PAPER *Special Section of Papers Selected from the 8th Digital Signal Processing Symposium*

An LS Based New Gradient Type Adaptive Algorithm ——Least Squares Gradient——*

Kiyoshi NISHIKAWA† *and* Hitoshi KIYA†, *Members*

SUMMARY A new gradient type adaptive algorithm is proposed in this paper. It is formulated based on the least squares criteria while the conventional gradient algorithms are based on the least mean square criteria. The proposed algorithm has two variable parameters and by changing them we can adjust the characteristic of the algorithm from the RLS to the LMS depending on the environment. This capability of adjustment achieves the possibility of providing better solutions. However, not only it provides better solutions than the conventional algorithms under some conditions but also it provides a very interesting theoretical view point. It provides a unified view point of the adaptive algorithms including the conventional ones, i.e., the LMS or the RLS, as limited cases and it enables us to analyze the bounds for those algorithms.
key words: *adaptive filters, least squares, system identification, spectral estimation*

1. Introduction

There are two types of algorithm for adaptive signal processing [1], [2], namely gradient type [3]–[6] and recursive type [7]–[10]. The LMS (least mean square) algorithm [3] is the typical one for the former and the RLS (recursive least squares) algorithm [1] for the later. These two types of algorithms have dual properties in each other [2]. That is, the gradient algorithms have slower convergence property but numerical stability in noisy environments and the recursive algorithms have faster convergence but numerical instability. Another desired feature of adaptive algorithms is tracking property for non-stationary environments as environments where algorithms are applied are normally varying with time. The algorithm which has all of the faster convergence property, numerical stability, and tracking property is desired.

To improve the numerical stability of the recursive type algorithm is very difficult for their recursive form of updating of an adaptive filter [11]. Furthermore, the exponential weighting, which is used by recursive type algorithms to ensure the tracking property, prevents the algorithms to converge to the optimum solution in the least squares sense although we cannot avoid the weighting. Therefore, acceleration of the gradient type algorithms has a significant meaning. To improve the convergence speed of the algorithms of this type, many researches have been done so far [12]–[17]. For example, the literature [12] and [13] realize the faster convergence speed by varying the step size parameter according to the state of the adaptive filter.

In this paper, we propose another method for improving gradient type algorithms. We propose a gradient type algorithm based on the least squares (LS) criteria although ordinary gradient algorithms are based on the least mean square (MSE) criteria. While the algorithms based on MSE must estimate the gradient vector, the algorithms based on LS can calculate the correct gradient vector. The conventional steepest descent algorithm base on LS criteria is one of such algorithms and its properties are considered in the literature [18]–[20]. However, the algorithm is normally formulated only for static applications and not for applications such as adaptive noise canceller [21] which requires real time processing. Also, the tracking property of the steepest descent algorithm merely mentioned. Therefore, extensions of the steepest descent algorithm for being able to apply to non-stationary applications are needed. The proposed algorithm is one of such algorithms.

Before considering the extensions, we first review the steepest descent algorithm under static conditions. Although analyses mentioned in the literature treat mainly on the convergence property under noise free case, we will consider a relation between the effect of noise and the number of iteration. From the consideration, it is shown that, under the low SNR (signal to noise ratio), the solutions of the normal equations does not the optimum ones in the point of coefficients error. The recursive type algorithms cannot realize the solutions because they solve the normal equations directly. Only the gradient type algorithms can realize this intermediate solution.

Then we extend the algorithm based on the previous consideration and formulate the proposed algorithm which is able to be applied to non-stationary applications such as adaptive signal processing. The proposed algorithm has two adjustable parameters and it is shown that we can adjust the performance characteristic of the algorithm from the LMS to the RLS by changing the two parameters. This gives the generalized view point of the adaptive algorithms including the LMS and the RLS as the special cases of the proposed algorithm. Conditions for convergence are considered

Manuscript received January 31, 1994.
Manuscript revised April 14, 1994.
†The authors are with the Faculty of Technology, Tokyo Metropolitan University, Hachioji-shi, 192–03 Japan.
*This paper was presented at the 8th DSP Symposium.

and the available range of the step size parameter is derived. Further, we show a method for adjusting the value of the step size parameter according to the variation of a input signal which can be thought to be a generalized version of the normalized LMS algorithm.

The rest of the paper is organized as follows. In Sect. 2 we formulate the method of least squares and give a brief derivation of the steepest descent algorithm. In Sect. 3, we describe the proposed algorithm and the normalized representation for the algorithm is also proposed. Then in Sect. 4 we show the results of computer simulations to show the validity of the proposed algorithm.

2. The Steepest Descent Algorithm

In this section, we show the relation between the effect of noise and the number of iteration under the least squares criteria. We show that there exists the optimum number of iteration in respect to IRER (impulse response error ratio) under low SNR conditions. For that, we first review the method of least squares and the steepest descent algorithm. Then, we consider the relation between the effect of noise and the number of iteration based on the previous study.

In the following, we assume signals are real and filters have real coefficients and consider only the case where filters are transversal ones.

