Arab J Sci Eng (2017) 42:2809–2816
DOI 10.1007/s13369-017-2453-y

RESEARCH ARTICLE - ELECTRICAL ENGINEERING

LMS-Based Variable Step-Size Algorithms: A Unified Analysis Approach

Muhammad Omer Bin Saeed[1]

Received: 4 August 2016 / Accepted: 5 March 2017 / Published online: 17 March 2017

Abstract Several variable step-size strategies have been suggested in the literature to improve the performance of the least-mean-square (LMS) algorithm. Although they enhance performance, a major drawback is the complexity in the theoretical analysis of these algorithms. Researchers use several assumptions to find closed-form analytical solutions. This work presents a unified approach for the analysis of variable step-size LMS algorithms. The approach is then applied to several variable step-size strategies, and theoretical and simulation results are compared.

Keywords Variable step-size · Least-mean-square algorithms · Mean-square analysis · Steady-state analysis

1 Introduction

Many algorithms have been proposed for estimation/system identification, with the LMS algorithm being the most popular as it is simple and effective [1]. However, a limiting factor of LMS is a trade-off between convergence speed and steady-state misadjustment. Various variable step-size (VSS) strategies have been proposed to rectify this problem. These strategies exhibit a high step-size initially for fast convergence but then reduce the step-size with time in order to achieve a low error performance [2–25]. Some algorithms proposed in literature are for specific applications [12,14,15,20–25]. There are several algorithms that are derived from a constraint on the cost function [2,7,8,10,14].

✉ Muhammad Omer Bin Saeed
m.binsaeed.1981@ieee.org

1 Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, Pakistan

In general, all VSS algorithms aim to improve performance at the cost of computational complexity. This trade-off is generally acceptable due to the improvement in performance. However, the additional complexity also results in difficulty in analyzing the algorithm. Authors use several assumptions to find closed-form solutions for the analysis of these algorithms. However, until now, each algorithm has been dealt with individually in order to find the steady-state misadjustment, leading to the steady-state excess-mean-square-error (EMSE). Similarly, the mean-square analysis for each algorithm has to be performed individually. An exact method of analysis has been proposed in [26,27]. Even though the results are very accurate, this method is highly complex as well as algorithm specific.

Inspired from the similarity of the assumptions used for the analyses of these algorithms, this work presents a unified approach for the analysis of VSS LMS algorithms. The proposed generalized analysis can be applied to most existing as well as any forthcoming VSS algorithms.

The rest of the paper is divided as follows. Section 2 presents a working system model and problem statement. Section 3 details the complete theoretical analysis for VSS LMS algorithms. Simulation results are presented in Sect. 4. Section 5 concludes this work.

2 System Model

The unknown system is modeled as an FIR filter in the form of a vector, $\mathbf{w}_o$, of size $(M \times 1)$. The input to the unknown system at any given time i is a $(1 \times M)$ complex-valued regressor vector, $\mathbf{u}(i)$. The observed output of the system is a noise corrupted scalar, $d(i)$, given by

$$d(i) = \mathbf{u}(i)\mathbf{w}_o + v(i), \tag{1}$$

where $v(i)$ is the complex-valued zero-mean additive noise.

The VSS LMS algorithm iteratively estimates the unknown system with an update equation given by

$$\mathbf{w}(i+1) = \mathbf{w}(i) + \mu(i)e(i)\,\mathbf{u}^*(i), \tag{2}$$

$$\mu(i+1) = f\{\mu(i)\}, \tag{3}$$

where $\mathbf{w}(i)$ is the estimate of the unknown system vector at time i, $e(i) = d(i) - \mathbf{u}(i)\mathbf{w}(i)$ is the instantaneous error and $(.)^*$ is the complex conjugate transpose operator. The step-size is denoted by $\mu(i)$, and $f\{.\}$ is a function that defines the update equation for the step-size and is different for every VSS algorithm.

While performing the analysis, the input regressor is assumed to be independent of the estimated vector. For the VSS algorithms, the control parameters are chosen such that the step-size and the input regressor vector are assumed to be asymptotically independent of each other. This helps in forming a closed-form steady-state solution that closely matches with the simulation results. For some VSS algorithms, the analytical and simulation results closely match during the transient stage as well but this is not always the case. The results are still acceptable for all algorithms as a closed-form solution is obtained.

The main objective of this work is to provide a generalized analysis for VSS algorithms, in lieu with the assumptions mentioned above. The results of this analysis can be applied to VSS algorithms in general as will be shown through specific examples.

3 Proposed Analysis

The weight-error vector is given by

$$\tilde{\mathbf{w}}(i) = \mathbf{w}_o - \mathbf{w}(i). \tag{4}$$

Using (4) in (2) results in

$$\tilde{\mathbf{w}}(i+1) = [\mathbf{I}_M - \mu(i)\mathbf{u}^*(i)\mathbf{u}(i)]\tilde{\mathbf{w}}(i) - \mu(i)\mathbf{u}^*(i)v(i), \tag{5}$$

where $\mathbf{I}_M$ is an identity matrix of size M. Before beginning the analysis, another assumption is made, without loss of generality. The input data are assumed to be circular Gaussian. The autocorrelation matrix of the input regressor vector is given by $\mathbf{R}_u = \mathbb{E}[\mathbf{u}^*(i)\mathbf{u}(i)]$, where $\mathbb{E}[.]$ is the expectation operator. Using the Gaussian data assumption, $\mathbf{R}_u$ can be decomposed into its component matrices of eigenvalues and eigenvectors. Thus, $\mathbf{R}_u = \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^*$, where $\mathbf{T}$ is the matrix of eigenvectors such that $\mathbf{T}^*\mathbf{T} = \mathbf{I}_M$ and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues. Using the matrix $\mathbf{T}$, the following transformations are made

$$\bar{\mathbf{w}}(i) = \mathbf{T}^*\tilde{\mathbf{w}}(i),\ \bar{\mathbf{u}}(i) = \mathbf{u}(i)\mathbf{T}$$

