

RESEARCH

Open Access

A variable step-size diffusion LMS algorithm with a quotient form



Muhammad Omer Bin Saeed¹ and Azzedine Zerguine^{2*}

Abstract

A new variable step-size strategy for the least mean square (LMS) algorithm is presented for distributed estimation in adaptive networks using the diffusion scheme. This approach utilizes the ratio of filtered and windowed versions of the squared instantaneous error for iteratively updating the step-size. The result is that the dependence of the update on the power of the error is reduced. The performance of the algorithm improves even though it is at the cost of added computational complexity. However, the increase in computational complexity can be minimized by careful manipulation of the update equation, resulting in an excellent performance-complexity trade-off. Complete theoretical analysis is presented for the proposed algorithm including stability, transient and steady-state analyses. Extensive experimental analysis is then done to show the performance of the proposed algorithm under various scenarios.

Keywords: Variable step-size, Least-mean-square algorithm, Adaptive networks, Mean-square analysis, Steady-state analysis

1 Introduction

The past decade has seen a lot of research in the area of distributed estimation for wireless sensor networks [1–4]. The diffusion scheme using the least mean square (LMS) algorithm was first introduced by Lopes and Sayed in [2]. The first algorithm to propose a variable step-size for the diffusion LMS algorithm was proposed by Saeed et al. in [5] and extended by Saeed et al. in [6]. Since then, several variable step-size algorithms have been proposed [7–12].

While each algorithm improves in performance, the improvement comes at the cost of increased computational complexity. The variable step-size strategy proposed by Kwong and Johnston in [13] was extended to the distributed network scenario by Saeed et al. in [5, 6]. This work was further expanded by Saeed and Zerguine in [7] by using the distributed nature of the network to improve performance. A noise constraint was introduced by Saeed et al. in [8] for improved performance. An optimal step-size for the distributed scenario was derived by

Ghazanfari-Rad and Labeau in [9], based on which they derived a variable step-size for each iteration. A variable step-size as well as combination weights were derived using an upper bound for the mean square deviation (MSD) by Jung et al. in [10]. Lee et al. derived an individual step-size for each sensor by minimizing the local MSD in [11]. Saeed et al. proposed a variable step-size strategy for distributed estimation in a compressible system in [12].

While all these algorithms provide excellent performance, they are all based on the l_2 -norm, except for the algorithm in [12]. This means that the power of the instantaneous error is the key factor for all these algorithms. This means that the performance of these algorithms suffers significantly in low signal-to-noise ratio (SNR) scenarios. Furthermore, there is significant increase in computational complexity as a trade-off for improved performance.

This work proposes a method where the step-size is varied using a quotient of the filtered error power. The algorithm is inspired from the work of Zhao et al. in [14]. The quotient form combined with the windowing effect reduces the effect of the power of the instantaneous error, making the algorithm more robust to low SNR scenarios.

*Correspondence: azzedine@kfupm.edu.sa

Azzedine Zerguine is a EURASIP member

²Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Full list of author information is available at the end of the article

The quotient form of [14] is combined with the diffusion scheme in this work. A theoretical analysis of the algorithm has been presented. The energy conservation approach used for the analysis in this work was not used in [14]. Saeed presented a generalized analysis for variable step-size schemes using the energy conservation approach in [15]. This work was extended to the diffusion scheme by Saeed et al. in [16]. However, the variable step-size using the quotient form was not included in [16]. The analysis presented here utilizes the approach of [16] and gives stability, transient, and steady-state analyses for the proposed algorithm. Simulation comparisons with the variable step-size diffusion LMS (VSSDLMS) algorithm of [6] show that the proposed algorithm achieves significant improvement with a reasonably acceptable increase in computational complexity.

The rest of the paper is divided as follows. Section 2 presents the system model and defines the problem statement. Section 3 proposes the new algorithm. The theoretical analysis is given in Section 4, followed by simulation results in Section 5. Section 6 concludes this work.