2.1 The Method of Least Squares

First, we formulate the method of least squares [22]. Suppose we have two sets of variables, $\{d(i)\}$ and $\{u(i)\}$. The variable $d(i)$ is observed at time i in response to the subset of variables $u(i)$, $u(i-1)$,...,$u(i-M+1)$ applied as inputs. This functional relationship is assumed to be linear. Then we can regard $d(i)$ as a output of some unknown filter with $\{u(i)\}$ as inputs. In Fig. 1 we show a model of adaptation.

Let us consider to estimate the parameters of the unknown system using the method of least squares. To do this, we postulate a linear transversal filter of length M as a model. Then we define the error $e(i)$ at time i as the difference between $d(i)$ and the filter's output $y(i)$, as shown by

$$e(i) = d(i) - y(i). \tag{1}$$

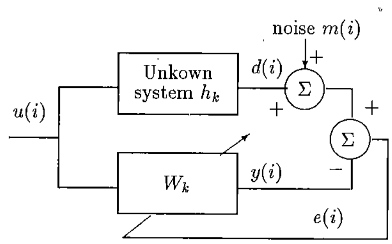


Fig. 1 Adaptive identification of an unknown system.

In this equation $y(i)$ is given by

$$y(i) = \sum_{k=0}^{M-1} w_k u(i-k), \tag{2}$$

where w_k shows the kth coefficient of the filter.

The optimum solution in least squares sense is the one which minimizes the sum of error squares ε defined as

$$\varepsilon(L) = \sum_{i=M}^{L} e^2(i), \tag{3}$$

where L is the length of data used for estimation. Using (1) and (2), (3) can be expressed as

$$\varepsilon(L) = E_d - 2\sum_{k=0}^{M-1} w_k\theta(-k) + \sum_{k=0}^{M-1}\sum_{t=0}^{M-1} w_k w_t \phi(k,t) \tag{4}$$

where we used the following definitions [2] for the summation terms in (4) that involve the variable i:

1. The deterministic auto-correlation of the input signals.

$$\phi(k,t) = \sum_{i=M}^{L} u(i-k)u(i-t) \tag{5}$$

2. The deterministic cross-correlation between the desired response and the input signal.

$$\theta(-k) = \sum_{i=M}^{L} d(i)u(i-k) \tag{6}$$

3. The energy of the desired response.

$$E_d = \sum_{i=M}^{L} d^2(i) \tag{7}$$

One way to obtain the optimum solution which gives the minimum $\varepsilon(L)$ is to solve the system of simultaneous equations [2],

$$\sum_{t=0}^{M-1} w_t\phi(t,k) = \theta(-k). \tag{8}$$

These equations are called *the system of the normal equations* for a linear least-squares filter. The RLS is the algorithm which recursively solves these equations with every new sample.

Another way to obtain the optimum solutions is using the gradient vector of $\varepsilon(L)$ [18]. Let $\nabla[\varepsilon(L)]$ denotes the M-by-1 gradient vector. The kth element of

$\nabla[\varepsilon(L)]$, by definition, equals the first derivative of the sum of squared errors:

$$\nabla_k[\varepsilon(L)] = \frac{\partial\varepsilon(L)}{\partial w_k} = -2\sum_{i=M}^{L} e(i)u(i-k). \tag{9}$$

This equation shows the direction of the gradient vector towards the minimum point. The steepest descent algorithm based on the least squares criteria utilizes (9) to obtain the optimum filter.

2.2 Formula for Update Recursion

Let us derive the steepest descent algorithm [18], [19] which gives the optimum filter coefficients in the LS sense. We must introduce a new index n to denotes the number of iteration which is independent of time i because the gradient algorithms obtain the optimum solution after some iterations. The error and the filter output become functions of n and are defined as

$$e(i;n) = d(i) - y(i;n), \quad \text{and} \tag{10}$$

$$y(i;n) = \sum_{k=0}^{M-1} w_k(n)u(i-k) \tag{11}$$

respectively where $w_k(n)$ shows the kth coefficient of the ADF at iteration n. From these equations $\varepsilon(L;n)$, the sum of squared errors at iteration n, is given by

$$\varepsilon(L;n) = E_d - 2\sum_{k=0}^{M-1} w_k(n)\theta(-k) + \sum_{k=0}^{M-1}\sum_{t=0}^{M-1} w_k(n)w_t(n)\phi(k,t). \tag{12}$$

Then the gradient vector at nth iteration is given as

$$\nabla_k[\varepsilon(L;n)] = -2\sum_{i=M}^{L} e(i;n)u(i-k) \qquad k = 0,\cdots,M-1 \tag{13}$$

Using (13), we can update the coefficients of an ADF by

$$w_k(n+1) = w_k(n) + \nu\nabla_k[\varepsilon(L;n)] \qquad k = 0,\cdots,M-1 \tag{14}$$

where ν is a step size parameter. We repeat Eqs. (10) to (14) until we obtain the optimum solution.