The weight-error update equation thus becomes

$$\bar{\mathbf{w}}(i+1) = [\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)]\bar{\mathbf{w}}(i) - \mu(i)\bar{\mathbf{u}}^*(i)v(i). \tag{6}$$

3.1 Mean Analysis

Applying the expectation operator to (6) results in

$$\begin{aligned}\mathbb{E}[\bar{\mathbf{w}}(i+1)] &= \mathbb{E}\left[\left\{\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right\}\bar{\mathbf{w}}(i) - \mu(i)\bar{\mathbf{u}}^*(i)v(i)\right] \\ &= \left\{\mathbf{I}_M - \mathbb{E}\left[\mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right]\right\}\mathbb{E}[\bar{\mathbf{w}}(i)], \end{aligned} \tag{7}$$

where the data independence assumption separates $\mathbb{E}[\mathbf{w}(i)]$ from the rest of the variables. The second term is 0 as additive noise is independent and zero-mean. Using the assumption that the step-size control parameters are chosen in such a way that the step-size and the input regressor data are asymptotically independent, (7) is further simplified as

$$\mathbb{E}[\bar{\mathbf{w}}(i+1)] = \{\mathbf{I}_M - \mathbb{E}[\mu(i)]\,\boldsymbol{\Lambda}\}\,\mathbb{E}[\bar{\mathbf{w}}(i)], \tag{8}$$

where $\boldsymbol{\Lambda} = \mathbb{E}\left[\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right]$. The sufficient condition for stability is evaluated from (8) and is given by

$$0 < \mathbb{E}[\mu(i)] < \frac{2}{\beta_{max}}, \tag{9}$$

where β_{max} is the maximum eigenvalue of $\boldsymbol{\Lambda}$.

3.2 Mean-Square Analysis

Taking the expectation of the squared weighted l_2-norm of (6) yields

$$\begin{aligned}&\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\Sigma}}\right] \\ &= \mathbb{E}\left[\bar{\mathbf{w}}^*(i)\boldsymbol{\Sigma}'\bar{\mathbf{w}}(i)\right] + \mathbb{E}\left[\mu^2(i)v^2(i)\bar{\mathbf{u}}(i)\boldsymbol{\Sigma}\bar{\mathbf{u}}^*(i)\right] \\ &\quad - \mathbb{E}\left[\mu(i)v(i)\bar{\mathbf{u}}(i)\boldsymbol{\Sigma}\left\{\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right\}\bar{\mathbf{w}}(i)\right] \\ &\quad - \mathbb{E}\left[\bar{\mathbf{w}}^*(i)\left\{\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right\}\boldsymbol{\Sigma}\mu(i)v(i)\bar{\mathbf{u}}^*(i)\right],\end{aligned} \tag{10}$$

where $\|.\|$ is the l_2-norm operator and $\boldsymbol{\Sigma}$ is a weighting matrix. The weighting matrix $\boldsymbol{\Sigma}'$ is given by

$$\begin{aligned}\boldsymbol{\Sigma}' &= \left\{\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right\}^*\boldsymbol{\Sigma}\left\{\mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\right\} \\ &= \mathbf{I}_M - \mu(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\boldsymbol{\Sigma} - \mu(i)\boldsymbol{\Sigma}\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i) \\ &\quad + \mu^2(i)\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\boldsymbol{\Sigma}\bar{\mathbf{u}}^*(i)\bar{\mathbf{u}}(i)\end{aligned} \tag{11}$$

The last two terms in (10) are zero due to independence of additive noise. Using the data independence assumption, the remaining two terms can be simplified as

$$\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\Sigma}}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\boldsymbol{\Sigma}'}\right] + \sigma_v^2 \mathbb{E}\left[\mu^2(i)\right] \mathrm{Tr}\left\{\boldsymbol{\Sigma}\boldsymbol{\Lambda}\right\}, \tag{12}$$

where σ_v^2 is the additive noise variance, $\mathrm{Tr}\{.\}$ is the *trace* operator and $\mathbb{E}\left[\mathbf{u}(i)\boldsymbol{\Sigma}\mathbf{u}^T(i)\right] = \mathrm{Tr}\{\boldsymbol{\Sigma}\boldsymbol{\Lambda}\}$. Once again invoking the data independence assumption, we write $\mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\boldsymbol{\Sigma}'}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\mathbb{E}[\boldsymbol{\Sigma}']}\right]$. Further, taking $\mathbb{E}\left[\boldsymbol{\Sigma}'\right] = \boldsymbol{\Sigma}'$ and simplifying, (11) is rewritten as

$$\boldsymbol{\Sigma}' = \mathbf{I}_M - 2\mathbb{E}\left[\mu(i)\right]\boldsymbol{\Lambda}\boldsymbol{\Sigma} + \mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\Lambda}\mathrm{Tr}\left[\boldsymbol{\Sigma}\boldsymbol{\Lambda}\right] + \mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}. \tag{13}$$