2 System model

A geographical area of N sensor nodes, each connected to its closest neighbors as shown in Fig. 1, is being considered. The unknown parameters are modeled as a vector, \mathbf{w}_o , of size $(M \times 1)$. The input to a node at any given time instant, i , is a $(1 \times M)$ regressor vector, $\mathbf{u}_k(i)$, where $k \in N$ is the node index. The resulting observed output for the node is a noise corrupted scalar, $d_k(i)$, given by

$$d_k(i) = \mathbf{u}_k(i)\mathbf{w}^o + v_k(i), \quad (1)$$

where $v_k(i)$ is the zero-mean additive noise.

The generic form of the *Adapt-then-Combine* (ATC) variable step-size diffusion LMS (VSDLMS) algorithm is given by [16]

$$\Psi_k(i+1) = \mathbf{w}_k(i) + \mu_k(i)e_k(i)\mathbf{u}_k^T(i), \quad (2)$$

$$\mathbf{w}_k(i+1) = \sum_{l \in \mathcal{N}_k} c_{lk} \Psi_l(i+1), \quad (3)$$

$$\mu_k(i+1) = f\{\mu_k(i)\}, \quad (4)$$

where $\mathbf{w}_k(i)$ is the estimate of the unknown vector at time instant i for node k , $b_f \Psi_k(i)$ is the intermediate update for node k , $e_k(i) = d_k(i) - \mathbf{u}_k(i)\mathbf{w}_k(i)$ is the instantaneous error, c_{lk} is the combination weight for the data transmitted from node l to node k , \mathcal{N}_k is the neighborhood of node k and $(\cdot)^T$ is the transpose operator. The step-size is denoted by $\mu_k(i)$ and $f\{\cdot\}$ is the function that updates the step-size at every iteration.

The main objective of this work is to propose a new function, $f\{\cdot\}$, for the VSDLMS algorithm defined by (2)–(4). The new function is derived in the next section, followed by theoretical analysis and experimental results.

3 Proposed algorithm

The proposed update equation for the step-size is given by

$$\mu_k(i+1) = \alpha \mu_k(i) + \gamma \theta_k(i), \quad (5)$$

$$\theta_k(i) = \frac{\sum_{m=0}^i a^m e_k^2(i-m)}{\sum_{n=0}^i b^n e_k^2(i-n)}, \quad (6)$$

where α and γ are positive parameters such that $0 < \alpha < 1$ and $\gamma > 0$ and a and b are forgetting factors for the decaying exponential windows in the numerator and denominator and are bounded as $0 < a < b < 1$. Although (6) represents the complete window for both the numerator and denominator, a more suitable and iterative representation is given by

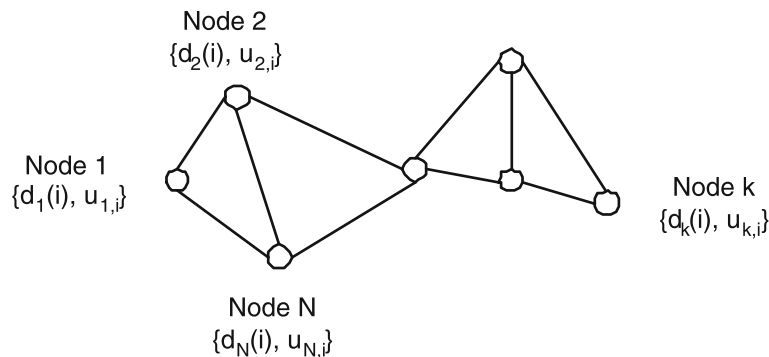


Fig. 1 An illustration of an adaptive network of N nodes

$$A_k(i) = \sum_{m=0}^i a^m e_k^2(i-m) = aA_k(i-1) + e_k^2(i), \quad (7)$$

$$B_k(i) = \sum_{n=0}^i b^n e_k^2(i-n) = bB_k(i-1) + e_k^2(i), \quad (8)$$

$$\theta_k(i) = \frac{A_k(i)}{B_k(i)} = \frac{aA_k(i-1) + e_k^2(i)}{bB_k(i-1) + e_k^2(i)}. \quad (9)$$

Combining (5) and (9) with (2) and (3) gives the proposed variable step-size diffusion LMS algorithm with a quotient form (VSQDLMS).