Let us introduce the vector notation for later convenience. We denote the M-by-1 tap-input vector $\mathbf{u}(i)$ as

$$\mathbf{u}(i) = [u(i), u(i-1), \cdots, u(i-M+1)]^T \tag{15}$$

and M-by-1 filter coefficient vector $\mathbf{w}(i)$ as

$$\mathbf{w}(i) = [w_0(i), w_1(i), \cdots, w_{M-1}(i)]^T \tag{16}$$

respectively. Then (14) can be written in a vector form as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \nu\nabla[\varepsilon(L;n)] = \mathbf{w}(n) + \nu\sum_{i=M}^{L} e(i;n)\mathbf{u}(i). \tag{17}$$

2.3 Consideration on the Convergence Property

In this section, we see the convergence property of the steepest descent algorithm in stationary situation [3]. First we define the coefficient error between the optimum filter and the ADF at iteration n as

$$\boldsymbol{\epsilon}(n) = \mathbf{w}(n) - \mathbf{w}_o \tag{18}$$

where $\mathbf{w}_o$ shows the coefficients of the optimum filter in the least squares sense.

From (10)–(14) we get the next recursion for $\mathbf{w}(n)$:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \nu\sum_{i=M}^{L}\mathbf{u}(i)[d(i) - \mathbf{u}^T(i)\mathbf{w}(n)]. \tag{19}$$

Subtracting $\mathbf{w}_o$ from both sides of (19) yields

$$\begin{aligned}\boldsymbol{\epsilon}(n+1) &= [\mathbf{I} - \nu\sum_{i=M}^{L}\mathbf{u}(i)\mathbf{u}^T(i)]\mathbf{w}(n) + \nu\sum_{i=M}^{L}\mathbf{u}(i)d(i) - \mathbf{w}_o \\ &= [\mathbf{I} - \nu\sum_{i=M}^{L}\mathbf{u}(i)\mathbf{u}^T(i)]\boldsymbol{\epsilon}(n) + \nu\sum_{i=M}^{L}\mathbf{u}(i)e_{\min}(i) \\ &= [\mathbf{I} - \nu\boldsymbol{\Phi}]\boldsymbol{\epsilon}(n) + \nu\sum_{i=M}^{L}\mathbf{u}(i)e_{\min}(i)\end{aligned} \tag{20}$$

where $\boldsymbol{\Phi}$ is

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(0,0) & \cdots & \phi(M-1,0) \\ \phi(0,1) & \cdots & \phi(M-1,1) \\ \vdots & \ddots & \vdots \\ \phi(0,M-1) & \cdots & \phi(M-1,M-1) \end{bmatrix} \tag{21}$$

and $e_{\min}$ is given by

$$e_{\min}(i) = d(i) - \mathbf{u}^T(i)\mathbf{w}_o \tag{22}$$

which shows the minimum estimation error over the values of time i in the interval (M, N). For the study of the convergence, the second term in the right hand side of Eq. (20) will be play the important roll. Let us define φ as

$$\varphi = \sum_{i=M}^{L}\mathbf{u}(i)e_{\min}(i) \tag{23}$$

and consider the convergence property of the algorithm in the two cases i) $\varphi = \mathbf{0}$ and ii) $\varphi \neq \mathbf{0}$ where $\mathbf{0}$ shows a zero-vector.

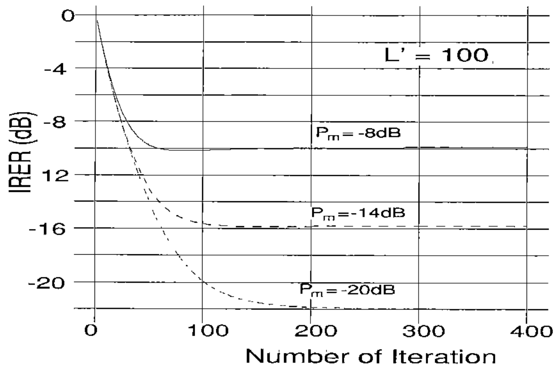


Fig. 2 The relation between noise and number of iteration. Number of input vectors is fixed to 100.

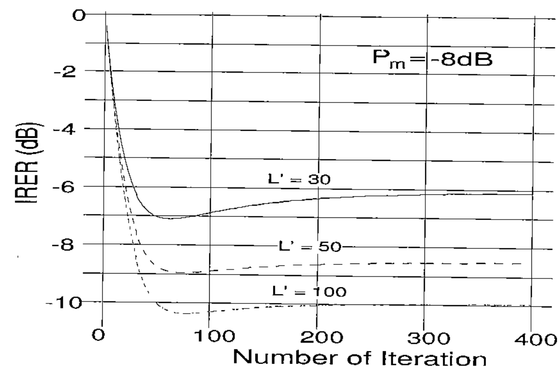


Fig. 3 The relation between noise and number of iteration. The power of noise is fixed to −8 dB.