Using the diag{.} operator, (12) is simplified as

$$\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\mathbf{F}(i)\boldsymbol{\sigma}}\right] + \sigma_v^2 \mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \tag{14}$$

where $\boldsymbol{\sigma} = \mathrm{diag}\{\boldsymbol{\Sigma}\}$, $\boldsymbol{\lambda} = \mathrm{diag}\{\boldsymbol{\Lambda}\}$ and the weighting matrix $\boldsymbol{\Sigma}'$ is replaced with $\mathrm{diag}\left\{\boldsymbol{\Sigma}'\right\} = \boldsymbol{\sigma}' = \mathbf{F}(i)\boldsymbol{\sigma}$, where $\mathbf{F}(i)$ is given by

$$\mathbf{F}(i) = \mathbf{I}_M - 2\mathbb{E}\left[\mu(i)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(i)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right]. \tag{15}$$

Now, using (14) and (15), the analysis iterates as

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(0)\|^2_{\boldsymbol{\sigma}}\right] &= \|\mathbf{w}_o\|^2_{\boldsymbol{\sigma}}, \\ \mathbf{F}(0) &= \mathbf{I}_M - 2\mu(0)\boldsymbol{\Lambda} + \mu^2(0)\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right],\end{aligned}$$

where $\mathbb{E}\left[\mu(0)\right] = \mu(0)$ and $\mathbb{E}\left[\mu^2(0)\right] = \mu^2(0)$ as this is the initial step-size value. The first iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(1)\|^2_{\boldsymbol{\sigma}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(0)\|^2_{\mathbf{F}(0)\boldsymbol{\sigma}}\right] + \sigma_v^2\mu^2(0)\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{F}(0)\boldsymbol{\sigma}} + \sigma_v^2\mu^2(0)\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ \mathbf{F}(1) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(1)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(1)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right],\end{aligned}$$

where the updates $\mathbb{E}\left[\mu(1)\right]$ and $\mathbb{E}\left[\mu^2(1)\right]$ are obtained from the particular step-size update equation of the VSS algorithm being used. Similarly, the second iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(2)\|^2_{\boldsymbol{\sigma}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(1)\|^2_{\mathbf{F}(1)\boldsymbol{\sigma}}\right] + \sigma_v^2\mathbb{E}\left[\mu^2(1)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{F}(0)\mathbf{F}(1)\boldsymbol{\sigma}} + \sigma_v^2\mu^2(0)\boldsymbol{\lambda}^T\mathbf{F}(1)\boldsymbol{\sigma} \\ &\quad + \sigma_v^2\mathbb{E}\left[\mu^2(1)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{F}(0)\mathbf{F}(1)\boldsymbol{\sigma}} \\ &\quad + \sigma_v^2\boldsymbol{\lambda}^T\left\{\mu^2(0)\mathbf{F}(1) + \mathbb{E}\left[\mu^2(1)\right]\mathbf{I}_M\right\}\boldsymbol{\sigma} \\ \mathbf{F}(2) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(2)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(2)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right].\end{aligned}$$

Continuing, the third iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(3)\|^2_{\boldsymbol{\sigma}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(2)\|^2_{\mathbf{F}(2)\boldsymbol{\sigma}}\right] + \sigma_v^2\mathbb{E}\left[\mu^2(2)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(2)\mathbf{F}(2)\boldsymbol{\sigma}} + \sigma_v^2\mathbb{E}\left[\mu^2(2)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &\quad + \sigma_v^2\boldsymbol{\lambda}^T\left\{\sum_{k=0}^{1}\mathbb{E}\left[\mu^2(k)\right]\prod_{m=k+1}^{2}\mathbf{F}(m)\right\}\boldsymbol{\sigma} \\ \mathbf{F}(3) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(3)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(3)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right],\end{aligned}$$

where the weighting matrix $\mathbf{A}(2) = \mathbf{F}(0)\mathbf{F}(1)$. The fourth iterative update is then given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(4)\|^2_{\boldsymbol{\sigma}}\right] &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(3)\mathbf{F}(3)\boldsymbol{\sigma}} + \sigma_v^2\mathbb{E}\left[\mu^2(3)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\ &\quad + \sigma_v^2\boldsymbol{\lambda}^T\left\{\sum_{k=0}^{2}\mathbb{E}\left[\mu^2(k)\right]\prod_{m=k+1}^{3}\mathbf{F}(m)\right\}\boldsymbol{\sigma} \\ \mathbf{F}(4) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(4)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(4)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right],\end{aligned}$$

where the weighting matrix $\mathbf{A}(3) = \mathbf{A}(2)\mathbf{F}(2)$. Now, from the third and fourth iterative updates, we generalize the recursion for the ith update as

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\boldsymbol{\sigma}}\right] &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(i-1)\mathbf{F}(i-1)\boldsymbol{\sigma}} \\ &\quad + \sigma_v^2\mathbb{E}\left[\mu^2(i-1)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \qquad (16) \\ &\quad + \sigma_v^2\boldsymbol{\lambda}^T\left\{\sum_{k=0}^{i-2}\mathbb{E}\left[\mu^2(k)\right]\prod_{m=k+1}^{i-1}\mathbf{F}(m)\right\}\boldsymbol{\sigma} \\ \mathbf{F}(i) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(i)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(i)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right]. \qquad (17)\end{aligned}$$

Similarly, the recursion for the $(i+1)^{th}$ update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\right] &= \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(i)\mathbf{F}(i)\boldsymbol{\sigma}} \\ &\quad + \sigma_v^2\mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \qquad (18) \\ &\quad + \sigma_v^2\boldsymbol{\lambda}^T\left\{\sum_{k=0}^{i-1}\mathbb{E}\left[\mu^2(k)\right]\prod_{m=k+1}^{i}\mathbf{F}(m)\right\}\boldsymbol{\sigma} \\ \mathbf{F}(i+1) &= \mathbf{I}_M - 2\mathbb{E}\left[\mu(i+1)\right]\boldsymbol{\Lambda} + \mathbb{E}\left[\mu^2(i+1)\right]\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right]. \qquad (19)\end{aligned}$$