4 Theoretical analysis

Recently, the authors of [16] proposed a unified theoretical analysis for VSDLMS algorithms. Following the same procedure as [16], the following global variables are introduced to represent the entire network:

$$\begin{aligned} \mathbf{w}(i) &= \text{col}\{\mathbf{w}_1(i), \dots, \mathbf{w}_N(i)\}, \\ \Psi(i) &= \text{col}\{\Psi_1(i), \dots, \Psi_N(i)\}, \\ \mathbf{U}(i) &= \text{diag}\{\mathbf{u}_1(i), \dots, \mathbf{u}_N(i)\}, \\ \mathbf{D}(i) &= \text{diag}\{\mu_1(i)\mathbf{I}_M, \dots, \mu_N(i)\mathbf{I}_M\}, \\ \Theta(i) &= \text{diag}\{\theta_1(i)\mathbf{I}_M, \dots, \theta_N(i)\mathbf{I}_M\}, \\ \bar{\mathbf{A}}(i) &= \text{diag}\{A_1(i)\mathbf{I}_M, \dots, A_N(i)\mathbf{I}_M\}, \\ \bar{\mathbf{B}}(i) &= \text{diag}\{B_1(i)\mathbf{I}_M, \dots, B_N(i)\mathbf{I}_M\}, \\ \mathbf{d}(i) &= \text{col}\{d_1(i), \dots, d_N(i)\}, \\ \mathbf{v}(i) &= \text{col}\{v_1(i), \dots, v_N(i)\}. \end{aligned}$$

The combination weight matrix is given by $\{\mathbf{C}\}_{lk} = c_{lk}$. When expanded to the entire network, this becomes $\mathbf{G} = \mathbf{C} \otimes \mathbf{I}_M$, where \otimes represents the Kronecker product. The unknown parameters vector \mathbf{w}^o is expanded for the entire network using $\mathbf{w}^{(o)} = \mathbf{Q}\mathbf{w}^o$, where $\mathbf{Q} = \text{col}\{\mathbf{I}_M, \mathbf{I}_M, \dots, \mathbf{I}_M\}$ is a $(MN \times M)$ sized matrix. The resulting global set of equations representing the VSQDLMS algorithm are given by

$$\mathbf{d}(i) = \mathbf{U}(i)\mathbf{w}^{(o)} + \mathbf{v}(i), \quad (10)$$

$$\Psi(i+1) = \mathbf{w}(i) + \mathbf{D}(i)\mathbf{U}^T(i)(\mathbf{d}(i) - \mathbf{U}(i)\mathbf{w}(i)), \quad (11)$$

$$\mathbf{w}(i+1) = \mathbf{G}\Psi(i+1), \quad (12)$$

$$\Theta(i) = \frac{\bar{\mathbf{A}}(i)}{\bar{\mathbf{B}}(i)}, \quad (13)$$

$$\mathbf{D}(i+1) = \alpha\mathbf{D}(i) + \gamma\Theta(i). \quad (14)$$

4.1 Mean analysis

Introducing the global weight-error vector as

$$\tilde{\mathbf{w}}(i) = \mathbf{w}^{(o)} - \mathbf{w}(i). \quad (15)$$

Using (15), the update in (2) is rewritten as

$$\tilde{\Psi}_k(i+1) = \tilde{\mathbf{w}}_k(i) - \mu_k(i)e_k(i)\mathbf{u}_k^T(i), \quad (16)$$

where the relation in (15) is extended for $\Psi_k(i)$. Expanding the error term, $e_k(i)$, and rearranging gives

$$\begin{aligned} \tilde{\Psi}_k(i+1) &= \left[\mathbf{I}_M - \mu_k(i)\mathbf{u}_k^T(i)\mathbf{u}_k(i) \right] \tilde{\mathbf{w}}_k(i) \\ &\quad - \mu_k(i)\mathbf{u}_k^T(i)v_k(i). \end{aligned} \quad (17)$$

