Total Least Mean Squares Algorithm

Da-Zheng Feng, Zheng Bao, *Senior Member, IEEE*, and Li-Cheng Jiao, *Senior Member, IEEE*

***Abstract*— Widrow proposed the least mean squares (LMS) algorithm, which has been extensively applied in adaptive signal processing and adaptive control. The LMS algorithm is based on the minimum mean squares error. On the basis of the total least mean squares error or the minimum Raleigh quotient, we propose the total least mean squares (TLMS) algorithm. The paper gives the statistical analysis for this algorithm, studies the global asymptotic convergence of this algorithm by an equivalent energy function, and evaluates the performances of this algorithm via computer simulations.**

***Index Terms*—Hebb learn rule, LMS algorithm, stability, statistical analysis, system identification, unsupervised learning.**

I. Introduction

BASED on the minimum mean squared error, Widrow proposed the well-known least mean squares (LMS) algorithm [1], [2], which has been successfully applied in adaptive interference canceling, adaptive beamforming, and adaptive control. The LMS algorithm is a random adaptive algorithm that fits in with nonstationary signal processing. The performances of the LMS algorithm have been extensively studied. If interference only exists in the output of the analyzed system, the LMS algorithm can only obtain the optimal solutions of signal processing problems. However, if there is interference in both input and output of the analyzed system, the LMS algorithm can only obtain the suboptimal solutions of signal processing problems. In order to modify the LMS algorithm, a new adaptive algorithm should be proposed on the basis of the total minimum mean squared error. This paper proposes a total least-mean-squares (TLMS) algorithm, which is also a random adaptive algorithm, and intrinsically solves the total least-squares (TLS) problems.

Although the total least-squares problems were proposed in 1901 [3], their basic performances had not been studied by Golub and Van Loan until 1980 [4]. The solutions of TLS problems were extensively applied in the fields of economics, signal processing, and so on [5]–[9]. The solution of a TLS problem can be obtained by the singular value decomposition (SVD) of matrices [4]. Since the multiplication number of the SVD for the $N \times N$ matrix is $6N^3$, the applications of TLS problems are limited in practice. To solve TLS problems in signal processing, we propose a TLMS algorithm that only requires about $4N$ multiplication per iteration. We give its statistical analysis, study its dynamic properties, and evaluate its behaviors via the computer simulations.

Recently, much attention has been paid to the unsupervised learning algorithm, in which the feature extraction is performed in a purely data-driven fashion without any index or category information for each data sample. The well-known approaches include Grossberg's adaptive resonance theory [12], Kohonen's self-organizing feature maps [13], and Fukushima's neocognitron networks [14]. Another unsupervised learning approach uses the principal component analysis [10]. It is shown that if the weight of a simple linear neuron is updated with an unsupervised constrained Hebbian learning rule, the neuron tends to extract the principal component from a stationary input vector sequence [10]. This is an important step in using the theory of neural networks to solve the problem of stochastic signal processing. In recent years, a number of new developments have taken place in this direction. For example, several algorithms for finding multiple eigenvectors of the correlation matrix have been proposed [15], [21]. For a good survey, see the book by Bose and Liang [22].

More recently, a new modified Hebbian learning procedure has been proposed for a linear neuron so that the neuron extracts the minor component of the input data sequence [11], [23], [24]. The value of the weight vector of the neuron has been shown to converge to a vector in the direction of the eigenvector associated with the smallest eigenvalue of correlation matrix of the input data sequence. This algorithm has been applied to fit curve, surface, or hypersurface optimally in the TLS sense [24]. This algorithm, for the first time, provided a neural-based adaptive scheme for the TLS estimation problem. In addition, Gao *et al.* proposed the constrained anti-Hebbian algorithm that has very simple structure, requires little computing volume at each iteration, and can be also used to solve total adaptive signal processing [30], [31]. However, as the autocorrelation matrix is positively definite, its weights will converge to zero or to infinity [32].

The TLMS algorithm also comes from Oja and Xu's learning algorithm for extracting the minor component of a multidimensional data sequence [11], [23], [24]. Note that the input number of the TLMS algorithm is more than the input number of the learning algorithm for extracting the minor component of a stochastic vector sequence. In adaptive signal processing, the inputs of the TLMS algorithm are divided into two groups corresponding to different weighting vectors dependent of the signal-noise ratio (SNR) of the input and the output, where one group consists of inputs of the analyzed system and another consists of outputs of the analyzed system, whereas inputs of Oja and Xu's learning algorithm represent a random data vector. If there is interference in both the input and the output

Manuscript received June 28, 1995; revised October 28, 1997. The associate editor coordinating the review of this paper and approving it for publication was Dr. Akihiko Sugiyama.

The authors are with the Research Institute of Electronic Engineering, Xidian University, Xi'an, China (e-mail: zhbao@rsp.xidian.edu.cn).

Publisher Item Identifier S 1053-587X(98)05220-9.

1053–587X/98$10.00 © 1998 IEEE

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

of the analyzed system, the behavior of the TLMS algorithm is superior to the LMS algorithm.

II. Total Least Squares Problem

The total least-squares approach is an optimal technique that considers both stimulation error and response error. Here, the implication of TLS problems is illustrated by the solution of a conflict linear equation

$$\mathbf{A}\mathbf{x} \approx \mathbf{b} \tag{1}$$

where $\mathbf{A} \in R^{m \times n}$, $\mathbf{x} \in R^n$, $\mathbf{b} \in R^m$, $m > n$. A conventional method for solving the problem is the least squares (LS) method. In the solution of a problem by LS, there are a data matrix and an observation vector. When there are more equations than unknowns, e.g., $m > n$, the set is overdetermined. Unless $\mathbf{b}$ belongs to $R(\mathbf{A})$ (the ranges of $\mathbf{A}$), the overdetermined set has no exact solution and is therefore denoted by $\mathbf{A}\mathbf{x} \approx \mathbf{b}$. The unique minimum norm Moore–Penrose solution to the LS problem is then given by

$$\|(\mathbf{b} - \mathbf{A}\mathbf{x})\|_2 = \min \tag{2}$$

where $\|\cdot\|_2$ indicates the Euclidean length of the vector. The solution to (2) is equivalent to solving

$$\mathbf{A}^T\mathbf{A}\mathbf{x}_{\mathrm{LS}} = \mathbf{A}^T\mathbf{b}$$

or

$$\mathbf{x}_{\mathrm{LS}} = \mathbf{A}^{+}\mathbf{b} \tag{3}$$

where "+" denotes the Moore–Penrose pseudoinverse of a matrix. The assumption in (3) is that the errors are confined only to the "observation" vector $\mathbf{b}$.

We can reformulate the ordinary LS problem as follows: Determine $\mathbf{x}_{\mathrm{LS}}$, which satisfies

$$\mathbf{A}\mathbf{x}_{\mathrm{LS}} = \mathbf{b} + \mathbf{r} \tag{4}$$

and for which

$$\min \|\mathbf{r}\|_2, \text{ subject to } \mathbf{b} + \mathbf{r} = \text{ Range } (\mathbf{A}). \tag{5}$$

The underlying assumption in the solution of the ordinary LS problem is that errors only occur in the observation vector $\mathbf{b}$, and the data matrix $\mathbf{A}$ is exactly known. Often, this assumption is not realistic because of sampling, modeling, or measurement error affecting the matrix. One way to take errors in the matrix $\mathbf{A}$ into account is to introduce perturbation in $\mathbf{A}$ and solve the following problem as outlined in the below.