2.3.1 The Case of $\varphi = \mathbf{0}$

When $\varphi = \mathbf{0}$ Eq. (20) becomes

$$\epsilon(n+1) = [\mathbf{I} - \nu\mathbf{\Phi}]\epsilon(n) \tag{24}$$

and is the same form for the steepest descent algorithm based on the least mean square criteria [3]. We can show the error $\epsilon(n)$ approaches to zero as n approaches infinity if we choose the suitable value for ν. By applying the study for the convergence properties of the steepest descent algorithm based on the least mean square criteria, we obtain the well known condition [3] for the value of ν:

$$0 < \nu < \frac{2}{\lambda_{\max}} \tag{25}$$

where $\lambda_{\max}$ is the largest eigenvalue of Φ. We can say that Eq. (24) converges when the value of ν is in this range.

2.3.2 The Case of $\varphi \neq 0$

Next, let us consider the effect of φ. Normally, this term is treated as zero from the principle of orthogonality [2]. However, when SNR is low or the interval of data is short the effect of φ will not be negligible. The recursion for the coefficients estimation error will be

$$\epsilon(n+1) = (\mathbf{I} - \nu\mathbf{\Phi})\epsilon(n) + \nu\varphi \tag{26}$$

To give the general analysis of this equation is difficult because it depends on both of the input signal and the statistical properties of the $e_{\min}$.

However, we can calculate Eq. (26) under certain conditions. We show two results of calculation in Figs. 2 and 3 under the conditions:

- Input signal is a white Gaussian process with variance $\sigma = 1$. We denote the number of input vectors by L' which has a relation $L' = L - M + 1$.
- $e_{\min}$ is assumed to be a white Gaussian process with variation σ_m. This assumption corresponding to add $m(n)$, see Fig. 1, of σ_m to the desired signal.
- The unknown system, or the optimum system in least squares sense, $\mathbf{w_o}$ is a FIR low pass filter with 20th order.
- Initial value for $\epsilon(0)$ is set to be $\epsilon(0) = -\mathbf{w}_o$. This assumption corresponding to set $\mathbf{w}(0) = \mathbf{0}$.
- We set the value for ν as $\nu = \frac{1}{\sum_{i=M}^{L} \mathbf{u}(i)^T\mathbf{u}(i)}$. The validity of this selection will be given in later.

As a reference of comparison we use the impulse response estimation ratio (IRER) which is defined as

$$10\log_{10}\frac{\epsilon(n)^T\epsilon(n)}{\mathbf{w}_o^T\mathbf{w}_o} \tag{27}$$

The results shown in Fig. 2 are calculated with various P_m, the power of $e_{\min}$, with fixing L' to 100 and they are averages of 100 independent processes. From this figure we know that as P_m getting higher, i.e. $P_m = -8$ dB, there exists a minimum respect to the number of iteration. To see this more precisely, we show the results of calculation with varying L' and Fixing P_m to −8 dB in Fig. 3. This figure suggests that there exists an optimum number of iteration with respect to IRER. Noting that the solutions provided by the recursive type algorithms correspond to the solutions provided by the steepest descent algorithm after infinite iterations with suitably selected ν. Therefore, only the gradient type algorithms realize the minimum of IRER.

3. The Least Squares Gradient Algorithm

In this section, a new gradient type adaptive algorithm which is the generalized version of the steepest descent algorithm is proposed. We refer to the proposed algorithm as the least squared gradient algorithm. We will show that the proposed algorithm has two parameters to control the performance of the algorithm, i.e., the speed of convergence, tracking abilibty, and the stability under noisy environments.

3.1 Problems with the Conventional LS Based Algorithms

Before implementing the proposed algorithm, let us consider the problems of the method of least squares.

When we use recursive type algorithms based on LS criteria, e.g., the RLS algorithm, the deterministic quantities as shown in Eqs. (5) and (6) are recursively updated in order to obtain solutions. This means that the algorithms use all the data obtained up to now for adaptation and this makes it impossible to adapt non-stationary states. To overcome this situation the forgetting factor, or exponentially weighting, is used to *forget* the past data in the RLS or other recursive type algorithms. However, the influence of the past data cannot perfectly removed even if we use the forgetting factor for the conventional algorithms.

There is another problem with the conventional LS based algorithms. When adaptive algorithms are applied in applicaitons under low SNR conditions, there exists the optimum number of iteration as shown in Sect. 2.3.2. However, the conventional recursive type algorithms have no mean to realize it, because they always search the solutions which correspod to the solutions after infinite iterations of the steepest descent algroithm.

In the following discussion we show that the proposed algorithm has the adjustable two parameters to solve the above mentioned problems.

3.2 The Basic Idea for Extension

To implement the proposed algorithm we must modify the steepest descent algorithm to have the formulation which enables us to update the filter coefficients as a new sample is obtained.