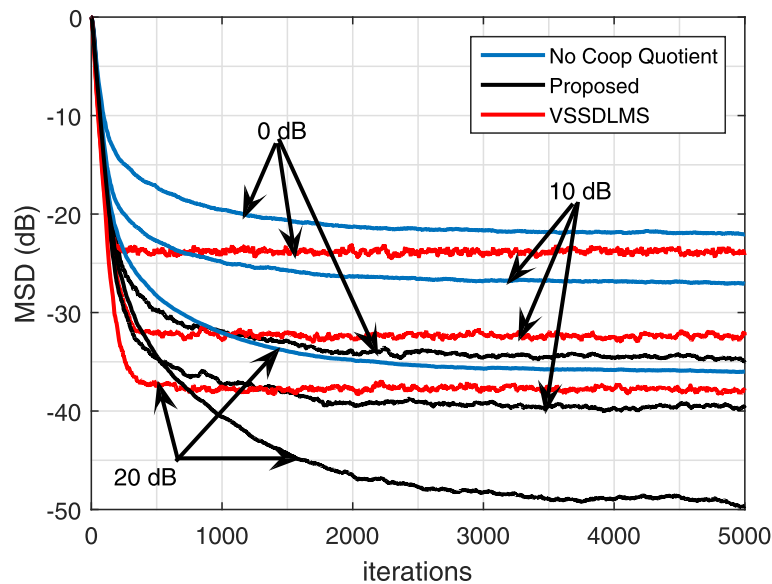


Fig. 2 Performance comparison for the proposed VSQDLMS vs VSSDLMS [6] algorithms for white input

Applying the expectation operator to (17) and using the independence assumption gives, after simplification

$$\mathbb{E} [\tilde{\Psi}_k(i+1)] = [\mathbf{I}_M - \mathbb{E} [\mu_k(i)] \mathbf{R}_{\mathbf{u}_k}] \mathbb{E} [\tilde{\mathbf{w}}_k(i)], \quad (18)$$

where $\mathbf{R}_{\mathbf{u}_k} = \mathbb{E} [\mathbf{u}_k^T(i) \mathbf{u}_k(i)]$ is the autocorrelation of the input data at node k .

The algorithm will be stable in the mean sense if the term $[\mathbf{I}_M - \mathbb{E} [\mu_k(i)] \mathbf{R}_{\mathbf{u}_k}]$ is bounded. This means that the algorithm will not diverge if this term is bounded. Thus, for the algorithm to be stable in the mean, the following condition needs to be satisfied:

$$0 < \mathbb{E} [\mu_k(i)] < \frac{2}{\lambda_{\max}(\mathbf{R}_{\mathbf{u}_k})}, \quad 1 \leq k \leq N, \quad (19)$$

where $\lambda_{\max}(\mathbf{R}_{\mathbf{u}_k})$ is the maximum eigenvalue of the auto-correlation matrix $\mathbf{R}_{\mathbf{u}_k}$.

4.2 Mean-square analysis

The mean-square analysis in [16] assumes Gaussian input data. The auto-correlation matrix is decomposed into its component eigenvector and eigenvalue matrices as $\mathbf{R}_{\mathbf{U}} = \mathbf{T} \mathbf{\Lambda} \mathbf{T}^T$, with $\mathbf{\Lambda}$ being the diagonal eigenvalue matrix and \mathbf{T} being the eigenvector matrix, such that $\mathbf{T}^T \mathbf{T} = \mathbf{I}$. The eigenvector matrix \mathbf{T} is then used to transform the remaining variables as follows:

$$\begin{aligned} \bar{\mathbf{w}}(i) &= \mathbf{T}^T \tilde{\mathbf{w}}(i) & \bar{\mathbf{U}}(i) &= \mathbf{U}(i) \mathbf{T} \\ \bar{\mathbf{G}} &= \mathbf{T}^T \mathbf{G} \mathbf{T} & \bar{\Sigma} &= \mathbf{T}^T \Sigma \mathbf{T} \\ \bar{\Sigma}' &= \mathbf{T}^T \Sigma' \mathbf{T} & \bar{\mathbf{D}}(i) &= \mathbf{T}^T \mathbf{D}(i) \mathbf{T} = \mathbf{D}(i), \end{aligned}$$

where Σ and Σ' are weighting matrices. The final iterative update equation for the mean-square transient analysis is given by [16]

