAHASHI[†a)], Yoshitaka TSUNEKAWA[††], Norio TAYAMA[††], *and* Kyoushirou SEKI[††], *Regular Members*

SUMMARY An LMS adaptive digital filter using distributed arithmetic (DA-ADF) has been proposed. Cowan and others proposed the DA adaptive algorithm with offset binary coding for the simple derivation of an algorithm and the use of an odd-symmetry property of adaptive function space (AFS) [3], [5], [10]. However, we indicated that a convergence speed of this DA adaptive algorithm degraded extremely by our computer simulations [6]. To overcome these problems, we have proposed the DA adaptive algorithm generalized with two's complement representation and effective architectures. Our DA-ADF has performances of a high speed, small output latency, a good convergence speed, small-scale hardware and lower power dissipation for higher order, simultaneously. In this paper, we analyze a convergence condition of DA adaptive algorithm that has never been considered theoretically [8], [9]. From this analysis, we indicate that the convergence speed is depended on a distribution of eigenvalues of an auto-correlation matrix of an extended input signal vector [8], [9]. Furthermore, we obtain the eigenvalues theoretically. As a result, we clearly show that our DA-ADF has an advantage of the conventional DA-ADF in the convergence speed.

key words: *distributed arithmetic, LMS algorithm, adaptive function space, convergence condition, offset bias*

1. Introduction

In recent years, adaptive filters are used in many applications, for example an echo canceller, a noise canceller, an adaptive equalizer and so on, and the necessity of their implementations is growing up in many fields. Adaptive filters require various performances of a high speed, lower power dissipation, good convergence characteristics, small output latency and so on, for their implementations. However it is difficult to satisfy these characteristics simultaneously, so efficient algorithms and effective architectures are desired. The echo canceller used in Videoconferencing requires the performances of fast convergence characteristics and a capability to track the time varying impulse response [1]. Therefore, it is necessary to implement very high order adaptive filters. It is well-known that a distributed arithmetic is an efficient calculation method of an inner product of constant vector [2]. The distributed arithmetic is able to calculate the inner product by shift and accumulation of partial products stored in the ROM(Read Only Memory) table, so it is possible to realize hardware without multipliers. Cowan and others proposed an LMS adaptive filter using the distributed arithmetic with offset binary coding [3], [5], [10]. In offset binary coding, the nature of odd-symmetry property of the AFS appears in the same as the case of constant vector, so it is possible to save half memory size. However, we indicated that the convergence speed of this method degrades extremely by our computer simulations [6]. This degradation results from an offset bias added to an input signal coded in offset binary, so that the offset binary coding is not suitable for the DA adaptive algorithm.

To overcome this problem, we developed an update algorithm generalized with two's complement representation [7]. Furthermore we find first the nature of approximate odd-symmetry of the AFS in the case of two's complement representation [7]. This nature is the characteristics of the adaptive filter, but has never been seen in the case of constant vector. Furthermore we proposed efficient structures of DA-ADF suitable for its implementation that has never been proposed [6]. Our proposed DA-ADF is a high-performance adaptive filter which has performances of a high speed and small output latency, a good convergence speed, small-scale hardware and lower power dissipation for higher order, simultaneously [6]. However, the convergence condition of DA adaptive algorithm has never been considered.

In this paper, we analyze the convergence condition of DA adaptive algorithm theoretically and indicate that our proposed DA-ADF has an advantage of the conventional DA-ADF in the convergence speed [8], [9]. To analyze that, we extend an original update and output equation to whole adaptive function space and define an extended input signal vector. Using these equations, we derive a convergence equation of an error vector of the adaptive function space and the convergence condition. This result clearly shows that the convergence speed of DA-ADF depends on a distribution of eigenvalues of an auto-correlation matrix of the extended input signal vector. We can evaluate the convergence speed of two DA-ADFs by inspecting the distribution of eigenvalues. Furthermore, we obtain the eigenvalues of the auto-correlation matrix theoret-

Manuscript received October 1, 2001.
Manuscript revised December 25, 2001.
Final manuscript received February 25, 2002.

[†]The author is with the Iwate Industrial Technology Junior College, Iwate-ken, 028-3615 Japan.
[††]The authors are with the Faculty of Engineering, Iwate University, Morioka-shi, 020-8551 Japan.
a) E-mail: kyo@iwate-it.ac.jp

ically. We show that the conventional DA-ADF has large distribution of eigenvalues, whereas our proposed DA-ADF has the same eigenvalues. This means that the convergence speed of the conventional DA-ADF degrades extremely, whereas our proposed DA-ADF has a good convergence speed.

2. LMS Adaptive Filter Using Distributed Arithmetic

It is well known that the distributed arithmetic is an efficient calculation method of the inner product of constant vector. Furthermore it is suitable for time varying coefficient vector in the adaptive filter. An N-th order input signal vector $\boldsymbol{S}(k)$ and an N-taps coefficients vector $\boldsymbol{W}(k)$ are represented as

$$\boldsymbol{S}(k) = [s(k), s(k-1), \cdots, s(k-N+1)]^T$$

and

$$\boldsymbol{W}(k) = [w_0(k), w_1(k), \cdots, w_{N-1}(k)]^T.$$

An output signal of FIR filter can be represented as

$$y(k) = \boldsymbol{S}^T(k)\,\boldsymbol{W}(k) = \boldsymbol{F}^T\boldsymbol{A}^T(k)\,\boldsymbol{W}(k), \qquad (1)$$

where an address matrix $\boldsymbol{A}(k)$ and a scaling vector $\boldsymbol{F}$ are represented as