Subtracting (16) from (18) and simplifying the terms gives the final recursive update equation

$$\begin{aligned}
&\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\right] - \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\boldsymbol{\sigma}}\right] \\
&\quad = \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(i)\mathbf{F}(i)\boldsymbol{\sigma}} - \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(i-1)\mathbf{F}(i-1)\boldsymbol{\sigma}} \\
&\quad\quad + \sigma_v^2 \mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} - \sigma_v^2 \mathbb{E}\left[\mu^2(i-1)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\
&\quad\quad + \sigma_v^2 \boldsymbol{\lambda}^T \left\{ \sum_{k=0}^{i-1} \mathbb{E}\left[\mu^2(k)\right] \prod_{m=k+1}^{i} \mathbf{F}(m) \right\} \boldsymbol{\sigma} \\
&\quad\quad - \sigma_v^2 \boldsymbol{\lambda}^T \left\{ \sum_{k=0}^{i-2} \mathbb{E}\left[\mu^2(k)\right] \prod_{m=k+1}^{i-1} \mathbf{F}(m) \right\} \boldsymbol{\sigma}. \qquad (20)
\end{aligned}$$

Simplifying (20) and rearranging the terms gives the final recursive update equation

$$\begin{aligned}
\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\boldsymbol{\sigma}}\right] + \|\bar{\mathbf{w}}_o\|^2_{\mathbf{A}(i)[\mathbf{F}(i)-\mathbf{I}_M]\boldsymbol{\sigma}} \\
&\quad + \sigma_v^2 \mathbb{E}\left[\mu^2(i)\right]\boldsymbol{\lambda}^T\boldsymbol{\sigma} \\
&\quad + \sigma_v^2 \boldsymbol{\lambda}^T \mathbf{B}(i)\left\{\mathbf{F}(i) - \mathbf{I}_M\right\}\boldsymbol{\sigma}, \qquad (21)
\end{aligned}$$

where

$$\mathbf{B}(i) = \left\{ \mathbb{E}\left[\mu^2(i-1)\right]\mathbf{I}_M + \sum_{k=0}^{i-2} \mathbb{E}\left[\mu^2(k)\right] \prod_{m=i-1}^{k+1} \mathbf{F}(m) \right\}. \qquad (22)$$

The final set of iterative equations for the mean-square learning curve is given by (21), (17) and

$$\mathbf{A}(i+1) = \mathbf{A}(i)\mathbf{F}(i) \qquad (23)$$

$$\mathbf{B}(i+1) = \mathbb{E}\left[\mu^2(i)\right]\mathbf{I}_M + \mathbf{B}(i)\mathbf{F}(i). \qquad (24)$$

Taking the weighting matrix $\boldsymbol{\Sigma} = \mathbf{I}_M$ results in the mean-square-deviation (MSD) while taking the weighting matrix $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}$ gives the EMSE.

It should be noted here that unlike the analysis given in [1] for the LMS algorithm, the weighting matrix $\mathbf{F}(i)$ is not constant. As a result, the Cayley–Hamilton theorem is not applicable. In this context, (17) and (21)–(24) are very significant contributions of this work.

3.3 Steady-State Analysis

At steady state, the recursions (14) and (15) become

$$\mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\boldsymbol{\sigma}}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\mathbf{F}_{ss}\boldsymbol{\sigma}}\right] + \sigma_v^2 \mu_{ss}^2 \boldsymbol{\lambda}^T \boldsymbol{\sigma} \qquad (25)$$

$$\mathbf{F}_{ss} = \mathbf{I}_M - 2\mu_{ss}\boldsymbol{\Lambda} + \mu_{ss}^2\left[\boldsymbol{\Lambda}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^T\right], \qquad (26)$$

Table 1 Step-size update equations for the VSSLMS algorithms

Algorithm	Step-size update equation
KJ [4]	$\mu(i+1) = \alpha_{kj}\mu(i) + \gamma_{kj}e^2(i)$
NC [10]	$\theta_{nc}(i+1) = (1-\alpha_{nc})\theta_{nc}(i) + \frac{\alpha_{nc}}{2}\left(e^2(i) - \sigma_v^2\right)$
	$\mu(i+1) = \mu_0\left(1 + \gamma_{nc}\theta_{nc}(i+1)\right)$
	$A_q(i) = a_q A_q(i-1) + e^2(i)$
VSQ [18]	$B_q(i) = b_q B_q(i-1) + e^2(i)$
	$\mu(i+1) = \alpha_q\mu(i) + \gamma_q\frac{A_q(i)}{B_q(i)}$
Sp [22]	$\mu(i+1) = \alpha_{sp}\mu(i) + \gamma_{sp}\lvert e(i)\rvert$

where the subscript ss denotes steady state. Simplifying (25) further gives

$$\mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\boldsymbol{\sigma}}\right] = \sigma_v^2\mu_{ss}^2\boldsymbol{\lambda}^T\left[\mathbf{I}_M - \mathbf{F}_{ss}\right]^{-1}\boldsymbol{\sigma}, \qquad (27)$$

which defines the steady-state MSD if $\boldsymbol{\Sigma} = \mathbf{I}_M$ and steady-state EMSE if $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}$.

3.4 Steady-State Step-Size Analysis

The analysis presented in the above section has been generic for any VSS algorithm. In this section, 4 different VSS algorithms are chosen to present the steady-state analysis for the step-size. These steady-state step-size values are then directly inserted into (27) and (26). The 4 different VSS algorithms and their step-size update equations are given in Table 1. The first algorithm denoted *KJ* is the work of Kwong and Johnston [4]. The *NC* algorithm refers to the noise-constrained LMS algorithm [10]. The *VSQ* algorithm is the variable step-size quotient LMS algorithm [18]. Finally, *Sp* refers to the Sparse VSSLMS algorithm of [22].