$$\begin{aligned} \mathbb{E} [\|\bar{\mathbf{w}}(i+1)\|_{\bar{\sigma}}^2] &= \mathbb{E} [\|\bar{\mathbf{w}}(i)\|_{\bar{\sigma}}^2] + \mathbf{b}^T(i) \bar{\sigma} \\ &\quad + \|\bar{\mathbf{w}}^{(o)}\|_{\mathbf{A}(i)[\mathbf{F}(i) - \mathbf{I}_{M^2 N^2}] \bar{\sigma}}^2 \\ &\quad + \mathcal{B}(i) [\mathbf{F}(i) - \mathbf{I}_{M^2 N^2}] \bar{\sigma}, \end{aligned} \quad (20)$$

where

$$\mathbf{b}(i) = \text{bvec} \{ \mathbf{R}_{\mathbf{v}} \mathbb{E} [\mathbf{D}^2(i)] \mathbf{\Lambda} \}, \quad (21)$$

$$\begin{aligned} \mathbf{F}(i) &= [\mathbf{I}_{M^2 N^2} - (\mathbf{I}_{MN} \odot \mathbf{\Lambda} \mathbb{E} [\mathbf{D}(i)]) \\ &\quad - (\mathbf{\Lambda} \mathbb{E} [\mathbf{D}(i)] \odot \mathbf{I}_{MN}) \\ &\quad + (\mathbb{E} [\mathbf{D}(i) \odot \mathbf{D}(i)]) \mathbf{A}] \cdot (\mathbf{G}^T \odot \mathbf{G}^T), \end{aligned} \quad (22)$$

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

$$\mathcal{B}(i+1) = \mathcal{B}(i) \mathbf{F}(i) + \mathbf{b}^T(i) \mathbf{I}_{M^2 N^2}, \quad (24)$$

Table 1 Computational complexity comparison for step-size update equations

Algorithm	Multiplications	Additions
VSSDLMS [6]	3	1
Proposed	6	3

Table 2 Parameter values for Experiment 1

SNR	Algorithm	Parameter	Value		
0	VSSDLMS [6]	α_{white}	0.95		
		$\alpha_{\text{correlated}}$	0.95		
		γ	10^{-3}		
		Proposed	α_{white}	0.96	
			$\alpha_{\text{correlated}}$	0.961	
			γ	10^{-3}	
	a		0.99		
	b		$1 - 10^{-3}$		
	VSSDLMS [6]		α_{white}	0.97	
		$\alpha_{\text{correlated}}$	0.99		
		γ	10^{-3}		
		10	Proposed	α_{white}	0.96
$\alpha_{\text{correlated}}$				0.9675	
γ				10^{-3}	
a	0.99				
b	$1 - 10^{-3}$				
VSSDLMS [6]	α_{white}			0.99	
	$\alpha_{\text{correlated}}$		0.99		
	γ		10^{-3}		
	20		Proposed	α_{white}	0.99
				$\alpha_{\text{correlated}}$	0.9725
				γ	10^{-3}
a				0.9	
b		$1 - 10^{-3}$			
VSSDLMS [6]		α_{white}		0.99	
		$\alpha_{\text{correlated}}$	0.99		
		γ	10^{-3}		

where \odot denotes the block Kronecker product, $\mathbf{R}_{\mathbf{v}} = \mathbf{\Lambda}_{\mathbf{v}} \odot \mathbf{I}_M$, $\mathbf{\Lambda}_{\mathbf{v}}$ is a diagonal noise variance matrix for the network and $\bar{\sigma} = \text{bvec} \{ \bar{\Sigma} \}$. The matrix \mathbf{A} is given in [2, 16] as $\mathbf{A} = \text{diag} \{ \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N \}$, with each component matrix defined as

$$\mathbf{A}_k = \text{diag} \left\{ \mathbf{\Lambda}_1 \otimes \mathbf{\Lambda}_k, \dots, \lambda_k \lambda_k^T + 2 \mathbf{\Lambda}_k \otimes \mathbf{\Lambda}_k, \dots, \mathbf{\Lambda}_N \otimes \mathbf{\Lambda}_k \right\}, \quad (25)$$

where $\mathbf{\Lambda}_k$ is the diagonal eigenvalue matrix and λ_k is the corresponding eigenvalue vector for node k .