$$\boldsymbol{A}(k) = \begin{bmatrix} b_0(k) & \cdots & b_0(k-N+1) \\ b_1(k) & \cdots & b_1(k-N+1) \\ \vdots & \ddots & \vdots \\ b_{B-1}(k) & \cdots & b_{B-1}(k-N+1) \end{bmatrix}^T$$

and

$$\boldsymbol{F} = [-2^0, 2^{-1}, \cdots, 2^{-(B-1)}]^T,$$

and B indicates word length of the input signal. The relation between the input signal and the address matrix is

$$s(k) = [b_0(k), b_1(k), \cdots, b_{B-1}(k)]\,\boldsymbol{F}. \qquad (2)$$

We define an address vector

$$\boldsymbol{A}_{vi}(k) = [b_i(k), b_i(k-1), \cdots, b_i(k-N+1)]^T, \quad i = 0, 1, \cdots, B-1.$$

An update equation of LMS algorithm is represented as

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + 2\mu e(k)\,\boldsymbol{S}(k) \qquad (3)$$

[1], multiplying the both sides by $\boldsymbol{A}^T(k)$ from the left,

$$\boldsymbol{A}^T(k)\boldsymbol{W}(k+1) = \boldsymbol{A}^T(k)\{\boldsymbol{W}(k) + 2\mu e(k)\boldsymbol{A}(k)\boldsymbol{F}\}. \qquad (4)$$

The error signal $e(k)$ is obtained by

$$e(k) = d(k) - y(k), \qquad (5)$$

where $d(k)$ is a desired signal. The AFS is defined as

$$\boldsymbol{P}(k) = \boldsymbol{A}^T(k)\,\boldsymbol{W}(k) = [p_0(k), \cdots, p_{B-1}(k)]^T \qquad (6)$$

and

$$\boldsymbol{P}(k+1) = \boldsymbol{A}^T(k)\,\boldsymbol{W}(k+1) = [p_0(k+1), \cdots, p_{B-1}(k+1)]^T. \qquad (7)$$

The i-th elements of $\boldsymbol{P}(k)$ and $\boldsymbol{P}(k+1)$ are partial products related to $\boldsymbol{A}_{vi}(k)$ which is the i-th row vector of $\boldsymbol{A}^T(k)$, so that the time indexes of $\boldsymbol{P}(k)$ and $\boldsymbol{P}(k{+}1)$ correspond to $\boldsymbol{W}(k)$ and $\boldsymbol{W}(k{+}1)$, respectively. There exist 2^N partial products for the N-th order input signal vector. We call this space including the 2^N partial products "whole AFS." $\boldsymbol{P}(k)$ and $\boldsymbol{P}(k+1)$ include B elements selected from the whole AFS by B address vectors. Substituting Eq. (6) and Eq. (7) for Eq. (4), we obtain

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + 2\mu e(k)\,\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}. \qquad (8)$$

Using Eq. (6), the output signal can be represented as

$$y(k) = \boldsymbol{F}^T\boldsymbol{P}(k). \qquad (9)$$

We enabled a diagonalization of $\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}$ first [6]. In the two's complement representation, it was thought that this diagonalization might be impossible heretofore. If we assume that the input signal is white noise with zero-mean, an expectation of $\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}$ becomes

$$E[\,\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}] = 0.25N\,\boldsymbol{F} \qquad (10)$$

[6]. Replacing $\boldsymbol{A}^T(k)\,\boldsymbol{A}(k)\,\boldsymbol{F}$ in Eq. (8) with Eq. (10), Eq. (8) is simplified to

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + 0.5\mu N e(k)\,\boldsymbol{F}. \qquad (11)$$

We confirmed that this simplified algorithm converges in computer simulation for many cases. The term $0.5\mu N$ in Eq. (11) can be treated as a constant, integer power of 2. Therefore it is possible to implement hardware without multipliers, so-called multiplier-less. Figure 1 shows a basic hardware configuration of the DA-ADF. The address vector $\boldsymbol{A}_{vi}$ is used as an address signal of RAM (Random Access Memory). Our proposed MDA-ADF can achieve good performances of a high speed, small output latency, a good convergence speed, small-scale hardware and lower power dissipation, simultaneously [6].

3. Derivation of the Convergence Condition

DA-ADF estimates a transfer function of an unknown system as the AFS, so that the convergence equation should be derived to the whole AFS. However, a subset of the whole AFS updated at the time k is only represented in Eq. (11), so that it is not able to derive the

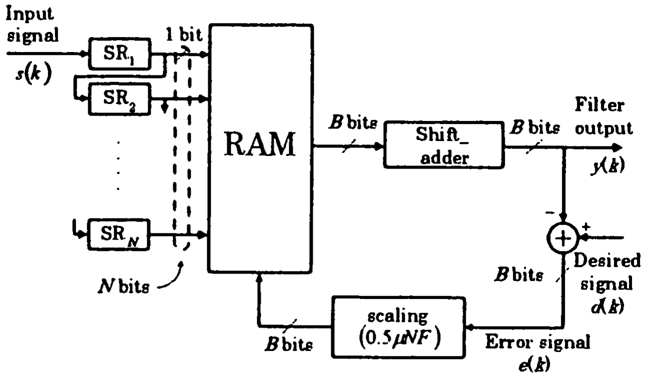


Fig. 1 Block diagram of DA adaptive filter. SR:B bits serial Shift Registers, RAM:Random Access Memory.

convergence condition for the whole AFS. Therefore, (1) We extend the update and output equation to the whole AFS and define an extended input signal vector. (2) We define an estimation error as the difference of an optimum value and an estimate. (3) The convergence condition is defined as the condition for the estimation error to decrease for increasing time k.