The expectation operator is applied to each of the VSS algorithms. For the *KJ* algorithm, the resultant equation is given by

$$\begin{aligned}
\mathbb{E}\left[\mu(i+1)\right] &= \mathbb{E}\left[\alpha_{kj}\mu(i) + \gamma_{kj}e^2(i)\right] \\
&= \alpha_{kj}\mathbb{E}\left[\mu(i)\right] + \gamma_{kj}\mathbb{E}\left[e^2(i)\right] \\
&= \alpha_{kj}\mathbb{E}\left[\mu(i)\right] + \gamma_{kj}\left[\zeta(i) + \sigma_v^2\right], \qquad (28)
\end{aligned}$$

where $\zeta(i)$ denotes the EMSE. At steady state, (28) becomes

$$\mu_{ss} = \alpha_{kj}\mu_{ss} + \gamma_{kj}\left[\zeta_{ss} + \sigma_v^2\right]. \qquad (29)$$

Assuming, without loss of generality, that the EMSE value at steady state is small enough to be ignored, and rearranging (29) gives the final steady-state equation

$$\mu_{ss} \approx \frac{\gamma_{kj}}{1-\alpha_{kj}}\sigma_v^2. \qquad (30)$$

Next, the expectation operator is applied to the set of equations defining the *NC* algorithm, which gives

$$\mathbb{E}\left[\theta_{nc}(i+1)\right] = \mathbb{E}\left[(1-\alpha_{nc})\,\theta_{nc}(i) + \frac{\alpha_{nc}}{2}\left(e^2(i)-\sigma_v^2\right)\right], \tag{31}$$

$$\mathbb{E}\left[\mu\,(i+1)\right] = \mathbb{E}\left[\mu_0\,(1+\gamma_{nc}\,\theta_{nc}(i+1))\right]. \tag{32}$$

Simplifying (31) and (32) gives

$$\begin{aligned}\mathbb{E}\left[\theta_{nc}(i+1)\right] &= (1-\alpha_{nc})\mathbb{E}\left[\theta_{nc}(i)\right] + \frac{\alpha_{nc}}{2}\left(\mathbb{E}\left[e^2(i)\right]-\sigma_v^2\right)\\ &= (1-\alpha_{nc})\mathbb{E}\left[\theta_{nc}(i)\right] + \frac{\alpha_{nc}}{2}\left(\zeta(i)+\sigma_v^2-\sigma_v^2\right)\\ &= (1-\alpha_{nc})\mathbb{E}\left[\theta_{nc}(i)\right] + \frac{\alpha_{nc}}{2}\zeta(i) \end{aligned}\tag{33}$$

$$\mathbb{E}\left[\mu\,(i+1)\right] = \mu_0\left(1+\gamma_{nc}\mathbb{E}\left[\theta_{nc}(i+1)\right]\right). \tag{34}$$

At steady state, (33) and (34) become

$$\theta_{nc,ss} = (1-\alpha_{nc})\,\theta_{nc,ss} + \frac{\alpha_{nc}}{2}\zeta_{ss} \tag{35}$$

$$\mu_{ss} = \mu_0\left(1+\gamma_{nc}\,\theta_{nc,ss}\right). \tag{36}$$

Using the assumption that the EMSE value at steady state is small enough to be ignored, (35) simplifies to give $\theta_{nc,ss} = 0$. Thus, (36) simplifies to

$$\mu_{ss} \approx \mu_0. \tag{37}$$

Now we apply the expectation operator to the set of the equations for the *VSQ* algorithm that gives

$$\begin{aligned}\mathbb{E}\left[A_q(i)\right] &= \mathbb{E}\left[a_q A_q(i-1)+e^2(i)\right]\\ &= a_q\mathbb{E}\left[A_q(i-1)\right]+\mathbb{E}\left[e^2(i)\right]\\ &= a_q\mathbb{E}\left[A_q(i-1)\right]+\zeta(i)+\sigma_v^2, \end{aligned}\tag{38}$$

$$\begin{aligned}\mathbb{E}\left[B_q(i)\right] &= \mathbb{E}\left[b_q B_q(i-1)+e^2(i)\right],\\ &= b_q\mathbb{E}\left[B_q(i-1)\right]+\mathbb{E}\left[e^2(i)\right]\\ &= b_q\mathbb{E}\left[B_q(i-1)\right]+\zeta(i)+\sigma_v^2, \end{aligned}\tag{39}$$

$$\mathbb{E}\left[\mu\,(i+1)\right] = \mathbb{E}\left[\alpha_q\mu\,(i)\right]+\gamma_q\frac{A_q(i)}{B_q(i)}. \tag{40}$$

Rearranging and simplifying, we get the final equation as

$$\mathbb{E}\left[\mu\,(i+1)\right] = \alpha_q\mathbb{E}\left[\mu\,(i)\right] + \gamma_q\frac{a_q\mathbb{E}\left[A_q(i-1)\right]+\zeta(i)+\sigma_v^2}{b_q\mathbb{E}\left[B_q(i-1)\right]+\zeta(i)+\sigma_v^2}. \tag{41}$$