In order to find the mean square deviation (MSD), the weighting matrix is chosen as $\bar{\Sigma} = \mathbf{I}_{N^2 M^2}$. To get the excess mean square error (EMSE), the weighting matrix is chosen as $\bar{\Sigma} = \mathbf{\Lambda}$.

4.3 Steady-state analysis

At steady-state, (20) becomes

$$\mathbb{E} [\|\bar{\mathbf{w}}_{ss}\|_{\bar{\sigma}}^2] = \mathbb{E} [\|\bar{\mathbf{w}}_{ss}\|_{\bar{\mathbf{F}}_{ss} \bar{\sigma}}^2] + \mathbf{b}_{ss}^T \bar{\sigma}, \quad (26)$$

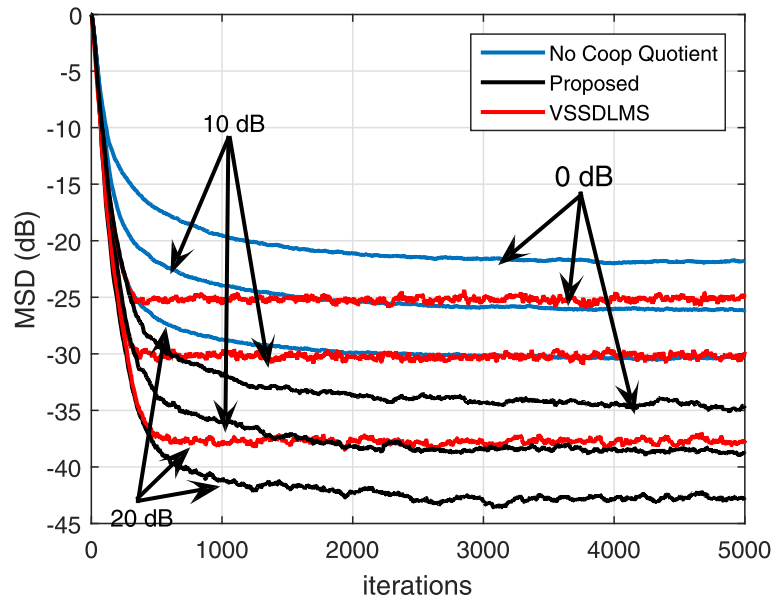


Fig. 3 Performance comparison for the proposed VSQDLMS vs VSSDLMS [6] algorithms for correlated input

where

$$\mathbf{F}_{ss} = [\mathbf{I}_{M^2N^2} - (\mathbf{I}_{MN} \odot \Lambda \mathbf{E}[\mathbf{D}_{ss}]) - (\Lambda \mathbf{E}[\mathbf{D}_{ss}] \odot \mathbf{I}_{MN}) + (\mathbf{E}[\mathbf{D}_{ss} \odot \mathbf{D}_{ss}]) \mathbf{A}] \cdot (\mathbf{G}^T \odot \mathbf{G}^T), \quad (27)$$

$$\mathbf{b}_{ss} = \mathbf{R}_v \mathbf{D}_{ss}^2 \mathbf{A}. \quad (28)$$

To get the steady-state MSD, we choose $\bar{\sigma} = \text{bvec}\{\mathbf{I}_{N^2M^2}\}$, and to get the steady-state EMSE, we choose $\bar{\sigma} = \text{bvec}\{\mathbf{A}\}$.

4.4 Steady-state step-size analysis

The step-size matrices, $\mathbf{D}_{ss} = \text{diag}\{\mu_{k,ss} \mathbf{I}_M\}$ and $\mathbf{D}_{ss}^2 = \text{diag}\{\mu_{k,ss}^2 \mathbf{I}_M\}$, are defined by the individual node values for the steady-state step-size. The step-size update for the proposed algorithm is defined by (5), (7)-(9). Taking the expectation gives

$$\begin{aligned} \mathbf{E}[A_k(i)] &= a \mathbf{E}[A_k(i-1)] + \mathbf{E}[e_k^2(i)] \\ &= a \mathbf{E}[A_k(i-1)] + \text{EMSE}(i) + \sigma_{v,k}^2, \end{aligned} \quad (29)$$