3.1 Extension of the Update and Output Equation

We show the extension procedure for the tap number $N=1$ and $N=2$ using examples, and finally we generalize the update equation to the tap number N. In our proposed method, we define the extended input signal vector to update the whole AFS for the tap number $N=1$. For instance, the input signal is represented as

$$s(k)=0\times(-2^0)+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}, \quad (12)$$

the whole AFS has two elements of $p0(k)$ and $p1(k)$, and they are updated using Eq. (11) as follows.

$$\begin{aligned} p0(k+1) &= p0(k)+0.5\mu Ne(k)[-2^0+2^{-2}] \\ &= p0(k)+0.5\mu Ne(k)\bar{s}(k) \end{aligned} \quad (13)$$

and

$$\begin{aligned} p1(k+1) &= p1(k)+0.5\mu Ne(k)[2^{-1}+2^{-3}] \\ &= p1(k)+0.5\mu Ne(k)s(k), \end{aligned} \quad (14)$$

where $\bar{s}(k)$ has the inverse bit pattern of $s(k)$. Equation (13) and Eq. (14) indicate that $p0(k)$ and $p1(k)$ are updated using $\bar{s}(k)$ and $s(k)$, respectively. For any input signal, the update equation is represented as follows.

$$\boldsymbol{P}_w(k+1)=\boldsymbol{P}_w(k)+0.5\mu e(k)\,\boldsymbol{S}_{DA}(k) \quad (15)$$

$$\boldsymbol{S}_{DA}(k)=N[\bar{s}(k),s(k)]^T \quad (16)$$

$$\boldsymbol{P}_w(k)=[p0(k),p1(k)]^T \quad (17)$$

$\boldsymbol{P}_w(k)$ indicates the whole AFS. We realize that the elements of the whole AFS is updated using the $\boldsymbol{S}_{DA}(k)$. We call this the extended input signal vector.

The next consideration is for the tap number $N=$

Table 1 Relation between symbols and bit patterns.

Symbol	Bit pattern	Symbol	Bit pattern
$\boldsymbol{a}$	$[00]^T$	$\boldsymbol{c}$	$[10]^T$
$\boldsymbol{b}$	$[01]^T$	$\boldsymbol{d}$	$[11]^T$

Table 2 Access pattern of adaptive function space.

Symbol	Bit pattern	-2^0	2^{-1}	2^{-2}	2^{-3}
$\boldsymbol{a}$	$[00]^T$	0	0	1	0
$\boldsymbol{b}$	$[01]^T$	0	1	0	0
$\boldsymbol{c}$	$[10]^T$	1	0	0	0
$\boldsymbol{d}$	$[11]^T$	0	0	0	1

2. The input signal vector is

$$\boldsymbol{S}(k)=[s(k),s(k-1)]^T. \quad (18)$$

The N-th order address vector $\boldsymbol{A}_{vi}(k)$ has 2^N patterns, so that four patterns of the address vector exist for the tap number $N=2$. They are identified using alphabets as in Table 1. Now, we consider an example of the input signal vector consisted of two signals

$$s(k)=1\times(-2^0)+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

and

$$s(k-1)=0\times(-2^0)+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}.$$

The input signal vector is represented using symbols as

$$\boldsymbol{S}_s=\boldsymbol{c}\times(-2^0)+\boldsymbol{b}\times2^{-1}+\boldsymbol{a}\times2^{-2}+\boldsymbol{d}\times2^{-3}, \quad (19)$$

where the word length is 4, and we selected the bit pattern of two input signals having the all symbols. In this case, the whole AFS is updated as follows.

$$pa(k+1) = pa(k)+0.5\mu\times2\times e(k)2^{-2} \quad (20)$$

$$pb(k+1) = pb(k)+0.5\mu\times2\times e(k)2^{-1} \quad (21)$$

$$pc(k+1) = pc(k)-0.5\mu\times2\times e(k)2^{0} \quad (22)$$

$$pd(k+1) = pd(k)+0.5\mu\times2\times e(k)2^{-3} \quad (23)$$

From Eq. (20) to Eq. (23), we obtain

$$\boldsymbol{P}_w(k+1)=\boldsymbol{P}_w(k)+0.5\mu Ne(k)\,\boldsymbol{A}_{ac}(k)\,\boldsymbol{F}. \quad (24)$$

Table 2 shows the relationship between elements of the scaling vector and elements of the AFS. The elements of $\boldsymbol{A}_{ac}(k)$ in Eq. (24) correspond to Table 2, so the $\boldsymbol{A}_{ac}(k)$ is 4×4 matrix as follows.

$$\boldsymbol{A}_{ac}(k)=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

This matrix is determined by the bit pattern of the input signal vector. We call this an access matrix. And

$$\boldsymbol{P}_w(k)=[pa(k),pb(k),pc(k),pd(k)]^T. \quad (26)$$

We generalize Eq. (24) to the bit number B and the tap number N, we obtain

$$\boldsymbol{P}_w(k+1)= \boldsymbol{P}_w(k)+0.5\mu e(k)\,\boldsymbol{S}_{DA}(k), \tag{27}$$

$$\begin{aligned}\boldsymbol{S}_{DA}(k) &= N\,\boldsymbol{A}_{ac}(k)\,\boldsymbol{F},\\ &= [s_{DA,0}(k),\cdots,s_{DA,2^N-1}(k)]^T,\end{aligned} \tag{28}$$