At steady state, (38)–(40) become

$$A_{q,ss} = a_q A_{q,ss}+\zeta_{ss}+\sigma_v^2, \tag{42}$$

$$B_{q,ss} = b_q B_{q,ss}+\zeta_{ss}+\sigma_v^2, \tag{43}$$

$$\mu_{ss} = \alpha_q\mu_{ss}+\gamma_q\frac{A_{q,ss}}{B_{q,ss}}. \tag{44}$$

Since the EMSE value is assumed to be small enough to be ignored at steady state, (42) and (43) are simplified as

$$A_{q,ss} = \frac{\sigma_v^2}{1-a_q}, \tag{45}$$

$$B_{q,ss} = \frac{\sigma_v^2}{1-b_q}. \tag{46}$$

Inserting (45) and (46) into (44) and simplifying, we get

$$\mu_{ss} \approx \frac{\gamma_q(1-b_q)}{(1-\alpha_q)(1-a_q)}. \tag{47}$$

Finally, we apply the expectation operator to the step-size update equation for the *Sp* algorithm to get

$$\begin{aligned}\mathbb{E}\left[\mu\,(i+1)\right] &= \mathbb{E}\left[\alpha_{sp}\mu\,(i)+\gamma_{sp}\left|e^2(i)\right|\right]\\ &= \alpha_{sp}\mathbb{E}\left[\mu\,(i)\right]+\gamma_{sp}\mathbb{E}\left[\left|e^2(i)\right|\right]\\ &= \alpha_{sp}\mathbb{E}\left[\mu\,(i)\right]+\gamma_{sp}\sqrt{\frac{2\sigma_v^2}{\pi}}, \end{aligned}\tag{48}$$

where $\mathbb{E}\left[\left|e^2(i)\right|\right] = \sqrt{2\sigma_v^2/\pi}$. At steady state, (48) becomes

$$\mu_{ss} = \alpha_{sp}\mu_{ss}+\gamma_{sp}\sqrt{\frac{2\sigma_v^2}{\pi}}. \tag{49}$$

Simplifying (49) gives

$$\mu_{ss} = \frac{\gamma_{sp}\sqrt{2\sigma_v^2/\pi}}{1-\alpha_{sp}}. \tag{50}$$

The set of equations after applying the expectation operator and simplifying is presented in Table 2. The approximate steady-state step-size equations are given in Table 3.

4 Results and Discussion

In this section, the analysis presented above will be tested upon the 4 VSS algorithms listed in Table 1. These algorithms are used in two different experiments to test the validity of the analysis. In the first experiment, MSD is plotted using (21) and compared with simulation results. The second experiment compares the steady-state simulation results with the

Table 2 Expectations of the update equations from Table 1

Algorithm	Expectation of update equation
KJ [4]	$\mathbb{E}\left[\mu\left(i+1\right)\right]=\alpha_{kj}\mathbb{E}\left[\mu\left(i\right)\right]+\gamma_{kj}\left[\zeta(i)+\sigma_v^2\right]$
NC [10]	$\mathbb{E}\left[\theta_{nc}(i+1)\right]=(1-\alpha_{nc})\mathbb{E}\left[\theta_{nc}(i)\right]+\frac{\alpha_{nc}}{2}\zeta(i)$
	$\mathbb{E}\left[\mu\left(i+1\right)\right]=\mu_0\left(1+\gamma_{nc}\mathbb{E}\left[\theta_{nc}(i+1)\right)\right]$
VSQ [18]	$\mathbb{E}\left[\mu\left(i+1\right)\right]=\alpha_q\mathbb{E}\left[\mu\left(i\right)\right]$
	$+\gamma_q\frac{a_q\mathbb{E}[A_q(i-1)]+\zeta(i)+\sigma_v^2}{b_q\mathbb{E}[B_q(i-1)]+\zeta(i)+\sigma_v^2}$
Sp [22]	$\mathbb{E}\left[\mu\left(i+1\right)\right]=\alpha_{sp}\mathbb{E}\left[\mu\left(i\right)\right]+\gamma_{sp}\sqrt{2\sigma_v^2/\pi}$

Table 3 Steady-state step-size values for equations from Table 1

Algorithm	Steady-state step-size value
KJ [4]	$\mu_{ss}\approx\frac{\gamma_{kj}}{1-\alpha_{kj}}\sigma_v^2.$
NC [10]	$\mu_{ss}\approx\mu_0.$
VSQ [18]	$\mu_{ss}\approx\frac{\gamma_q(1-b_q)}{(1-\alpha_q)(1-a_q)}.$
Sp [22]	$\mu_{ss}\approx\frac{\gamma_{sp}}{1-\alpha_{sp}}\sqrt{2\sigma_v^2/\pi}.$

Table 4 Step-size control parameters for the VSSLMS algorithms

Algorithm	SNR (dB)	Parameters
	All	$\gamma_{kj}=1e-3$
KJ [4]	0	$\alpha_{kj}=0.95$
	10	$\alpha_{kj}=0.97$
	20	$\alpha_{kj}=0.99$
NC [10]	All	$\gamma_{nc}=1, \alpha_{nc}=1e-2$
	All	$a_q=0.99, b_q=1e-3, \gamma_q=1e-3$
VSQ [18]	0	$\alpha_q=0.95$
	10	$\alpha_q=0.96$
	20	$\alpha_q=0.97$
	All	$\gamma_{sp}=1e-3$
Sp [22]	0	$\alpha_{sp}=0.95$
	10	$\alpha_{sp}=0.96$
	20	$\alpha_{sp}=0.99$

theoretical results obtained using (27). The length of the unknown vector is $M=4$. The signal-to-noise ratio (SNR) is varied between 0, 10 and 20 dB. The input regressor vector is a realization of a zero-mean Gaussian random variable with unit variance. The step-size control parameters used are given in Table 4. The values are slightly different in some cases in order to maintain the same convergence speed.