$$\mathbf{E}[B_k(i)] = b \mathbf{E}[B_k(i-1)] + \text{EMSE}(i) + \sigma_{v,k}^2, \quad (30)$$

$$\mathbf{E}[\theta_k(i)] \approx \frac{\mathbf{E}[A_k(i)]}{\mathbf{E}[B_k(i)]}, \quad (31)$$

$$\mathbf{E}[\mu_k(i+1)] = \alpha \mathbf{E}[\mu_k(i)] + \gamma \mathbf{E}[\theta_k(i)]. \quad (32)$$

At steady-state, the EMSE value is assumed to go to 0, the expectation operator is removed and after simple manipulations, we get

$$A_{k,ss} = \frac{1}{1-a} \sigma_{v,k}^2, \quad (33)$$

$$B_{k,ss} = \frac{1}{1-b} \sigma_{v,k}^2, \quad (34)$$

$$\theta_{k,ss} = \frac{A_{k,ss}}{B_{k,ss}} = \frac{1-b}{1-a}, \quad (35)$$

$$\mu_{k,ss} = \frac{\gamma(1-b)}{(1-\alpha)(1-a)}, \quad (36)$$

where the term *ss* indicates steady-state value. Thus, the steady-state step-size value is given by (36) and the value for $\mu_{k,ss}^2$ is simply the square of (36).

5 Results and discussion

This section presents experimental results for the proposed algorithm. The algorithm is tested in four different scenarios. First, the proposed algorithm is compared with the algorithm from [6] as well with the no cooperation case, in which each sensor applies the variable step-size quotient strategy for estimation without cooperating with any other sensor. The reason for the comparison with only

Table 3 Steady-state results for Figs. 1 and 2

SNR	No coop	VSSDLMS [6]	Proposed
White input			
0	-21.85	-23.73	-34.38
10	-26.95	-32.22	-39.39
20	-35.96	-37.78	-49.05
Correlated input			
0	-21.79	-24.77	-34.22
10	-26.04	-30.21	-38.72
20	-30.38	-37.85	-43.04

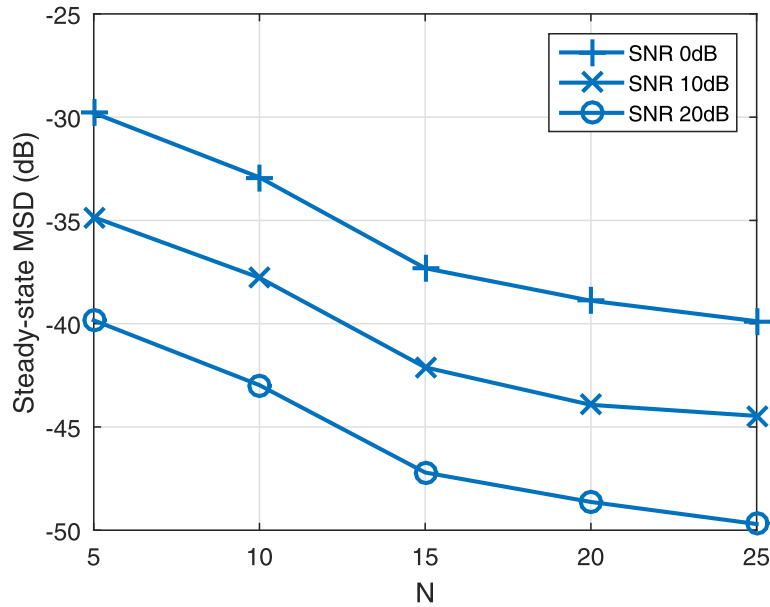


Fig. 4 Steady-state results for varying N with $M = 5$ and SNR 20 dB

the algorithm of [6] is that the computational complexity for both algorithms is similar. However, it is shown that with a small increase in computational complexity, the proposed algorithm performs much better. Next, the theoretical transient analysis of (20) is compared with simulation results. In the third experiment, the steady-state results from (26) are compared with steady-state simulation results. Finally, the performance of the proposed

algorithm is tested in a scenario where the exact value of M is unknown.