In the TLS problem, there are perturbations of both the observation vector $\mathbf{b}$ and the data matrix $\mathbf{A}$. We can consider the TLS problem to be the problem of determining the $\mathbf{x}_{\mathrm{TLS}}$, which satisfies

$$(\mathbf{A} + \mathbf{E})\mathbf{x}_{\mathrm{TLS}} = \mathbf{b} + \mathbf{r} \tag{6}$$

where $\mathbf{E}$ and $\mathbf{r}$ are perturbations of $\mathbf{A}$ and $\mathbf{b}$, respectively, and for which

$$\min \|[\mathbf{E}|\mathbf{r}]\|_F, \text{ subject to } \mathbf{b} + \mathbf{r} \in \text{ Range } (\mathbf{A} + \mathbf{E}) \tag{7}$$

where $[\mathbf{E}|\mathbf{r}]$ represents the matrix $\mathbf{E}$ augmented by the vector $\mathbf{r}$, and $\|.\|_F$ denotes the Frobenius norm viz. $\|\mathbf{B}\|_F^2 = \sum_i \sum_j |b_{ij}|^2$. Once a minimum solution is found, then any $\mathbf{x}_{\mathrm{TLS}}$ satisfying

$$(\mathbf{A} + \mathbf{E})\mathbf{x}_{\mathrm{TLS}} = \mathbf{b} + \mathbf{r}$$

is said to be the solution of the TLS problem (7). Thus, the problem is equivalent to the problem for solving a nearest compatible LS problem

$$\min \|(\mathbf{A} + \mathbf{E})\mathbf{x}_{\mathrm{TLS}} - (\mathbf{b} + \mathbf{r})\|_2 \tag{8}$$

where "nearest" is measured by the weighted Frobenius norm above.

In the TLS problem, unlike the LS problems, the vector $\mathbf{b}$ or its estimate does not lie in the range space of matrix $\mathbf{A}$.

Consider matrix $\mathbf{C} \in R^{m \times (n+1)}$

$$\mathbf{C} = [\mathbf{A} \,|\, \mathbf{b}]. \tag{9}$$

The singular value decomposition (SVD) of matrix $\mathbf{C}$ can be written as [4]

$$\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

or

$$\mathbf{D} = \mathbf{U}^T\mathbf{C}\mathbf{V} = \mathrm{diag}(\sigma_1, \cdots, \sigma_p), \quad p = \min\{m, n+1\} \tag{10}$$

where the superscript T denotes transposition, $\mathbf{U}$ is $m \times p$ and unitary, $\mathbf{V}$ is $(n+1) \times p$ and unitary, and $\mathbf{U}$ and $\mathbf{V}$, respectively, contain the first p left singular vectors and the first p right singular vectors of $\mathbf{C}$. $\mathbf{U}$ and $\mathbf{V}$ can be expressed as

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_p] \in R^{m \times p}, \\ \mathbf{V} &= [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p] \in R^{(n+1) \times p}. \end{aligned} \tag{11}$$

Let σ_i, $\mathbf{u}_i$, $\mathbf{v}_i$ be the ith singular value, left singular vector, and right singular vector of $\mathbf{C}$, respectively. They are related by

$$\mathbf{C}\mathbf{v}_i = \sigma_i\mathbf{u}_i, \qquad \mathbf{C}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i. \tag{12}$$

The $\mathbf{v}_{n+1}$ is the right singular vector corresponding to the smallest singular value of $\mathbf{C}$, and then, the vector$[\mathbf{x}_{\mathrm{TLS}}^T, -1]^T$ is parallel to the right singular vector [4]. The TLS solution $\mathbf{x}_{\mathrm{TLS}}$ is obtained from

$$\begin{pmatrix} \mathbf{x}_{\mathrm{TLS}} \\ -1 \end{pmatrix} = -\frac{\mathbf{v}_{n+1}}{v_{n+1,\, n+1}} \tag{13}$$

where $v_{n+1,\, n+1}$ is the last component of $\mathbf{v}_{n+1}$.

The vector $\mathbf{v}_{n+1}$ is equivalent to the eigenvector corresponding to the smallest eigenvalue of the correlation matrix $\mathbf{R} = (\mathbf{C}^T\mathbf{C})/m$. Thus, the TLS solution can also be achieved via the eigenvalue decomposition of the correlation matrix. The correlation matrix $\mathbf{R}$ is normally estimated from the data samples in many applications, whereas the SVD operates on the data samples directly. In practice, the SVD technique is mainly used to solve the TLS problems since it offers some advantages over the eigenvalue decomposition technique in

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

terms of tolerance to quantization and lower sensitivity to computational errors [25]. However, adaptive algorithms have also been used to estimate the eigenvector corresponding to the smallest eigenvalue of the data covariance matrix [27], [28].

III. Derivation of the Total Least Mean Squares Algorithm

We consider a problem of adaptive signal processing. Let

$\mathbf{x}(k)$ n-dimensional input sequence of the system;
$d(k)$ output sequence of the system;
k time sequence.

Both the input and output signal samples are corrupted by additive white noise, quantization, and computation error and man-made interference called interference. Let $\Delta\mathbf{x}(k)$ be the interference of input-vector sequence $\mathbf{x}(k)$ and $\Delta d(k)$ the interference of output sequence $d(k)$.

Define an augmented data vector sequence as

$$\mathbf{z}(k) = [\mathbf{x}^T(k)|d(k)]^T \tag{14}$$

where "T" denotes transposition. Let the augmented interference vector sequence be

$$\Delta\mathbf{z}(k) = [\Delta\mathbf{x}^T(k)|\Delta d(k)]^T. \tag{15}$$

Then, the augmented "observation" vector can be represented as

$$\tilde{\mathbf{z}}(k) = \mathbf{z}(k) + \Delta\mathbf{z}(k) = [\tilde{\mathbf{x}}^T(k)|\tilde{d}(k)]^T \tag{16}$$

where

$$\tilde{\mathbf{x}}(k) = \mathbf{x}(k) + \Delta\mathbf{x}(k), \qquad \tilde{d}(k) = d(k) + \Delta d(k).$$

Define an augmented weight vector sequence as

$$\tilde{\mathbf{w}}(k) = [\mathbf{w}^T(k)|w_{n+1}]^T \tag{17}$$

where vector $\mathbf{w}(k)$ can be expressed as

$$\mathbf{w}(k) = [w_1, \cdots, w_n]^T. \tag{18}$$

In the LMS algorithm [2], the estimation of the output is represented as a linear combination of the input samples, i.e.,

$$y(k) = \tilde{\mathbf{x}}^T(k)\mathbf{w}(k). \tag{19}$$

The output error signal with time index k is

$$\varepsilon(k) = \tilde{d}(k) - y(k). \tag{20}$$

Substituting (19) into this expression yields

$$\varepsilon(k) = \tilde{d}(k) - \tilde{\mathbf{x}}^T(k)\mathbf{w}(k). \tag{21}$$

The LS solutions about the above problem can be obtained by solving the optimization problem

$$\min E\{\varepsilon^2(k)\}. \tag{22}$$

Here, we drop the time index k from the weight vector $\mathbf{w}(k)$ for convenience and expand $\varepsilon^2(k)$ to obtain the instantaneous error

$$\varepsilon^2(k) = \tilde{d}^2(k) - 2\tilde{d}(k)\tilde{\mathbf{x}}^T(k)\mathbf{w} + \mathbf{w}^T\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}^T(k)\mathbf{w}. \tag{23}$$