where $\boldsymbol{A}_{ac}(k)$ indicates the access matrix of $2^N \times B$,

$$\boldsymbol{P}_w(k) = [p_0(k),p_1(k),\cdots,p_{2^N-1}(k)]^T \tag{29}$$

and

$$\boldsymbol{F} = [-2^0,2^{-1},\cdots,2^{-B+1}]^T. \tag{30}$$

The output signal $y(k)$ and the error signal $e(k)$ are

$$\begin{aligned} y(k) &= \boldsymbol{F}^T\,\boldsymbol{A}_{ac}^T(k)\,\boldsymbol{P}_w(k)\\ &= \frac{1}{N}\,\boldsymbol{S}_{DA}^T(k)\,\boldsymbol{P}_w(k)\end{aligned} \tag{31}$$

and

$$e(k) = d(k) - y(k), \tag{32}$$

respectively. Comparing Eq. (27) and Eq. (3), $\boldsymbol{S}_{DA}(k)$ corresponds to the input signal vector in Eq. (3). So $\boldsymbol{S}_{DA}(k)$ determines the update value of the whole AFS.

The update equation of the conventional method can be derived in the same manner, therefore

$$\boldsymbol{P}'_w(k+1)= \boldsymbol{P}'_w(k) + 2\mu e'(k)\,\boldsymbol{S}'_{DA}(k), \tag{33}$$

$$\begin{aligned}\boldsymbol{S}'_{DA}(k) &= N\,\boldsymbol{A}'_{ac}(k)\,\boldsymbol{F}',\\ &= [s'_{DA,0}(k),\cdots,s'_{DA,2^N-1}(k)]^T,\end{aligned} \tag{34}$$

where the bit have two values of "0" and "1." $\boldsymbol{A}'_{ac}(k)$ indicates the access matrix of $2^N \times B$,

$$\boldsymbol{P}'_w(k) = [p'_0(k),p'_1(k),\cdots,p'_{2^N-1}(k)]^T \tag{35}$$

and

$$\boldsymbol{F}' = [2^{-1},2^{-2},\cdots,2^{-B}]^T. \tag{36}$$

The output signal $y'(k)$ and the error signal $e'(k)$ are

$$\begin{aligned} y'(k) &= \boldsymbol{F}'^T\,\boldsymbol{A}'^T_{ac}(k)\,\boldsymbol{P}'_w(k)\\ &= \frac{1}{N}\,\boldsymbol{S}'^T_{DA}(k)\,\boldsymbol{P}'_w(k), \end{aligned} \tag{37}$$

$$e'(k) = d(k) - y'(k). \tag{38}$$

Until now, it is understood that the input signal of DA-ADF was only used to specify the element of the whole AFS. However, introducing the access matrix $\boldsymbol{A}_{ac}(k)$, we extended the update and output equation to the whole AFS. From our extension, it is clearly shown that the whole AFS is updated by using the extended input signal vector.

3.2 Convergence Condition

The convergence condition of DA-ADF is obtained as follows [9]. Substituting Eq. (31) for Eq. (27), we obtain

$$\begin{aligned}\boldsymbol{P}_w(k+1) &= \left[\boldsymbol{I} - 0.5\frac{1}{N}\mu\,\boldsymbol{S}_{DA}(k)\,\boldsymbol{S}_{DA}^T(k)\right]\boldsymbol{P}_w(k)\\ &\quad + 0.5\mu d(k)\,\boldsymbol{S}_{DA}(k),\end{aligned} \tag{39}$$

where the matrix $\boldsymbol{I}$ is a unit matrix of $2^N \times 2^N$. Now, we define the error vector of the whole AFS as

$$\boldsymbol{c}(k) = \boldsymbol{P}_w(k) - \boldsymbol{P}_w^*, \tag{40}$$

$$\boldsymbol{P}_w^* = N\,\boldsymbol{R}^{-1}\,\boldsymbol{q} \tag{41}$$

where the matrix $\boldsymbol{P}_w^*$ indicates the optimum value of $\boldsymbol{P}_w(k)$ which is derived from the normal equation of DA-ADF

$$\boldsymbol{q} = \frac{1}{N}\,\boldsymbol{R}\,\boldsymbol{P}_w^* \tag{42}$$

derived in [9]. The matrix $\boldsymbol{R}$ is an auto-correlation matrix of the extended input signal vector defined by

$$\boldsymbol{R} = E[\,\boldsymbol{S}_{DA}(k)\,\boldsymbol{S}_{DA}^T(k)], \tag{43}$$

and the vector $\boldsymbol{q}$ is defined by

$$\boldsymbol{q} = E[d(k)\,\boldsymbol{S}_{DA}(k)]. \tag{44}$$

$\boldsymbol{R}^{-1}$ indicates the inverse matrix of $\boldsymbol{R}$. The $\boldsymbol{P}_w^*$ is optimum in a sense of least mean square of the error signal. Using these relations, Eq. (39) becomes

$$\begin{aligned}\boldsymbol{c}(k+1) &= \left[\boldsymbol{I} - 0.5\frac{1}{N}\mu\,\boldsymbol{S}_{DA}(k)\,\boldsymbol{S}_{DA}^T(k)\right]\boldsymbol{c}(k)\\ &\quad + 0.5\mu\Bigg[d(k)\,\boldsymbol{S}_{DA}(k)\\ &\quad - \frac{1}{N}\,\boldsymbol{S}_{DA}(k)\,\boldsymbol{S}_{DA}^T(k)\,\boldsymbol{P}_w^*\Bigg].\end{aligned} \tag{45}$$