The basic idea for extending the steepest descent algorithm for non-stationary environments is to repeat the algorithm described in Sect. 2 for each time i so that we have the optimum solution for each i. The sum of error squares ε at time i is defined as

$$\varepsilon(i) = \sum_{m=M}^{i} e^2(m). \tag{28}$$

The ADF's coefficients at nth iteration of time i, we denote it as $\mathbf{w}(i;n)$, are updated as

$$\mathbf{w}(i;n+1) = \mathbf{w}(i;n) + \nabla[\varepsilon(i;n)] \tag{29}$$

where $\nabla[\varepsilon(i;n)]$ denotes the gradient vector at nth iteration of time i and is expressed as

$$\nabla[\varepsilon(i;n)] = \nu \sum_{\ell=M}^{i} e(\ell;n)\mathbf{u}(\ell) \tag{30}$$

where ℓ is a dummy index. We can choose any vector as the initial value for the tap-weight vector at time i. It might be set

$$\mathbf{w}(i+1;0) = \mathbf{0} \tag{31}$$

to fit the least squares sense.

However, this procedure is not a good one in practical view point because of its requirement for very amount of calculation and of no tracking ability for varying environment.

3.3 The Proposed Procedure

We introduce a technic, which is resemble to the forgetting factor of the recursive type algorithms. We propose to use only several terms in (29), say P, for updating the coefficients and P is one of the two parameters of the algorithm. Although this is similar to introduce the forgetting factor, they have different effect on the adaptive process. We can perfectly limit the effect of the past data on adaptive process by changing P but the forgetting factor only weights the past data so that the effect of the past will remain.

We must modify the definition of the sum of error square ε as

$$\varepsilon^{(P)}(i) = \sum_{m=i-P+1}^{i} e^2(m) \tag{32}$$

to fit the meaning of the technic where the superscript (P) shows that the values are calculated using P-point least squares. Using this definition, the Eq. (29) is modified as

$$\mathbf{w}^{(P)}(i;n+1) = \mathbf{w}^{(P)}(i;n) + \nabla^{(P)}[\varepsilon^{(P)}(i;n)]. \tag{33}$$
$$n = 0, 1, \cdots, N$$

where N shows the number of iteration at time i and is the second parameter of the algorithm. As shown in Sect. 2.3.2, we can control the effect of noise by changing N and there are no counterpart of N in the conventional recursive type algorithms so that we cannot control the effect of noise with them.

As the initial condition for each time i, we propose to use

$$\mathbf{w}(i+1;0) = \mathbf{w}(i;N) \tag{34}$$

instead of using (31). Noting that the gradient vector $\nabla^{(P)}[\varepsilon^{(P)}(i;n)]$ at nth iteration at time i is given by

$$\nabla^{(P)}[\varepsilon^{(P)}(i;n)] = \nu \sum_{\ell=i-P+1}^{i} e(\ell;n)\mathbf{u}(\ell). \tag{35}$$

Here we have shown that the proposed algorithm has two parameters P and N. P limits the effect of the past data and N controls both of the convergence speed and the effect of noise. Under these representation, we can treat the conventional algorithms as limited cases of the proposed algorithm. The LMS algorithm can be regarded as the one point least squares algorithm, $P = 1$, with only one iteration, $N=1$, for each time i and the RLS algorithm can be thought of as the case of $P=i$ and $N=\infty$. This interpretation gives a unified view point of the adaptive algorithms and provides a way for analyzing the performance bounds of them. Moreover, we can realize the intermediate performance characteristic by suitably choosing the parameters which we cannot obtain with the conventional algorithms.

3.4 Consideration on the Convergence Property

Here let us consider about the convergence property of the proposed algorithm. When we use Eq. (32) as the definition of $\varepsilon^{(P)}(i)$ then the correlation matrices at time i are defined as follows:

1. The deterministic auto-correlation of the input signals.

$$\Phi^{(P)}(i) = \sum_{\ell=i-P+1}^{i} \mathbf{u}(\ell)\mathbf{u}(\ell)^T \tag{36}$$

2. The deterministic cross-correlation between the desired response and the input signal.

$$\Theta^{(P)}(i) = \sum_{\ell=i-P+1}^{i} d(\ell)\mathbf{u}(\ell). \tag{37}$$

From the similar study as in Sect. 2.3.2, we can easily show that the condition for convergence at time i is given by

$$0 < \nu < \frac{2}{\lambda_{\max}(i)} \tag{38}$$

where $\lambda_{\max}(i)$ is the largest eigenvalue of the correlation matrix $\Phi^{(P)}(i)$.

3.5 The Normalized Least Squares Gradient

Let us consider the method to automatically determine the step size parameter ν. In the previous discussion we showed that the algorithm will converge when the step size parameter ν is in the range

$$0 < \nu < \frac{2}{\lambda_{\max}(i)}. \tag{39}$$

However, when the property of $\Phi^{(P)}(i)$ is varying then constant ν may lead the algorithm to diverge so that ν must be varying according to the variation of $\Phi^{(P)}(i)$.