We assume that these are statistically stationary and take the expected value of (23)

$$\begin{aligned} E\{\varepsilon^2(k)\} = & E\{\tilde{d}^2(k)\} - 2E\{d(k)\tilde{\mathbf{x}}^T(k)\}\mathbf{w} \\ & + \mathbf{w}^T E\{\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}^T(k)\}\mathbf{w}. \end{aligned} \tag{24}$$

Let $\mathbf{R}$ be similarly defined as the autocorrelation matrix

$$\mathbf{R} = E\{\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}^T(k)\}. \tag{25}$$

Let $\mathbf{P}$ be similarly defined as the column vector

$$\mathbf{P} = E\{\tilde{d}(k)\tilde{\mathbf{x}}(k)\}. \tag{26}$$

Thus, $E\{\varepsilon^2(k)\}$ is re-expressed as

$$E\{\varepsilon^2(k)\} = E\{\tilde{d}^2(k)\} - 2\mathbf{P}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}\mathbf{w}. \tag{27}$$

The gradient can be obtained as

$$\nabla E\{\varepsilon^2(k)\} = -2\mathbf{P} + 2\mathbf{R}\mathbf{w}. \tag{28}$$

A simple gradient search algorithm for optimization problem is

$$\mathbf{w}(l+1) = \mathbf{w}(l) - \mu(\mathbf{R}\mathbf{w}(l) - \mathbf{P}) \tag{29}$$

where l is the iteration number, and μ is called the step length or learning rate. Thus, $\mathbf{w}(l)$ is the "present" adjustment value, whereas $\mathbf{w}(l+1)$ is the "new" value. The gradient at $\mathbf{w} = \mathbf{w}(l)$ is designated by $\nabla = 2(\mathbf{R}\mathbf{w}(l) - \mathbf{P})$. The parameter μ is a positive constant that governs stability and rate of convergence and is smaller than $1/2\lambda_{\max}$ ($\lambda_{\max}$ is the largest eigenvalue of the correlation matrix $\mathbf{R}$). To develop an adaptive algorithm using the gradient search algorithm, we would estimate the gradient of $E\{\varepsilon^2(k)\}$ by taking differences between short-term averages of $\varepsilon^2(k)$. In the LMS algorithm [2], Widrow has taken the squared-error itself as an estimation of $E\{\varepsilon^2(k)\}$. Then, at each iteration in the adaptive process, we have a gradient estimate of the form

$$\tilde{\nabla} = -2\tilde{d}(k)\tilde{\mathbf{x}}(k) + 2y(k)\tilde{\mathbf{x}}(k) = 2(y(k) - \tilde{d}(k))\tilde{\mathbf{x}}(k). \tag{30}$$

With this simple estimate of gradient, we can specify a steepest descent type of adaptive algorithm. From (29) and (30), we have

$$\begin{aligned} y(l) &= \mathbf{w}^T(l)\tilde{\mathbf{x}}(l) \\ \mathbf{w}(l+1) &= \mathbf{w}(l) - \mu(y(l) - \tilde{d}(l))\tilde{\mathbf{x}}(l). \end{aligned} \tag{31}$$

This is the LMS algorithm [2]. As before, μ is the gain constant that regulates the speed and stability of adaptation. Since the weight changes at each iteration are based on imperfect gradient estimates, we would expect the adaptive process to be noisy. Thus, the LMS algorithm only obtains an approximate LS solution for the above adaptive signal-processing problem.

In the TLMS algorithm below, the estimate of the desired output $d(k)$ is expressed as a linear combination of the desire input sequence $\mathbf{x}(k)$, i.e.,

$$\hat{d}(k) = \mathbf{x}^T(k)\mathbf{w}(k). \tag{32}$$

The TLS solution of the above signal processing problem can be obtained by solving

$$\min E\{||[\Delta\mathbf{x}^T(k)|\Delta d(k)]||_2\} = \min E\{||\Delta\mathbf{z}(k)||_2\}. \tag{33}$$

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

The above optimization problem is equivalent to the problem for solving nearest compatible LS problem

$$\min E\left\{\left|\tilde{\mathbf{x}}^T(k)\mathbf{z}_{\text{TLS}} - \tilde{d}(k)\right|^2\right\}. \tag{34}$$

Furthermore, the optimization problem (33) is equivalent to the optimization problem

$$\begin{aligned} &\min E\{\left|\tilde{\mathbf{z}}^T(k)\tilde{\mathbf{w}}(k)\right|^2\} \quad \|\tilde{\mathbf{w}}\|_2 = \alpha \\ &\qquad\qquad \begin{pmatrix} \mathbf{z}_{\text{TLS}} \\ -1 \end{pmatrix} = -\frac{\tilde{\mathbf{w}}}{w_{n+1}} \end{aligned} \tag{35}$$

where α can be any positive constant. Expanding (35), we get

$$\min \tilde{\mathbf{w}}^T\tilde{\mathbf{R}}\tilde{\mathbf{w}}, \qquad \|\tilde{\mathbf{w}}\|_2 = \alpha \tag{36}$$

where

$$\tilde{\mathbf{R}} = E\{\tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^T(k)\} \tag{37}$$

represents the autocorrelation matrix of the augmented data-vector sequence and is simply called the augmented correlation matrix. It is easily shown that the solution vector of the optimization problem (36) is the eigenvector associated with the smallest eigenvalue of the augmented autocorrelation matrix.

An iterative search procedure for this eigenvector of $\tilde{\mathbf{R}}$ can be represented algebraically as

$$\tilde{\mathbf{w}}(l+1) = \tilde{\mathbf{w}}(l) + \mu[\tilde{\mathbf{w}}(l) - \|\tilde{\mathbf{w}}(l)\|_2^2\tilde{\mathbf{R}}\tilde{\mathbf{w}}(l)] \tag{38}$$

where l is the step or iteration number, and μ is a positive constant that governs stability and rate of convergence; its choice is discussed later. The stability and convergence of the above iteration search algorithm will also be discussed later. When $\tilde{\mathbf{R}}$ is a positive definite matrix, the term $(-\|\tilde{\mathbf{w}}(l)\|_2^2\tilde{\mathbf{R}}\tilde{\mathbf{w}}(l))$ in (38) is a higher order decay term. Thus, $\|\tilde{\mathbf{w}}(l)\|_2$ is bounded.

To develop an adaptive algorithm, we would estimate the augmented correlation matrix by computing

$$\tilde{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K}\tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^T(k) \tag{39}$$

where K is a large-enough positive integer number. Instead, to develop the TLMS algorithm, we take $\tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^T(k)$ itself as an estimate of $\tilde{\mathbf{R}}$. Then, at each iteration in the adaptive process, we have an estimate of the augmented correlation matrix

$$\tilde{\mathbf{R}} \approx \tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^T(k). \tag{40}$$

From (38) and (40), we have

$$\begin{aligned} \tilde{y}(l) &= \tilde{\mathbf{z}}^T(l)\tilde{\mathbf{w}}(l) \\ \tilde{\mathbf{w}}(l+1) &= \tilde{\mathbf{w}}(l) + \mu[\tilde{\mathbf{w}}(l) - \|\tilde{\mathbf{w}}(l)\|_2^2\tilde{y}(l)\tilde{\mathbf{z}}(l)]. \end{aligned} \tag{41}$$

This is the TLMS algorithm. As before, μ is the gain constant that regulates the speed and stability of adaptation. Since the solution changes at each iteration are based on imperfect estimates of the augmented correlation matrix, we would expect the adaptive process to be noisy. From its form in (41), we can see that the TLMS algorithm can be implemented in a practical system without averaging or differentiation and is also elegant in its simplicity and efficiency.