The expectation of this equation is

$$\begin{aligned}E[\,\boldsymbol{c}(k+1)] &= \left[\boldsymbol{I} - 0.5\frac{1}{N}\mu\,\boldsymbol{R}\right]E[\,\boldsymbol{c}(k)]\\ &\quad + 0.5\mu\left[\boldsymbol{q} - \frac{1}{N}\,\boldsymbol{R}\,\boldsymbol{P}_w^*\right]\end{aligned} \tag{46}$$

from the independence of $\boldsymbol{c}(k)$ and $\boldsymbol{S}_{DA}(k)$. Substituting Eq. (42) for Eq. (46), we obtain

$$\begin{aligned}E[\,\boldsymbol{c}(k+1)] &= \left[\boldsymbol{I} - 0.5\frac{1}{N}\mu\,\boldsymbol{R}\right]E[\,\boldsymbol{c}(k)]\\ &= [\boldsymbol{I} - \mu_a\,\boldsymbol{R}]E[\,\boldsymbol{c}(k)],\end{aligned} \tag{47}$$

where

$$\mu_a = \frac{0.5}{N}\mu \tag{48}$$

This equation indicates the update equation of the error vector of AFS, and whether $\boldsymbol{c}(k)$ decrease or not depends on μ_a and $\boldsymbol{R}$ for increasing time k. To indicate the property clearly, $\boldsymbol{R}$ can be modified to

$$\boldsymbol{R} = \boldsymbol{Q}\,\boldsymbol{D}\,\boldsymbol{Q}^T, \tag{49}$$

where $\boldsymbol{Q}$ is an orthogonal matrix of

$$\boldsymbol{Q}^T = \boldsymbol{Q}^{-1},$$

and $\boldsymbol{D}$ is a diagonal matrix of

$$\boldsymbol{D} = Diag(\lambda_1, \lambda_2, \cdots, \lambda_{2^N})$$

where the λ_i, $(i = 1, \cdots, 2^N)$ indicate eigenvalues of $\boldsymbol{R}$. From Eq. (49), Eq. (47) is modified to

$$E[\,\boldsymbol{c}(k+1)] = \boldsymbol{Q}[\boldsymbol{I} - \mu_a\,\boldsymbol{D}]\,\boldsymbol{Q}^T E[\,\boldsymbol{c}(k)]. \tag{50}$$

Hence the expectation of the error vector is able to decrease in the condition of

$$0 < \mu_a < \frac{1}{\lambda_{max}}, \tag{51}$$

where λ_{max} is a maximum eigenvalue of $\boldsymbol{R}$.

We can derive the convergence condition of the conventional method in the same manner, so that the update equation of the error vector is

$$\begin{aligned} E[\,\boldsymbol{c}'(k+1)] &= \left[\boldsymbol{I} - 2\frac{1}{N}\mu'\,\boldsymbol{R}'\right] E[\,\boldsymbol{c}'(k)] \\ &= [\,\boldsymbol{I} - \mu'_a\,\boldsymbol{R}']E[\,\boldsymbol{c}'(k)], \end{aligned} \tag{52}$$

where

$$\boldsymbol{c}'(k) = \boldsymbol{P}'_w(k) - \boldsymbol{P}'^*_w \tag{53}$$

and

$$\mu'_a = \frac{2}{N}\mu'. \tag{54}$$

The convergence condition is

$$0 < \mu'_a < \frac{1}{\lambda'_{max}}, \tag{55}$$

where λ'_{max} is a maximum eigenvalue of $\boldsymbol{R}'$. The convergence speed is depended on the distribution of eigenvalues of the auto-correlation matrix, so that we can evaluate the convergence speed of DA-ADF by inspecting the distribution of eigenvalues.

4. Eigenvalues of the Auto-Correlation Matrix

4.1 Eigenvalues of Our Proposed Method

We try to obtain the eigenvalues of the auto-correlation matrix of our proposed method. In the following discussion, we assume that the input signal $s(k)$ is stationary, zero-mean and that successive samples are uncorrelated, and further assume that the constituent bits of a signal sample are themselves uncorrelated [5]. Based on these assumptions, the only one element of "1" occurs at random in the each column vector of the access matrix, and row vectors are independent each other. So $s_{DA,i}(k)$ is the sum of $s_{i,j}(k)$ that occurs depending on the j-th column of the access matrix. From Eq. (28), the signal $s_{i,j}(k)$ is obtained as follows.

$$s_{i,j}(k) = \begin{cases} N \times \boldsymbol{F}_j & \text{in a probability of } Pr_1 \\ 0 & \text{in a probability of } Pr_0 \end{cases}$$

$\boldsymbol{F}_j$ indicates the j-th element of the scaling vector, and the Pr_1 is the probability that one element of the column vector has "1" and the Pr_0 is the probability that the element of the column vector doesn't have "1," therefore

$$\begin{aligned} Pr_1 &= \frac{1}{2^N} \\ Pr_0 &= 1 - Pr_1 = \frac{2^N - 1}{2^N} \end{aligned}$$

We derive an average and a variance of diagonal elements of the auto-correlation matrix. First, the average ave_0 and the variance var_0 of $s_{i,0}(k)$ determined by the first column vector are

$$ave_0 = (-2^0 \times Pr_1 + 0 \times Pr_0) \times N$$

and

$$\begin{aligned} var_0 = &(-2^0 \times N - ave_0)^2 \times Pr_1 \\ &+ (0 \times N - ave_0)^2 \times Pr_0. \end{aligned}$$