For the first experiment, the results are shown separately for each algorithm in Figs. 1, 2, 3 and 4. The results for the *KJ* algorithm of [4] show a slight mismatch during the transient stage, but this mismatch disappears at steady state. Since the parameters are chosen such that the step-size is asymptotically independent, this result is justified. The results for the

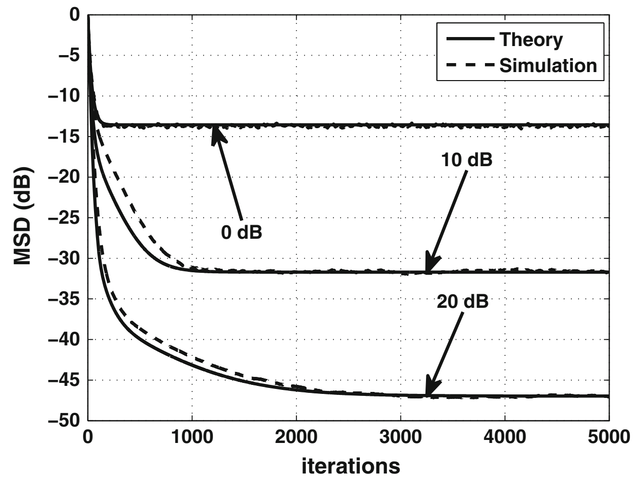

Fig. 1 Theory (21) v simulation MSD comparison for the KJ algorithm [4]

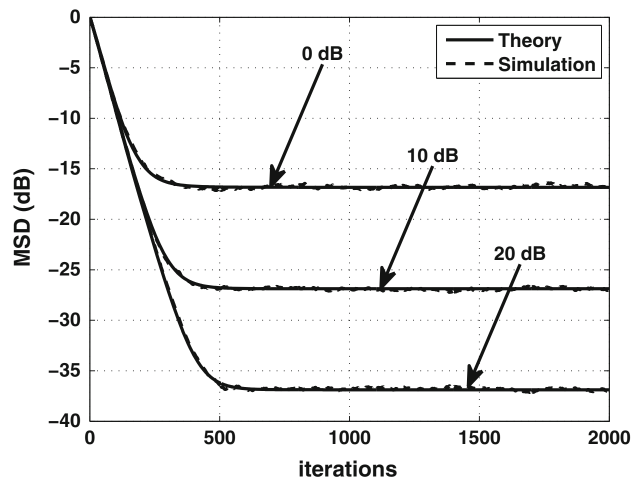

Fig. 2 Theory (21) v simulation MSD comparison for the NC algorithm [10]

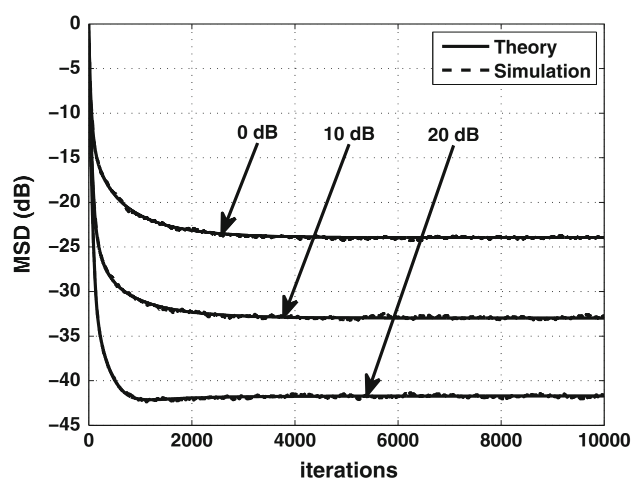

Fig. 3 Theory (21) v simulation MSD comparison for the VSQ algorithm [18]

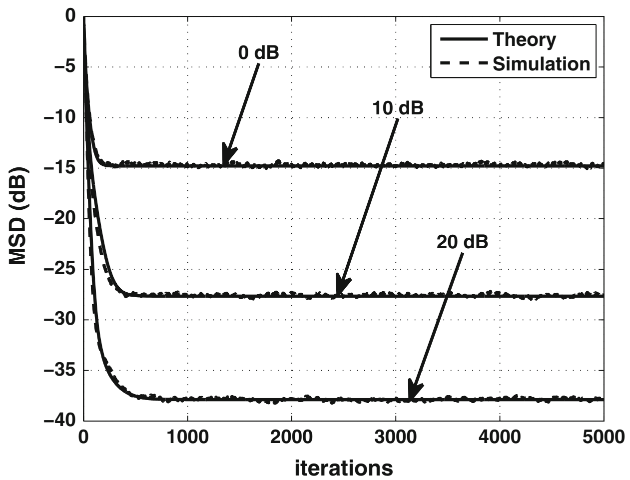

Fig. 4 Theory (21) v simulation MSD comparison for the Sp algorithm [22]

Table 5 Theory v simulation comparison for steady-state MSD for the different VSSLMS algorithms

Algorithm	SNR (dB)	MSD (dB) Eq. (27)	MSD (dB) sim.
	0	−13.76	−13.74
KJ [4]	10	−31.72	−31.55
	20	−46.98	−46.98
	0	−16.88	−16.84
NC [10]	10	−26.88	−26.93
	20	−36.88	−36.85
	0	−23.96	−24.00
VSQ [18]	10	−32.98	−33.01
	20	−41.72	−41.72
	0	−14.78	−14.64
Sp [22]	10	−27.65	−27.53
	20	−37.88	−37.84

remaining algorithms show an almost exact match during the transient state as well as at steady state.

The results for the second experiment are given in Table 5. It can be seen that there is an excellent match between theory and simulation results.