To develop the above TLMS algorithm, we adopt the method similar to that used in the LMS algorithm. When the TLMS algorithm is formulated in the framework of an adaptive FIR filtering, its structure, computational complexity, and numerical performance are very similar to those of the well-known LMS algorithm [2]. Note that the LMS algorithm requires $2n$ multiplication, whereas the TLMS algorithm needs about $4n$ multiplication.

In neural network theory, the term $(-\tilde{y}(l)\tilde{\mathbf{z}}(l))$ in (41) is generally called the anti-Hebb learning rule. The term $(-\|\tilde{\mathbf{w}}(l)\|_2^2\tilde{y}(l)\tilde{\mathbf{z}}(l))$ in (41) is a higher order decay term. In the section below, we shall prove that the algorithm is globally asymptotically convergent in the averaging sense. Once a stable $\tilde{\mathbf{w}}$ is found, the TLS solution of the above adaptive signal processing problem is

$$\mathbf{z}_{\text{TLS}} = -\frac{\mathbf{w}}{w_{n+1}}. \tag{42}$$

Discussion: Since any eigenvector of the augmented correlation matrix is not unique, any random algorithm for solving (34) is also not unique. For example, the algorithm

$$\begin{aligned} \tilde{y}(l) &= \tilde{\mathbf{z}}^T(l)\tilde{\mathbf{w}}(l) \\ \tilde{\mathbf{w}}(l+1) &= \tilde{\mathbf{w}}(l) + \mu(\tilde{\mathbf{w}}(l) - \|\tilde{\mathbf{w}}(l)\|_2\tilde{y}(l)\tilde{\mathbf{z}}(l)) \end{aligned}$$

and other algorithms [11], [23] can also be turned into the TLMS algorithm, but we have not proved that those algorithms in [11] and [23], as well as the above algorithm, are globally asymptotically stable.

IV. Statistical Analysis and Stability

Following the reasoning of Oja [10], Xu *et al.* [25], and others [15], [26], [27], if the distribution of $\tilde{\mathbf{z}}(k)$ satisfies some realistic assumptions and the gain coefficient decreases in a suitable way, as given in the stochastic approximation literature, (41) can be approximated by a differential equation

$$\frac{d\tilde{\mathbf{w}}(t)}{dt} = \tilde{\mathbf{w}}(t) - \|\tilde{\mathbf{w}}(t)\|_2^2\tilde{\mathbf{R}}\tilde{\mathbf{w}}(t) \tag{43}$$

where t denotes time. We shall illustrate the process of derivation of the above formula. For the sake of simplicity, we make the following two statistical assumptions:

Assumption 1: The augmented data vector sequence $\tilde{\mathbf{z}}(k)$ is not correlated with the weight vector sequence $\tilde{\mathbf{w}}(k)$.

Discussion: When the changes of the signal are much faster than those of the weight, Assumption 1 can be approximately satisfied. Assumption 1 implies that the learning rate must be very small, which means that the weight only varies a little bit at each iteration.

Assumption 2: Signal $\tilde{\mathbf{z}}(k)$ is the bounded continuous-valued stationary ergodic data stream with finite second-order moment.

According to Assumption 2, the augmented correlation matrix $\tilde{\mathbf{R}}$ can be expressed as

$$\tilde{\mathbf{R}} = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^T(k). \tag{44}$$

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

In order to obtain a realistic model, we shall use the following two approximate conditions: There exists a positive integer K large enough, and μ is a learning rate small enough that makes

$$\tilde{\mathbf{R}} \approx \frac{1}{K} \sum_{i=1}^{K} \tilde{\mathbf{z}}(k_{ti})\tilde{\mathbf{z}}^T(k_{ti}) \tag{45}$$

for any k and

$$\tilde{\mathbf{w}}(k) \approx \frac{1}{K} \sum_{i=0}^{K-1} \tilde{\mathbf{w}}(k+i) \tag{46}$$

for any k.

The implication of the above approximation conditions is that $\tilde{\mathbf{z}}(k)$ varies much faster than $\tilde{\mathbf{w}}(k)$. For a stationary signal, we have

$$\begin{aligned}\tilde{\mathbf{w}}(l+K) - \tilde{\mathbf{w}}(l) &= \mu \sum_{i=0}^{K-1} \tilde{\mathbf{w}}(l+i) - \mu \sum_{i=0}^{K-1} \|\tilde{\mathbf{w}}(l+i)\|_2^2 \\ &\quad \cdot \tilde{\mathbf{z}}(l+i)\tilde{\mathbf{z}}^T(l+i)\tilde{\mathbf{w}}(l+i) \\ &\approx K\mu\,\tilde{\mathbf{w}}(l) - K\mu\|\tilde{\mathbf{w}}(l)\|_2^2 \frac{1}{K} \sum_{i=0}^{K-1} \\ &\quad \cdot \tilde{\mathbf{z}}(l+i)\tilde{\mathbf{z}}^T(l+i)\mathbf{w}(l) \\ &\approx K\mu(\tilde{\mathbf{w}}(l) - \|\tilde{\mathbf{w}}(l)\|_2^2 \tilde{\mathbf{R}}\tilde{\mathbf{w}}(l)). \end{aligned} \tag{47}$$

It is worth mentioning that in this key step, a random system (41) is approximately represented by a deterministic system (47).

In order to simplify mathematical expression, we shall replace time index $l+K$ with $l+1$ and learning rate or gain constant $K\mu$ with μ again; then, (47) is changed into

$$\tilde{\mathbf{w}}(l+1) = \tilde{\mathbf{w}}(l) + \mu[\tilde{\mathbf{w}}(l) - \|\tilde{\mathbf{w}}(l)\|_2^2 \tilde{\mathbf{R}}\tilde{\mathbf{w}}(l)]. \tag{48}$$

Now, $\tilde{\mathbf{w}}(l)$ should be viewed as the mean weight vector. It is easily shown that the original differential equation of (48) is (43). We shall study the convergence of the TLMS algorithm below by analyzing the stability of (43).

Since (43) is an autonomous deterministic system, Lasalle's invariance principle [29] and Liapunov's first method can be used to study its global asymptotic stability. Let $\tilde{\mathbf{w}}^*$ represent an equilibrium point of (43). Let $\mathbf{e}$ represent the right singular vector associated with the smallest singular value $\lambda_{\min}$ of $\tilde{\mathbf{R}}$. Our objective is to make

$$\lim_{t\to\infty} \tilde{\mathbf{w}}(t) = \pm \frac{1}{\sqrt{\lambda_{\min}}}\,\mathbf{e}. \tag{49}$$

Since $\tilde{\mathbf{R}}$ is a symmetric positive definite matrix, then there must be a unitary orthogonal matrix $\mathbf{V}$ such that

$$\tilde{\mathbf{R}} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \quad \mathbf{D} = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{n+1}) \tag{50}$$

where λ_i indicates the ith singular value of $\tilde{\mathbf{R}}$, and

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n+1}]$$

$\mathbf{v}_i$ is the ith eigenvector of $\tilde{\mathbf{R}}$.

The global asymptotic convergence of the ordinary differential equation (43) can be established by the following theorem. Before giving and proving the theorem, we shall give a corollary. From Lasalle's invariance principle [29], we easily introduce the following result on global asymptotic stability.