Secondarily, the average ave_1 and the variance var_1 of $s_{i,1}(k)$ determined by the second column vector are

$$ave_1 = (2^{-1} \times Pr_1 + 0 \times Pr_0) \times N$$

and

$$\begin{aligned} var_1 = &(2^{-1} \times N - ave_1)^2 \times Pr_1 \\ &+ (0 \times N - ave_1)^2 \times Pr_0. \end{aligned}$$

We can derive them for the third to the B-th column vector in the same manner. $s_{DA,i}(k)$ is the sum of the random signals determined by these column vector, so that, from the central limit theorem, the average ave and the variance var of $s_{DA,i}(k)$, $(i = 0, \cdots, 2^N - 1)$ are as follows [14].

$$ave = \sum_{j=0}^{B-1} ave_j \approx 0 \tag{56}$$

$$var = \sum_{j=0}^{B-1} var_j \tag{57}$$

The next consideration is the non-diagonal elements of the auto-correlation matrix. From Eq. (28), the elements of the extended input signal vector satisfy the following relation.

$$s_{DA,0}(k)+s_{DA,1}(k)+\cdots+s_{DA,2^N-1}(k) \approx 0 \quad (58)$$

To obtain the product of $s_{DA,0}(k)$ and $s_{DA,1}(k)$, multiplying both sides by $s_{DA,1}(k)$, Eq. (58) becomes

$$\begin{aligned} &s_{DA,0}(k)s_{DA,1}(k) \\ &\quad \approx -\{s_{DA,1}(k)s_{DA,1}(k) + \\ &\qquad \cdots + s_{DA,2^N-1}(k)s_{DA,1}(k)\}. \end{aligned}$$

The expectation of this equation is

$$\begin{aligned} &E[s_{DA,0}(k)s_{DA,1}(k)] \\ &\quad \approx -E[s_{DA,1}(k)s_{DA,1}(k)] - \\ &\qquad \cdots - E[s_{DA,2^N-1}(k)s_{DA,1}(k)]. \end{aligned} \quad (59)$$

The each element of Eq. (28) is determined by the pattern of the access matrix. However all the patterns occur in equal probability at random, so that

$$E[s_{DA,i}(k)s_{DA,j}(k)] = cor, \ i \neq j.$$

Therefore Eq. (59) becomes

$$cor \approx -var - (2^N - 2) \times cor,$$

finally, cor is obtained as

$$cor \approx -\frac{var}{2^N - 1}. \quad (60)$$

From the above discussion, $\boldsymbol{R}$ is $2^N \times 2^N$ matrix having the diagonal elements of var and the non-diagonal elements of cor.

Furthermore $\boldsymbol{R}$ can be modified to

$$\boldsymbol{R} = \boldsymbol{D} + \boldsymbol{Q},$$

where

$$\boldsymbol{D} = diag[d_0, d_1, \cdots, d_{2^N-1}],$$

$$\boldsymbol{Q} = \begin{bmatrix} q & \cdots & q \\ \vdots & \ddots & \vdots \\ q & \cdots & q \end{bmatrix}^T,$$

and the operator $diag[\,]$ indicates the diagonal elements of the specified matrix. The diagonal elements of $\boldsymbol{R}$ are

$$E[s_{DA,i}(k)s_{DA,i}] = var, \quad i = 0, 1, \cdots, 2^N - 1,$$

and the non-diagonal elements are

$$cor \approx -\frac{var}{2^N - 1},$$

so that the elements of $\boldsymbol{D}$ and $\boldsymbol{Q}$ are

$$d = d_i = var + \frac{var}{2^N - 1} = \frac{2^N \times var}{2^N - 1} \quad i = 0, 1, \cdots, 2^N - 1$$

and

Table 3 Comparison of eigenvalue for our proposed DA-ADF.

Tap number	Theory	Simulation
2	1.333(3),0.000(1)	1.333(3),0.000(1)
3	1.500(7),0.000(1)	1.500(7),0.000(1)
4	1.333(15),0.000(1)	1.333(15),0.000(1)
5	1.042(31),0.000(1)	1.042(31),0.000(1)

$$q = -\frac{var}{2^N - 1},$$

respectively. All the eigenvalues of $\boldsymbol{D}$ equal to the diagonal element d. The trace of $\boldsymbol{Q}$ equals to the sum of the eigenvalues,

$$tr[\boldsymbol{Q}] = -2^N \times \frac{var}{2^N - 1},$$

where $tr[\boldsymbol{Q}]$ indicates the trace of $\boldsymbol{Q}$. Furthermore the rank of $\boldsymbol{Q}$ is the size of non-singular sub-matrix, so that

$$rank[\boldsymbol{Q}] = 1.$$

From the above discussion, one eigenvalue of $\boldsymbol{Q}$ equals to $tr[\boldsymbol{Q}]$ and others are "0." Therefore the eigenvalues of $\boldsymbol{R}$ are

$$\begin{aligned} eig[\boldsymbol{R}] &= eig[\boldsymbol{D}] + eig[\boldsymbol{Q}] \\ &= \left[0, \frac{2^N \times var}{2^N - 1}, \cdots, \frac{2^N \times var}{2^N - 1}\right]^T, \end{aligned} \quad (61)$$

where

$$\frac{2^N \times var}{2^N - 1} = \frac{4}{3}N^2(1-2^{-N+1}+2^{-N})(2^N-1)^{-1},$$

and the operator $eig[\,]$ gives the eigenvalues of the specified matrix. Table 3 shows the comparison of the eigenvalues obtained by the theory and the computer simulations, where the numbers of eigenvavlues is represented in the parenthesis. The size of the auto-correlation matrix $\boldsymbol{R}$ is $2^N \times 2^N$, so that 2^N eigenvalues exist. The input signal $s(k)$ was zero-mean white signal with a variance of 0.333 having uniform distribution and was represented in the two's complement representation with word length of 16 bits as follows.