5 Conclusion

This work presents a unified approach for the theoretical analysis of LMS-based VSS algorithms. The iterative recursions presented here differentiate this work from previous analyses in that this set of equations provides a generic treatment of the analysis for this class of algorithms for the first time. This work provides an excellent tool for the analysis of any future VSSLMS algorithm. Several algorithms have been tested thoroughly to verify the results of this work under different SNR conditions. Simulation results confirm the generic behavior of the presented work, for both the transient state as well as steady state.

References

1. Sayed, A.H.: Fundamentals of Adaptive Filtering. Wiley, New York (2003)
2. Nagumo, J.; Noda, A.: A learning method for system identification. IEEE Trans. Autom. Control. **12**, 282–287 (1967)
3. Harris, R.; Chabries, D.M.; Bishop, F.: A variable step (VS) adaptive filter algorithm. IEEE Trans. Acoust. Speech Signal Process. **34**, 309–316 (1986)
4. Kwong, R.H.; Johnston, E.W.: A variable step-size LMS algorithm. IEEE Trans. Signal Process. **40**, 1633–1642 (1992)
5. Mathews, V.J.; Xie, Z.: A stochastic gradient adaptive filter with gradient adaptive step size. IEEE Trans. Signal Process. **41**, 2075–2087 (1993)
6. Aboulnasr, T.; Mayyas, K.: A robust variable step size LMS-type algorithm: analysis and simulations. IEEE Trans. Signal Process. **45**, 631–639 (1997)
7. Gollamudi, S.; Nagaraj, S.; Kapoor, S.; Huang, Y.-F.: Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size. IEEE Signal Process. Lett. **5**, 111–114 (1998)
8. Pazaitis, D.I.; Constantinides, A.G.: A novel kurtosis driven variable step-size adaptive algorithm. IEEE Trans. Signal Process. **47**, 864–872 (1999)
9. Ang, W.-P.; Farhang-Boroujeny, B.: A new class of gradient adaptive step-size LMS algorithms. IEEE Trans. Signal Process. **49**, 805–810 (2001)
10. Wei, Y.; Gelfand, S.B.; Krogmeier, J.V.: Noise-constrained least mean squares algorithm. IEEE Trans. Signal Process. **49**, 1961–1970 (2001)
11. Sulyman, A.I.; Zerguine, A.: Convergence and steady-state analysis of a variable step-size NLMS algorithm. Signal Process. **83**, 1255–1273 (2003)
12. Egiazarian, K.; Kuosmanen, P.; Bilcu, R.C.: Variable step-size LMS adaptive filters for CDMA multiuser detection. In: Proceedings of TELSIKS ’03, pp. 259–264 (2003)
13. Shin, H.-C.; Sayed, A.H.; Song, W.-J.: Variable step-size NLMS and affine projection algorithms. IEEE Signal Process. Lett. **11**, 132–135 (2004)
14. Benesty, J.; Rey, H.; Vega, L.R.; Tressens, S.: A nonparametric VSS NLMS algorithm. IEEE Signal Process. Lett. **13**, 581–584 (2006)
15. Akhtar, M.T.; Abe, M.; Kawamata, M.: A new variable step size LMS algorithm-based method for improved online secondary path modeling in active noise control systems. IEEE Trans. Audio Speech Lang. Process. **14**, 720–726 (2006)
16. Zhang, Y.; Li, N.; Chambers, J.A.; Hao, Y.: New gradient-based variable step size LMS algorithms. EURASIP J. Adv. Signal Process. **2008**, 1–9 (2008)
17. Costa, M.H.; Bermudez, J.C.M.: A noise resilient variable step-size LMS algorithm. Signal Process. **88**, 733–748 (2008)
18. Zhao, S.; Man, Z.; Khoo, S.; Wu, H.R.: Variable step-size LMS algorithm with a quotient form. Signal Process. **89**, 67–76 (2009)
19. Hwang, J.-K.; Li, Y.-P.: Variable step-size LMS algorithm with a gradient-based weighted average. IEEE Signal Process. Lett. **16**, 1043–1046 (2009)
20. Filho, A.M.A.; Pinto, E.L.; Galdino, J.F.: Simple and robust analytically derived variable step-size least mean squares algorithm for channel estimation. IET Commun. **3**, 1832–1842 (2009)

21. Narasimhan, S.V.: New unbiased adaptive IIR filter: its robust and variable step-size versions and application to active noise control. Signal Image Video Process. **7**, 197–207 (2013)
22. Bin Saeed, M.O.; Zerguine, A.: A variable step-size strategy for sparse system identification. In: Proceedings of SSD '13, pp. 1–4 (2013).
23. Bin Saeed, M.O.; Zerguine, A.; Zummo, S.A.: A variable step-size strategy for distributed estimation over adaptive networks. EURASIP J. Adv. Signal Process. **2013**, 135 (2013)
24. Bin Saeed, M.O.; Zerguine, A.; Zummo, S.A.: A noise-constrained algorithm for estimation over distributed networks. Int. J. Adapt. Control Signal Process. **27**, 827–845 (2013)
25. Gil-Cacho, J.M.; Waterschoot, Tv; Moonen, M.; Jensen, S.H.: Wiener variable step size and gradient spectral variance smoothing for double-talk-robust acoustic echo cancellation and acoustic feedback cancellation. Signal Process. **104**, 1–14 (2014)
26. Al-Naffouri, T.Y.; Moinuddin, M.: Exact performance analysis of the ϵ-NLMS algorithm for colored circular gaussian inputs. IEEE Trans. Signal Process. **58**, 5080–5090 (2010)
27. Al-Naffouri, T.Y.; Moinuddin, M.; Sohail, M.S.: Mean weight behavior of the NLMS algorithm for correlated gaussian inputs. IEEE Signal Process. Lett. **18**, 7–10 (2011)