Definition [29]: Let G be any set in R^{n+1}. We say that $E(\tilde{\mathbf{w}})$ is a Liapunov function of an $(n+1)$-dimensional dynamic system on G if i) $E(\tilde{\mathbf{w}})$ is continuous and if ii) the inner product $(\nabla E(\tilde{\mathbf{w}}), d\tilde{\mathbf{w}}/dt) \le 0$ for all $\tilde{\mathbf{w}} \in G$.

Note that the Liapunov function in Lasalle's invariance principle need not be positive definite or positive, and a positive or positive definite function is certainly not the Liapunov function.

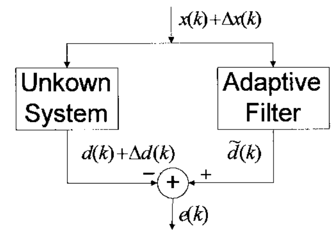

Fig. 1. Unknown system $h(k)(k=0, 1, \cdots, N-1)$ identified by filter.

TABLE I
$\mathbf{h}_p$ AND $\mathbf{h}_{pp}$ ARE, RESPECTIVELY, IMPULSE RESPONSE ESTIMATED BY TLMS ALGORITHM AND BY LMS ALGORITHM. μ_1 AND μ_2 ARE, RESPECTIVELY, THE LEARNING RATE OF TLMS AND LMS ALGORITHMS

Theoretical impulse response	$\mathbf{h}$=[-0.3, -0.9, 0.8, -0.7,0.6]
Signal-Noise Rate (dB)	Estimate of impulse response
SNR=80.0	$\mathbf{h}_p$ = [-0.3000,-0.9000,0.8000,-0.7000,0.6000]
$\mu_1 = 0.024$, $\mu_2 = 0.028$	$\mathbf{h}_{pp}$ = [-0.3000,-0.9000,0.8000,-0.7000,0.6000]
SNR=40.0	$\mathbf{h}_p$ = [-0.2999,-0.8998,0.7999,-0.6995,0.5990]
$\mu_1 = 0.015$, $\mu_2 = 0.02$	$\mathbf{h}_{pp}$ = [-0.2997,-0.9004,0.8003,-0.6995,0.5989]
SNR=20.0	$\mathbf{h}_p$ = [-0.3048,-0.9024,0.8043,-0.6991,0.5996]
$\mu_1 = 0.005$, $\mu_2 = 0.01$	$\mathbf{h}_{pp}$ = [-0.2975,-0.8922,0.7930,-0.6932,0.5913]
SNR=10.0	$\mathbf{h}_p$ = [-0.2947,-0.8975,0.8024,-0.6985,0.5943]
$\mu_1 = 0.006$, $\mu_2 = 0.018$	$\mathbf{h}_{pp}$ = [-0.2653,-0.8190,0.7206,-0.6436,0.5194]
SNR=5.0	$\mathbf{h}_p$ = [-0.3086,-0.9110,0.8163,-0.7031,-0.5942]
$\mu_1 = 0.004$, $\mu_2 = 0.016$	$\mathbf{h}_{pp}$ = [-0.2421,-0.7035,0.5916,-0.5519,0.4329]
SNR=2.0	$\mathbf{h}_p$ = [-0.3115,-0.9130,0.8223,-0.7058,0.5916]
$\mu_1 = 0.004$, $\mu_2 = 0.016$	$\mathbf{h}_{pp}$ = [-0.2046,-0.5797,0.4525,-0.4592,0.3369]
SNR=0.0	$\mathbf{h}_p$ = [-0.3155,-0.9133,0.8271,-0.7080,0.5893]
$\mu_1 = 0.004$, $\mu_2 = 0.016$	$\mathbf{h}_{pp}$ = [-0.1751,-0.4850,0.3443,-0.3856,0.6500]
SNR=-2.0	$\mathbf{h}_p$ = [-0.3222,-0.9117,0.8330,-0.7102,0.5863]
$\mu_1 = 0.004$, $\mu_2 = 0.012$	$\mathbf{h}_{pp}$ = [-0.1493,-0.3885,0.2381,-0.3129,0.1948]

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

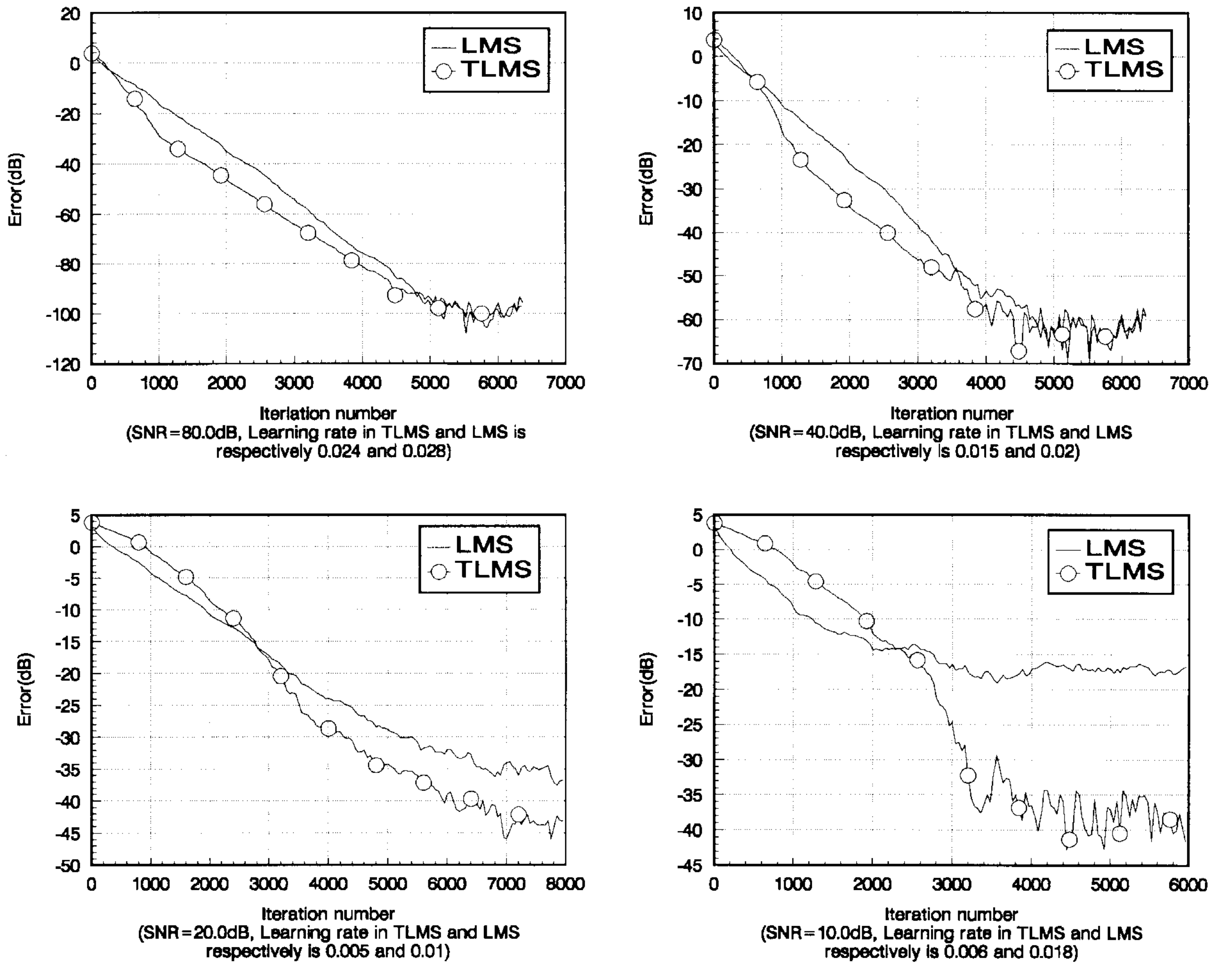

Fig. 2. Curves with large and small stable value are obtained by the LMS and the TLMS algorithm, respectively, where vertical and horizontal coordinates represent identification error and iteration number, respectively. Note that the oscillation in the learning curves originates from the noise in the learning process, the statistical rise and fall of the pseudo-random number, and the sensitivity of the estimate of the smallest singular value to the fluctuation of the psuedo-random sequence.