$$s(k) = [b_0(k), b_1(k), \cdots, b_{15}(k)]\,\boldsymbol{F}$$
$$\boldsymbol{F} = [-2^0, 2^{-1}, \cdots, 2^{-15}]^T$$

The auto-correlation matrix $\boldsymbol{R}$ is ensemble average of 100 independent processes over 10^5 iterations. The eigenvalues of $\boldsymbol{R}$ are obtained by MATLAB ver.6. The simulation results quite agree with the theoretical results, and one zero eigenvalue and the same 2^N–1 eigenvalues exist. Furthermore one zero eigenvalue is theoretically zero, so that the rank of $\boldsymbol{R}$ equals to $2^N - 1$ and our proposed DA-ADF has all the same eigenvalues. Thus, in the adaptation process, the selected step-size parameter guarantees to converge for all the same eigenvalues, so that our proposed DA-ADF has a good convergence speed.

Table 4 Comparison of eigenvalue for the conventional method.

Tap number	Theory	Simulation
2	0.333(3) ,1.000(1)	0.333(3) ,1.000(1)
3	0.375(7) ,1.125(1)	0.375(7) ,1.125(1)
4	0.333(15),1.000(1)	0.333(15),1.000(1)
5	0.260(31),0.781(1)	0.260(31),0.781(1)

4.2 Eigenvalues of the Conventional Method

We can obtain the eigenvalues of the auto-correlation matrix in the same manner of our proposed method. The average ave' and the variance var' of the $s'_{DA,i}(k)$ are as follows.

$$ave' \approx N \times 2^{-N}, \tag{62}$$

$$var' = \sum_{j=0}^{B-1} var'_j. \tag{63}$$

Thus the each element of the extended input signal has the average $ave'(\neq 0)$.

The non-diagonal elements cor' is

$$cor' \approx -\frac{var' + ave'^2 - N \times ave'}{2^N - 1}. \tag{64}$$

From these results, $\boldsymbol{R}'$ is $2^N \times 2^N$ matrix having diagonal elements of $var' + ave'^2$ and non-diagonal elements of cor'. The eigenvalues of $\boldsymbol{R}'$ are

$$\begin{aligned} eig[\boldsymbol{R}'] &= eig[\boldsymbol{D}'] + eig[\boldsymbol{Q}'] && (65)\\ &= [d', \cdots, d']^T + [2^N \times q', 0, \cdots, 0]^T \\ &= [d' + 2^N \times q', d', \cdots, d']^T \\ &= [N \times ave', d', \cdots, d']^T, && (66)\end{aligned}$$

where

$$N \times ave' = N^2 \times 2^{-N} \tag{67}$$

and

$$d' = \frac{1}{3}N^2(1 - 2^{-N+1} + 2^{-N})(2^N - 1)^{-1}. \tag{68}$$

Table 4 shows the comparison of the eigenvalues obtained by the theory and the computer simulations, where the number in the parenthesis indicates the number of the eigenvalue. The size of the auto-correlation matrix $\boldsymbol{R}$ is $2^N \times 2^N$, so that 2^N eigenvalues exist. The input signal $s'(k)$ was zero-mean white signal with a variance of 0.333 having uniform distribution and was represented in the offset binary representation with word length of 16 bits indicated as follows.

$$s'(k) = [b'_0(k), b'_1(k), \cdots, b'_{15}(k)]\,\boldsymbol{F}'$$
$$\boldsymbol{F}' = [2^{-1}, 2^{-2}, \cdots, 2^{-16}]^T$$

The auto-correlation matrix $\boldsymbol{R}$ is ensemble average of 100 independent processes over 10^5 iterations. The eigenvalues of $\boldsymbol{R}$ are obtained by MATLAB ver.6. The simulation results quite agree with the theoretical results, and the one large eigenvalue and the same $2^N - 1$ eigenvalues exist. The large eigenvalue is caused by the offset bias of the $s'_{DA,i}(k)$. Figure 2 shows the theoretical convergence characteristics for $N = 4$ and the step size parameter

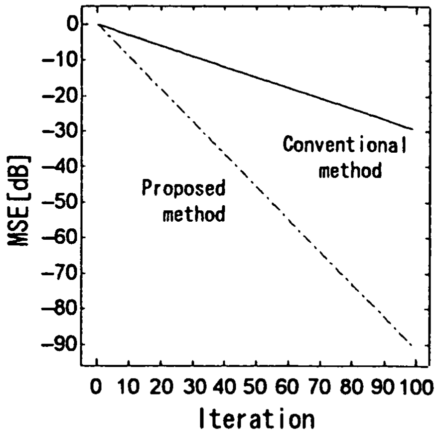


Fig. 2 Comparison of convergence speed using convergence equation.

$$\mu = 0.1 \times \frac{1}{\lambda_{max}},$$
$$\mu' = 0.1 \times \frac{1}{\lambda'_{max}}.$$

From these results, our proposed DA-ADF has good convergence speed, whereas the conventional DA-ADF degrades extremely.

5. Conclusions

In this paper, we have derived the convergence condition of DA-ADF theoretically and evaluated the convergence speed of our proposed and the conventional method. To analyze, (1) we extended the update and output equation to the whole AFS and defined the extended input signal vector, (2) we defined the estimation error as the difference between the optimum value and the estimate and (3) we derived the convergence condition as the condition for the estimation error to decrease for increasing time. From our extension, we found out the new understanding that the whole AFS is updated by using the extended input signal vector. As a result, the convergence condition is depended on eigenvalues of the auto-correlation matrix of the extended input signal vector. Furthermore, we obtained the eigenvalues theoretically. In our proposed method, all eigenvalues are equal, so that the convergence speed is very fast. However, in the conventional method, one eigenvalue becomes very large because the input signal coded in offset binary include the offset bias, so that the convergence speed degrades extremely.