Since all the eigenvalues of $\Phi^{(P)}(i)$ is real and non-negative [2], we can say that

$$\sum \lambda(i) > \lambda_{\max}(i) \tag{40}$$

where summation in the left hand side shows the sum of all the eigenvalues of $\Phi^{(P)}(i)$. Using the property of the trace of a matrix, we obtain the next relation:

$$\begin{aligned}\sum \lambda &= \mathrm{Tr}\ \Phi^{(P)}(i) \\ &= \mathrm{Tr} \left\{ \sum_{\ell=i-P+1}^{i} \mathbf{u}(\ell)\mathbf{u}(\ell)^T \right\} \\ &= \sum_{\ell=i-P+1}^{i} \mathrm{Tr}\ \left\{\mathbf{u}(\ell)\mathbf{u}(\ell)^T\right\} \\ &= \sum_{\ell=i-P+1}^{i} \mathbf{u}(\ell)^T\mathbf{u}(\ell)\end{aligned} \tag{41}$$

where Tr stands for the trace of a matrix. This relation shows that if we determine the value of ν as

$$\nu = \frac{\alpha}{\sum_{\ell=i-P+1}^{i} \mathbf{u}(\ell)^T\mathbf{u}(\ell)} \tag{42}$$

$$\text{where } 0 < \alpha < 2$$

then we always satisfy the condition (38) automatically. Noting that the Eq. (42) can be viewed as the generalization of the normalized LMS algorithm in which the step size parameter μ is calculated by

$$\mu = \frac{\alpha}{\mathbf{u}(i)^T\mathbf{u}(i)} \tag{43}$$

for every input vectors.

4. Simulation Results

To show the validity of the proposed algorithm, we did computer simulations of system identification. As the unknown system we choose a FIR low pass filter of 20th order. We compared the LMS, the RLS, and the proposed algorithm in terms of IRER. We used the normalized LMS with $\alpha=1$ to determine μ for the LMS and used forgetting factor $\lambda=0.9$ for the RLS. For the proposed algorithm we changed N, the number of iteration, while P, the length of least squares, was fixed to 21, the length of the unknown system. The step size parameter ν was calculated by Eq. (42) with $\alpha=1$. We compared two cases, i.e. i) when the input signal is a white Gaussian process and ii) when the input signal is an output of AR(1) process, or a colored input, with the AR coefficient 0.9. To show the effect of noise we added $m(n)$, see Fig. 1, with varying the power P_m. Noting that the results are averages of ten independent processes.

First, we show the results of when the input signal is a white Gaussian random process with covariance $\sigma=1.0$. The results are shown in Figs. 4 and 5. Figure 4 is the results of when $P_m=-40$ dB and Fig. 6 shows that of when $P_m=-8$ dB. From Fig. 4, we see that the proposed algorithm provides the intermediate performance between the LMS and the RLS in high SNR situation as we described in Sect. 3. From Fig. 5,

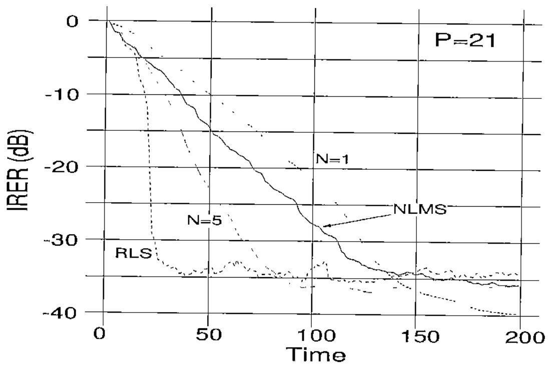


Fig. 4 Comparison of the performance of three adaptive algorithms, the LMS, the RLS, and the proposed. Power of additive noise is −40 dB.

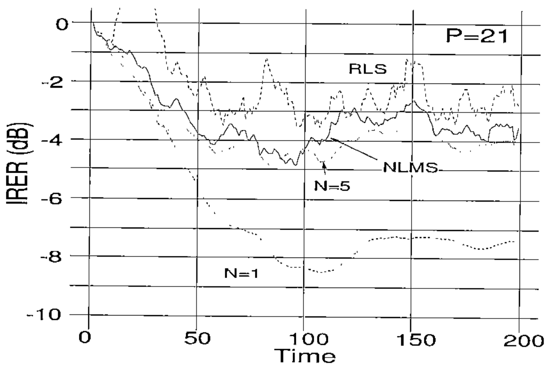


Fig. 5 Comparison of the performance of three adaptive algorithms, LMS, RLS, and the proposed. Power of additive noise is −8 dB.

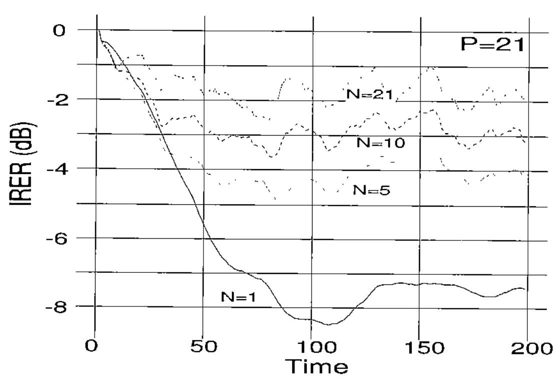


Fig. 6 Comparison of the performance of the proposed algorithms with various N, the number of iteration. Power of additive noise is −8 dB.