Corollary: If

1) $E(\tilde{\mathbf{w}})$ is a Liapunov function of (43);
2) $G_c = \{\tilde{\mathbf{w}};\ E(\tilde{\mathbf{w}}) < c\}$ is bounded for each c;
3) $E(\tilde{\mathbf{w}})$ is constant on $M \subset \{\tilde{\mathbf{w}};\ d\tilde{\mathbf{w}}/dt = 0\}$;

then M is globally asymptotically stable, where M is the stable equilibrium point set or invariance set of (43).

Theorem I: In (43), let $\tilde{\mathbf{R}}$ be a positive definite matrix with smallest eigenvalue of multiplicity one; then, $\tilde{\mathbf{w}}(t)$ globally asymptotically converge to the stable equilibrium point given by (49).

Proof: First, we prove that $\tilde{\mathbf{w}}(t)$ globally asymptotically converges to the equilibrium point of (43) as $t \to \infty$. Then, we prove that the two equilibrium points

$$\tilde{\mathbf{w}}^* = \pm\frac{1}{\sqrt{\lambda_{\min}}}\mathbf{e} \tag{51}$$

are only the two fixed points, whereas the other equilibrium points are saddle points.

We can find the following Liapunov function of (43)

$$E(t) = \tfrac{1}{2}(-\ln\|\tilde{\mathbf{w}}(t)\|_2^2 + \tilde{\mathbf{w}}^T(t)\tilde{\mathbf{R}}\tilde{\mathbf{w}}(t)). \tag{52}$$

Since $\|\tilde{\mathbf{w}}(t)\| \to 0$ or $\to \infty$, $E(t) \to \infty$, it is shown that $G_c = \{\tilde{\mathbf{w}};\ E(\tilde{\mathbf{w}}) < c\}$ is bounded for each c. Differentiating $E(t)$ along the solution of (43), we have

$$\begin{aligned}\frac{dE(t)}{dt} &= -\frac{1}{\|\tilde{\mathbf{w}}(t)\|_2^2}\frac{d\tilde{\mathbf{w}}^T(t)}{dt}\,\tilde{\mathbf{w}}(t) + \frac{d\tilde{\mathbf{w}}^T(t)}{dt}\,\tilde{\mathbf{R}}\tilde{\mathbf{w}}(t)\\ &= -\frac{1}{\|\tilde{\mathbf{w}}(t)\|_2^2}\frac{d\tilde{\mathbf{w}}^T(t)}{dt}\left[\tilde{\mathbf{w}}(t) - \|\tilde{\mathbf{w}}(t)\|_2^2\tilde{\mathbf{R}}\tilde{\mathbf{w}}(t)\right]\\ &= -\frac{1}{\|\tilde{\mathbf{w}}(t)\|_2^2}\left\|\frac{d\tilde{\mathbf{w}}(t)}{dt}\right\|_2^2.\end{aligned} \tag{53}$$

In the above formula, if $d\tilde{\mathbf{w}}(t)/dt \neq 0$, then $dE(t)/dt < 0$; iff $d\tilde{\mathbf{w}}(t)/dt = 0$, then $dE(t)/dt = 0$. Therefore, $E(t)$ globally asymptotically tends to an extreme value that corresponds to a critical point of differential equation (43). This shows that $\tilde{\mathbf{w}}(t)$ in (43) globally asymptotically converges to equilibrium points.

Let $\tilde{\mathbf{w}}(t)$ at an equilibrium point of (43) be $\tilde{\mathbf{w}}^*$; then, from (43), we have

$$\tilde{\mathbf{w}}^* - \|\tilde{\mathbf{w}}^*\|_2^2\tilde{\mathbf{R}}\tilde{\mathbf{w}}^* = 0 \tag{54}$$

or

$$\tilde{\mathbf{R}}\tilde{\mathbf{w}}^* = \frac{\tilde{\mathbf{w}}^*}{\|\tilde{\mathbf{w}}^*\|_2^2}. \tag{55}$$