Considerations on the convergence condition of the non-stationary and the colored input signal are considered as a future work.

References

[1] B. Widrow, J.R. Glover, Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong, Jr., and R.C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proc. IEEE, vol.63, pp.1692–1716, Dec. 1975.

[2] A. Peled and B. Liu, "A new hardware realization of digital filters," IEEE Trans. Acoust., Speech & Signal Process., vol.22, no.12, pp.456–462, Dec. 1974.

[3] C.F.N. Cowan and J. Mavor, "New digital adaptive filter implementation using distributed-arithmetic techniques," IEE Proc., vol.128, Pt.F, no.4, pp.225–230, Aug. 1981.

[4] C.H. Wei and J.J. Lou, "Multimemory block structure for implementing a digital adaptive filter using distributed arithmetic," IEE Proc., vol.133, Pt.G, no.1, pp.19–26, Feb. 1986.

[5] C.F.N. Cowan, S.G. Smith, and J.H. Elliott, "A digital adaptive filter using a memory-accumulator architecture: Theory and realization," IEEE Trans. Acoust., Speech & Signal Process., vol.31, no.3, pp.541–549, June 1983.

[6] Y. Tsunekawa, K. Takahashi, S. Toyoda, and M. Miura, "High-performance VLSI architecture of multiplierless LMS adaptive filters using distributed arithmetic," IEICE Trans. Fundamentals (Japanese Edition), vol.J82-A, no.10, pp.1518–1528, Oct. 1999.

[7] K. Takahashi, Y. Tsunekawa, S. Toyoda, and M. Miura, "High-performance architecture of LMS adaptive filters using distributed arithmetic based on half-memory algorithm," IEICE Trans. Fundamentals, vol.J84-A, no.6, pp.777–787, June 2001.

[8] K. Takahashi, Y. Tsunekawa, N. Tayama, and K. Seki, "Analysis of the convergence condition of LMS adaptive filter using distributed arithmetic," Proc. ITC-CSCC'01, vol.1, pp.576–579, July 2001.

[9] K. Takahashi, Y. Tsunekawa, N. Tayama, and K. Seki, "Convergence condition analysis of LMS adaptive filters using distributed arithmetic," Proc. 187-th SICE Tohoku-Branch Research Convention, vol.187-3, June 2000.

[10] C.F.N. Cowan and P.M. Grant, Adaptive Filters, Prentice-Hall, New Jersey, 1985.

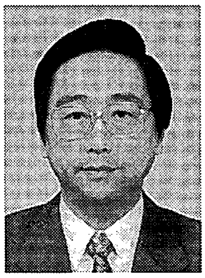
[11] S. Haykin, Introduction to Adaptive Filters, Macmillan Publishing Company, New York, 1984.

[12] K. Matsubara, K. Nishikawa, and H. Kiya, "Pipelined adaptive filters based on delayed LMS algorithm," IEICE Trans. Fundamentals (Japanese Edition), vol.J79-A, no.5, pp.1050–1057, May 1996.

[13] A. Harada, K. Nishikawa, and H. Kiya, "Pipelined architecture of the LMS adaptive digital filter with the minimum output latency," IEICE Trans. Fundamentals, vol.E81-A, no.8, pp.1578–1585, Aug. 1998.

[14] L. Maisel, Probability, Statics and Random Process, Simon & Schuster, New York, 1971.

Kyo Takahashi received B.E. and M.E. degrees from Faculty of Engineering of Iwate University, Japan, in 1987 and 1989. In 1987, 1994 and 1998, he joined Japan Radio Corporation, Iwate High-Tech Junior College and Iwate Industrial Junior College, respectively. He is now working toward the D.E. degree in Iwate University. His research interests are in the areà of adaptive digital signal processing and their implementations.



Yoshitaka Tsunekawa received the B.E. degree from Iwate University, Morioka, Japan, in 1980 and the M.E. and the Doctor of Engineering degrees from Tohoku University, Sendai, Japan, in 1983 and 1993, respectively. Since 1983, he has been with the Faculty of Engineering, Iwate University, where he is now an Associate Professor at the Department of Electrical and Electronic Engineering. His current research interests include digital signal processing and digital control. He is a member of the Society of Instrument and Control Engineers of Japan, and IEEE.

Norio Tayama graduated from Iwate University, Japan, in 1966. He received the M.S. degree and the Ph.D. degree from Tohoku University, in 1968 and 1978, respectively. In 1968 he joined Nippon Signal Co. In 1972 he was a lecturer of Dept. of Electronics, Iwate University. In 1972 he was associate professor of Dept. of Computer Science, and he is currently a professor and the head of Dept. of Electrical Engineering and Electronics, Iwate University. His research interests are image processing, signal processing and parallel computer architecture. He is a member of IEEE.

Kyoushirou Seki received the B.E. degree from Iwate University, Morioka, Japan, in 1960, and the Doctor of Engineering degree from Tohoku University, Sendai, Japan, in 1987. Since 1964, he has been with the Faculty of Engineering, Iwate University, where he is now a professor at the Department of Electrical and Electronic Engineering. His current research interests include magnetic semiconductor materials and applications. He is a member of the Society of Instrument and Control Engineers, the Society of Applied Magnetics of Japan, and IEEE.