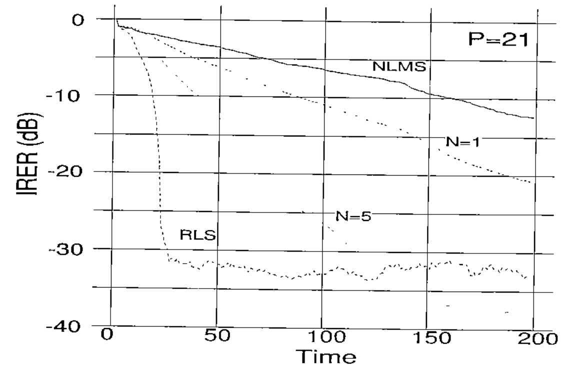


Fig. 7 Comparison of the performance of three adaptive algorithms, LMS, RLS, and the proposed. Input signal is a output of AR(1) process. Power of additive noise is −40 dB.

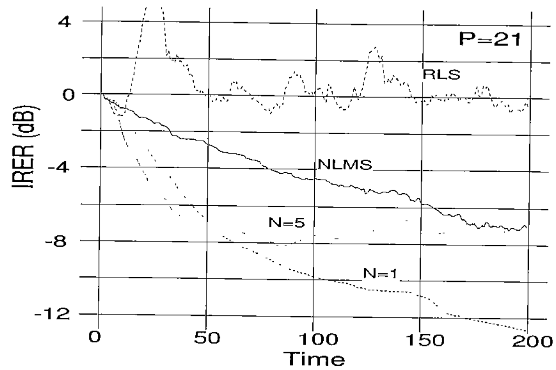


Fig. 8 Comparison of the performance of three adaptive algorithms, LMS, RLS, and the proposed. Input signal is a output of AR(1) process. Power of additive noise is −8 dB.

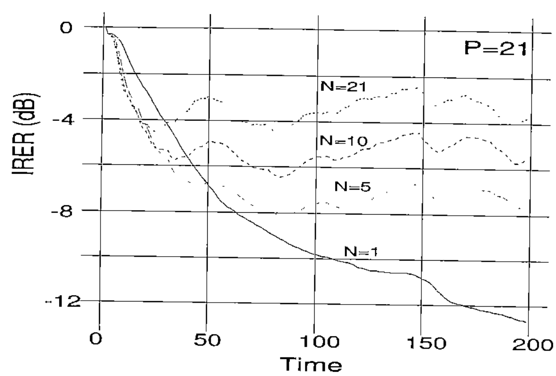


Fig. 9 Comparison of the performance of the proposed algorithms with various N, the number of iteration. Input signal is a output of AR(1) process. Power of additive noise is −8 dB.

we verify that the property described in Sect. 2.3.2 is holds in the proposed algorithm, i.e., the proposed algorithm with small N provides the best solution. To show more about the noise-iteration relationship, results of simulations with various N are shown in Fig. 6.

Next, we show the results of simulations with colored inputs. As mentioned before, the colored inputs is generated by an AR(1) filter with a white Gaussian process as an input where the AR coefficient is 0.9. The results are shown in Figs. 7, 8, and 9. From these figures, we see that the proposed algorithm's performance has the same tendency under white noise inputs. The validity of the selection of ν described in Sect. 3.5 is also shown from these figures. The proposed algorithm converges in both of the white-input case and the color-input case although we used the same formula (42) to calculate ν.

5. Conclusion

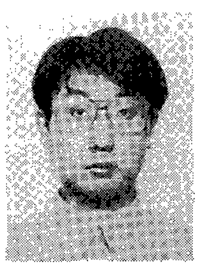
In this paper, we proposed a gradient type adaptive algorithm which is able to apply to non-stationary applications and is derived from the steepest descent algorithm based on the least squares criteria. First we reviewed the steepest descent algorithm for stationary processing to search the minimum point in the stationary plane. We considered the relation between the effect of noise and the number of iteration and showed that the steepest descent algorithm have the ability for providing better solutions than the RLS algorithm does under low signal to noise ratio conditions.

Then we proposed a gradient type algorithm which is derived from the steepest descent algorithms based on the least squares criteria. We showed that the proposed algorithm has two adjustable parameters and the characteristic of the algorithm can be controllable by adjusting them. The proposed algorithm provides a general theoretical view point of the adaptive algorithm which includes the conventional algorithms such as the LMS and the RLS. Actually, the proposed algorithm can be adjusted to provide the characteristic of the RLS or the LMS by changing the two parameters. We also presented a method for adjusting the value of the step size parameter according to the variation of a input signal which can be thought to be a generalized version of the normalized algorithm.

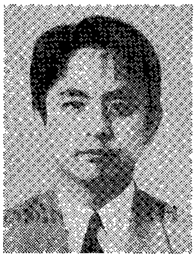
Finally, we showed the results of computer simulations. The results show that the proposed algorithm provides the reasonable solutions under possible conditions which occur in the actual application.