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

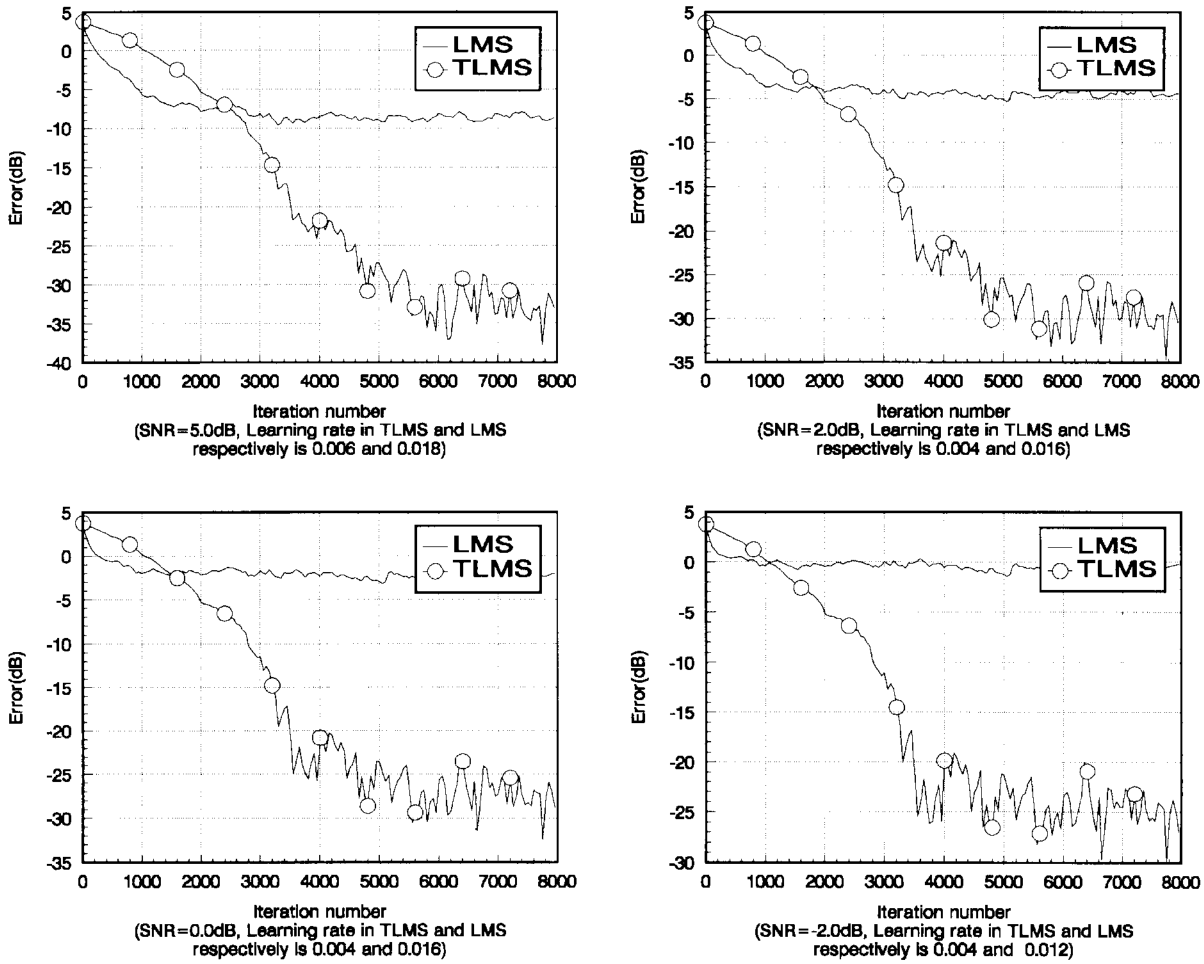

Fig. 2. *Continued.* Curves with large and small stable value are obtained by the LMS and the TLMS algorithm, respectively, where vertical and horizontal coordinates represent identification error and iteration number, respectively. Note that the oscillation in the learning curves originates from the noise in the learning process, the statistical rise and fall of the pseudo-random number, and the sensitivity of the estimate of the smallest singular value to the fluctuation of the psuedo-random sequence.

Formula (54) shows that $\tilde{\mathbf{w}}^*$ is an eigenvector of the augmented correlation matrix. Let

$$\mathbf{u}(t) = \mathbf{V}^T \tilde{\mathbf{w}}(t). \tag{56}$$

From (43), (51), and (56), we have

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{u}(t) - \|\mathbf{u}(t)\|_2^2 \mathbf{D}\, \mathbf{u}(t). \tag{57}$$

It is easily shown that (57) has $(n+1)$ equilibrium points. Let the ith equilibrium point of (57) be

$$\overline{\mathbf{u}}_i = \left[0, \cdots, 0, \pm\frac{1}{\sqrt{\lambda_i}}, 0, \cdots, 0\right]^T \quad (i = 1, 2, \cdots, n+1). \tag{58}$$

Then, the ith equilibrium point of (43) is

$$\tilde{\mathbf{w}}^* = \pm\frac{1}{\sqrt{\lambda_i}}\, \mathbf{v}_i, \qquad (i = 1, 2, \cdots, n+1). \tag{59}$$

It is obvious that $E(\pm(1/\sqrt{\lambda_i})\, \mathbf{v}_i) = (1+\ln\, \lambda_i)/2$. Within the neighborhood near the ith point of (57), $\mathbf{u}(t)$ be represented as

$$\mathbf{u}(t) = \overline{\mathbf{u}}_i + \delta(t) \tag{60}$$

where $\delta(t)$ is the disturbance vector near the equilibrium point. Substituting (60) into (57), we can obtain

$$\frac{d\delta(t)}{dt} = \delta(t) - \frac{2\delta_i(t)}{\sqrt{\lambda_i}}\, \mathbf{D}\overline{\mathbf{u}}_i - \frac{1}{\lambda_i}\, \mathbf{D}\delta(t) \tag{61}$$

where $\delta_i(t)$ is the ith component of $\delta(t)$. The above formula has discarded the higher order terms of $\delta(t)$ and used the equilibrium equation

$$\overline{\mathbf{u}}_i - \|\overline{\mathbf{u}}_i\|_2^2 \mathbf{D}\overline{\mathbf{u}}_i = 0. \tag{62}$$

The components of $\delta(t)$ are governed by equation

$$\begin{aligned} \frac{d\delta_i(t)}{dt} &= -2\delta_i(t) \\ \frac{d\delta_j(t)}{dt} &= \left(1 - \frac{\lambda_j}{\lambda_i}\right)\delta_j(t), \qquad j \neq i,\ j = 1, 2, \cdots, n+1 \end{aligned} \tag{63}$$

when $i < n+1$, $j > i$, $\delta_j(t)$ exponentially increases, whereas $i < n+1$, $j < i$, $\delta_j(t)$ exponentially decreases as $t \to \infty$ in (63). Thus, the ith equilibrium point is a saddle point.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

When $i = n+1$, the above formulae are changed into

$$\begin{aligned} \frac{d\delta_j(t)}{dt} &= \left(1 - \frac{\lambda_j}{\lambda_{n+1}}\right)\delta_j(t) \\ \frac{d\delta_{n+1}(t)}{dt} &= -2\delta_{n+1}(t) \end{aligned} \qquad j = 1, 2, \cdots, n. \tag{64}$$

Obviously, $\delta_j(t)(j = 1, \cdots, n+1)$ in (64) exponentially decreases with time. This shows that the $(n+1)$th equilibrium point is the only stable point of (57). Since a practical system is certainly corrupted by noise or interference [see (57)], (43) is not stable at any saddle point. From the above reasoning and from the corollary, we can conclude that $\mathbf{u}(t)$ of (57) globally asymptotically converges to the $(n+1)$th stable equilibrium point, i.e., $\tilde{\mathbf{w}}(t)$ of (43) globally asymptotically tends to the point

$$\tilde{\mathbf{w}}^* = \pm\frac{1}{\sqrt{\lambda_{n+1}}}\,\mathbf{e} = \pm\frac{1}{\sqrt{\lambda_{\min}}}\,\mathbf{e}. \tag{65}$$

This completes the proof of the theorem.

V. SIMULATIONS

In the simulations, the system identification shown in Fig. 1 is discussed. For a causal linear system, its input $x(k)$ and impulse response $h(k)$ can represent its output $d(k)$, i.e.,

$$d(k) = \sum_{l=0}^{\infty} h(l)x(k-l).$$

In the above equation, the real impulse response is unknown and remains to be identified. Let the length of $h(k)$ be N; then, we have

$$d(k) = \sum_{l=0}^{N-1} h(l)x(k-l)$$

as the output of the real system. The observational value of the input and of the output is $\tilde{x}(k) = x(k) + \Delta x(k)$ and $\tilde{d}(k) = d(k) + \Delta d(k)$, respectively. Here, $\Delta x(k)$ and $\Delta d(k)$ are, respectively, the interference of the input and of the output. The total adaptive filter is on the basis of

$$\min E\{\|[\Delta\mathbf{x}^T(k)|\Delta d(k)]\|_F\}$$

where

$$\mathbf{x}(k) = [x(k), x(k-1), \cdots, x(k-N+1)]^T.$$

The TLMS algorithm can be used to solve the above optimization problem. Let the impulse response of a known system be $\mathbf{h} = [-0.3, -0.9, 0.8, -0.7, 0.6]^T$ and its input and interference be a independent zero-mean white Gaussian psuedostochastic process. Assume that the SNR of the input is equal to the SNR of the output. The TLMS algorithm can derive the TLS solutions listed in Table I, whereas $\mathbf{h}_{pp}$ is derived by the LMS algorithm and

$$\begin{aligned} \text{SNR} &= 20\,\log\left(E\{\|\mathbf{x}(k)\|^2\}/E\{\|\Delta\mathbf{x}(k)\|^2\}\right) \\ \text{error1} &= 20\,\log\left(\|\mathbf{h} - \mathbf{h}_p\|^2\right) \\ \text{error2} &= 20\,\log\left(\|\mathbf{h} - \mathbf{h}_{pp}\|^2\right). \end{aligned}$$

The curves of convergence of error1 and error2 are shown in Fig. 2, where the horizontal coordinate represents the iteration number. It is obvious that the TLMS algorithm is advantageous over the LMS algorithm for this problem. The results show that the demerits of the TLMS algorithm are the slow convergence in the first segment of the learning curves and the sensitivity of the estimate of the smallest singular value to the statistical fluctuation and error.

VI. CONCLUSIONS

This paper proposes a total adaptive algorithm based on the total minimum mean-squares error. While input and output have interference, performance of the TLMS algorithm is obviously advantageous over the LMS algorithm. Since the assumption that the input and the output have noise is realistic, this TLMS algorithm has extensive applicability. The TLMS algorithm is also simple and only requires about $4n$-multiplication in each iteration. From a statistical analysis and stability study, we can know that if an appropriate learning rate μ is selected, the TLMS algorithm will be globally asymptotically convergent.

ACKNOWLEDGMENT

The authors wish to thank Prof. A. Sugiyama, Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the anonymous reviewers of this paper for their constructive criticism and suggestions for improvements.

REFERENCES

[1] B. Widrow, "Adaptive filters," in *Aspects of Network and System Theory*, N. de Claris and E. Kalman, Eds. New York: Holt, Rinehart, and Winston, 1971.
[2] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[3] K. Pearson, "On lines and planes of closest fit to points in space," *Philos. Mag.*, vol. 2, pp. 559–572, 1901.
[4] G. Holub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM Numer. Anal.*, vol. 17, no. 6, pp. 883–893, Dec. 1980.
[5] S. Van Huffel and J. Vanderwalle, "The total least squares problem, computational aspects and analysis," in *SIAM Frontiers in Applied Mathematics*. Philadelphia, PA: SIAM, 1991.
[6] M. A. Rahman and K. B. Yu, "Total least square approach for frequency estimation using linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1440–1454, 1987.
[7] T. J. Abatzoglou and J. M. Mendel, "Constrained total least squares," in *Proc. ICASSP*, Dallas, TX, Apr. 6–9, 1987, pp. 1485–1488.
[8] T. J. Abatzoglou, " Frequency superresolution performance of the constrained total least squares method," in *Proc. ICASSP*, 1990, pp. 2603–2605.
[9] N. K. Bose, H. C. Kim, and H. M. Valenzuela, "Recursive total least square algorithm for image reconstruction," *Multidim. Syst. Signal Process.*, vol. 4, pp. 253–268, July 1993.
[10] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 5, pp. 267–273, 1982.
[11] ——, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 15, pp. 927–935, 1992.
[12] G. A. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by self-organizing neural network," *IEEE Comput.*, vol. 21, pp. 77–88, 1988.
[13] T. Kohonen, K. Makisara, and T. Saramaki, "Phontopic maps insightful representation of phonologically features for speech recognition," in *Proc. 7th Int. Conf. Pattern Recoil.*, Montreal, P.Q., Canada, pp. 182–185.
[14] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biolog. Cybern.*, vol. 36, no. 2, pp. 193–202,1980.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.

[15] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of random matrix," *J. Math. Anal. Appl.,* vol. 106, pp. 69–94, 1985.
[16] P. Baldi and K. Hornik, "Neural network and principal component analysis: Learning form example without local minimal," *Neural Networks,* vol. 2, no. 1, pp. 52–58, 1989.
[17] P. Foldiak, "Adaptive network for optimal linear feature extraction," in *Proc. Int. Joint Conf. Neural Networks,* Washington, DC, 1989, vol. 1, pp. 401–405.
[18] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforword neural network," *Neural Networks,* vol. 2, pp. 459–473, 1989.
[19] J. Rubner and P. Tavan, "A self-organizing network for principal component analysis," *Europhys. Lett.,* vol. 10, no. 7, pp. 693–698, 1989.
[20] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications.* New York: Wiley, 1996.
[21] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks,* vol. 8, no. 4, pp. 549–562, 1995.
[22] N. K. Bose and P. Liang, *Neural Networks Fundamentals: Graphs, Algorithms, and Applications.* New York: McGraw-Hill, 1996.
[23] L. Xu, E. Oja, and C. Y. Suen, "Modified Hebbian learning for curve and surface fitting," *Neural Networks,* vol. 5, pp. 441–457, 1992.
[24] L. Xu, A. Kryzak, and E. Oja, "Neural-net method for dual surface pattern recognition," in *Proc. Int. Joint Conf. Neural Networks,* Seattle, WA, July 1992, vol. 2, pp. II379–II384.
[25] F. Deprettere, Ed., *SVD and Signal Processing, Algorithms, Applications, and Architecture.* Amsterdam, The Netherlands: North Holland, 1988.
[26] M. G. Larimore, "Adaptive convergence of spectral estimation based on Pisarenko harmonic retrieval," *IEEE Trans. Acoust., Speech, Signal Processing* vol. ASSP-31, pp. 955–962, 1983.
[27] S. Haykin, *Adaptive Filter Theory,* 2nd Ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[28] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.,* vol. 22, pp. 400–407, 1951.
[29] J. P. Lasalle, *The Stability of Dynamical Systems.* Philadelphia, PA: SIAM, 1976.
[30] K. Q. Gao, M. Omair Ahmad, and M. N. S. Swamy, "Learning algorithm for total least squares adaptive signal processing," *Electron. Lett.,* vol. 28, pp. 430–432, 1992.
[31] ——, "A constrained anti-Hebbian learning algorithm for total least-squares estimation with applications to adaptive FIR and IIF filtering," *IEEE Trans. Circuits Syst. II,* vol. 41, 1994.
[32] Q. F. Zhang and Z. Bao, "Neural networks for solving TLS problems," *J. China Inst. Commun.,* vol. 16, no. 4, 1994 (in Chinese).

Da-Zheng Feng was born on December 9, 1959. He received the B.E. degree in water-electrical engineering in 1982 from Xi'an Science and Technology University, Xi'an, China, the M.S. degree in electrical engineering in 1986 from Xi'an Jiaotong University, and the Ph.D. degree in electronic engineering in 1995 from Xidian University, Xi'an. His Ph.D. research was in the field of neural formation processing and adaptive signal processing. His dissertation described the distributed parameter neural network theory and applications.

Since 1996, he has been a Post-Doctoral Research Affiliate and Associate Professor at the Department of Mechatronic Engineering, Xi'an Jiaotong University. He has published more than 30 journal papers. His research interests include signal processing, intelligence information processing, legged machine distributed control, and an artificial brain.

Dr. Feng received the Outstanding Ph.D. Student award, presented by the Chinese Ministry of the Electronics Industry, in December 1995.

Zheng Bao (M'80–SM'85) received the B.S. degree in radar engineering from Xidian University, Xi'an, China, in 1953.

He is now a Professor at Xidian University. He has published more than 200 journal papers and is the author of ten books. He has worked in a number of areas, including radar systems, signal processing, and neural networks. His research interests include array signal processing, radar signal processing, SAR, and ISAR imaging.

Prof. Bao is a member of the Chinese Academy of Sciences.

Li-Cheng Jiao (M'86–SM'90) received the B.S. degree in electrical engineering and computer science from Shanghai Jiaotong University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

He is now a Professor at Xidian University, Xi'an. He has published more than 100 papers and is the author of four books. His research interests include nonlinear networks and systems, neural networks, intelligence information processing, and adaptive signal processing.

Dr. Jiao is a member of the China Neural Networks Council.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on August 14, 2009 at 08:20 from IEEE Xplore. Restrictions apply.