References

[1] Alexander, S.T., *Adaptive Signal Processing*, Springer-Verlag, N.Y., 1986.
[2] Haykin, S., *Adaptive Filter Theory*, Second Edition, Prentice-Hall, Englewood Cliffs, N.J., 1991.
[3] Widrow, B. and Stearns, S.D., *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1985.
[4] Solo, V., "The limiting behavior of LMS," *IEEE Trans. Signal Processing*, vol.SP-37, pp.1909–1922, Dec. 1989.
[5] Solo, V., "The error variance of LMS with time-varing weights," *IEEE Trans. Signal Processing*, vol.SP-40, pp.803–813, Apr. 1992.
[6] Slock, D.T.M., "On the Convergence Behavior of the LMS and the Normarized LMS Algorithm," *IEEE Trans. Signal Processing*, vol.SP-41, pp.2811–2825, Sep. 1993.
[7] Carayannis, G., Manolakis, D.G. and Kalouptsidis, N., "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction," *IEEE Trans. Acous., Speech, Signal Processing*, vol.ASSP-31, pp. 1394–1402, Dec. 1983.
[8] Cioffi J.M. and Kailath, T., "Fast, Recursive-Least-Squares Transversal Filters for Adaptive Filtering," *IEEE Trans. Acous., Speech, Signal Processing*, vol.ASSP-32, pp.304–338, Apr. 1984.
[9] Cioffi J.M. and Kailath, T., "Windowed Fast Transversal Filters Adaptive Algorithms with Normalization," *IEEE Trans. Acous., Speech, Signal Processing*, vol.ASSP-33, pp.607–625, Jun. 1985.
[10] Cioffi J.M. and Kailath, T., "An Efficient RLS Data-Driven Echo Canceller for Fast Initialization of Full-Duplex Data Transmission," *IEEE Trans. Commun.*, vol.ASSP-33, pp.601–611, Jul. 1985.
[11] Cioffi, J.M., "Limited-Precision Effects in Adaptive Filtering," *IEEE Trans. Circuits Syst.*, vol.ASSP-34, pp.821–833, Jul. 1987.
[12] Harris, R.W., Chabries, D.M. and Bishop, F.A., "A Variable Step (VS) Adaptive Filter Algorithm," *IEEE Trans. Acous., Speech, Signal Processing*, vol.ASSP-34, pp.309–316, Apr. 1986.
[13] Kwong, R.H. and Jonhston, E.W., "A Variable Step Size LMS Algorithm," *IEEE Trans. Signal Processing*, vol.SP-40, pp.1633–1642, Jul. 1992.
[14] Clark, G.A., Mitra, S.K. and Parker, S.R., "Block implementation of adaptive deigital filters," *IEEE Trans. Acous., Speech, Signal Processing*, vol.ASSP-29, pp.744–752, Jun. 1981.
[15] Petraglia M.R. and Mitra, S.K., "Adaptive FIR Filter Structure Based on the Generalized Subband Decomposition of FIR Filters," *IEEE Trans. Circuits Syst. II: Analog and Digital Siganl Processing*, vol.ASSP-40, pp.354–362, Jun. 1993.
[16] Sommen, P.T., Van Gerwen, P.J., Kotmans, H.J. and Janssen, A.J.E.M., "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. Circuits Syst.*, vol.ASSP-34,, pp.788–798, Jul. 1987.
[17] Gilloire, A.G. and Vetterli, M. "Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments, and Application to Acoustic Echo Cancellation," *IEEE Trans. Signal Processing*, vol.SP-40, pp.1862–1875, Aug. 1992.
[18] Luenberger, D.G., *Linear and Nonlinear Programming*, Second Edition, Addison Wesley, 1982.
[19] Arora, J.S., *Introduction to Optimum Design*, McGraw-Hill, 1989.
[20] Adachi, S. and Sano, A., "Recursive Identification Algorithm Based on Gradient Method with Application to Adaptive Bit Detector," *IEICE Trans.* vol.J67-A Dec. 1984.
[21] Widrow, B., Glover, Jr. J.R., McCool, J.M., Kaunitz, J., Williams, C.S., Hearn, R.H., Zeidler, J.R., Dong, Jr. E., and Goodlin, R.C., "Adaptive Noise Cancelling: Principles and Applications," *Proc. IEEE*, vol.63, pp.1692–1716, Dec. 1975.
[22] Rao, C.R., *Linear Statistical Inference and Its Applications*, 2nd ed., Wiley, NY, 1973.



Kiyoshi Nishikawa was born in Tokyo, Japan on December 12, 1966. He received the B.E. and the M.E. degrees in electrical engineering from Tokyo Metropolitan University in 1990 and 1992, respectively. Since 1993, he has been with Tokyo Metropolitan Universtiy as an Assistant Professor. His research interest includes the adaptive digital processing. He is a member of IEEE SP, CAS, Information, and Computer Society.



Hitoshi Kiya was born in Yamagata, Japan on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, and the D.E. degree in electrical engineering from Tokyo Metropolitan University in 1980, 1982, and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currency an Associate Professor of Electronics and Information Engineering. His current research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. Dr. Kiya is a member of IEEE CAS, ASSP, IT, and COM Society, the institute of image electronics engineers of Japan and the institute of television engineers of Japan.