5.1 Experiment I

In the first experiment, the proposed VSQDLMS algorithm is compared with the variable step-size diffusion LMS (VSSDLMS) algorithm from [6]. A comparison of computational complexity is given in Table 1. As can

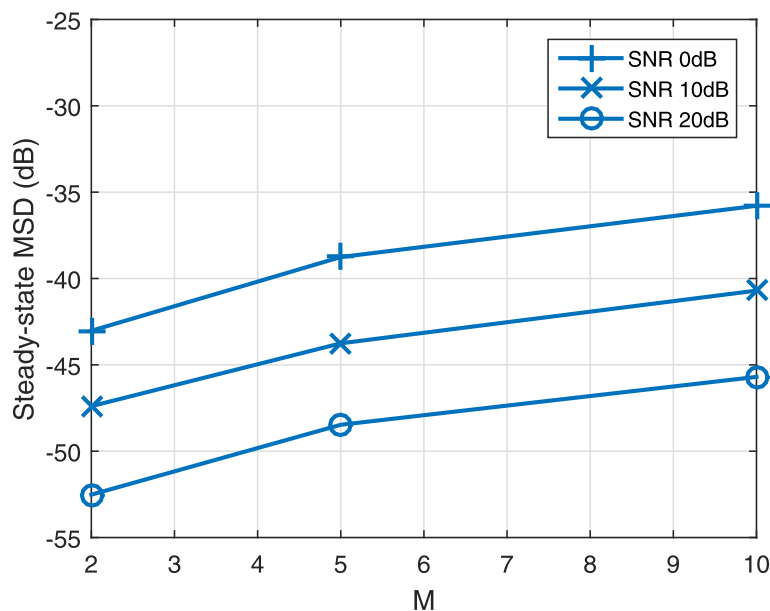


Fig. 5 Steady-state results for varying M with $N = 20$ and SNR 20 dB

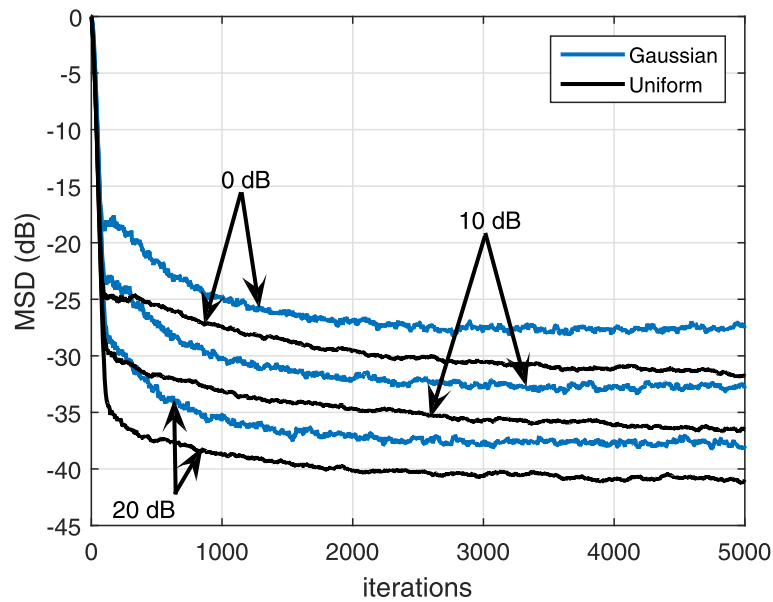


Fig. 6 Performance comparison for Gaussian and uniform input data

be seen, the increase in computations is small. The performance of the two algorithms is compared through simulations. The no cooperation case has been added to compare the performance of the proposed algorithm with the scenario where each sensor operates as if it were acting in a stand-alone environment. The size of \mathbf{w}^o is chosen as $M = 5$. The size of the network is $N = 20$. The SNR value is varied from 0 dB to 20 dB. The

parameters defining the step-size update equations are shown in Table 2. The values of the parameters are chosen such that the convergence speed is the same for all algorithms and the performance is measured on the basis of the steady-state MSD, as shown in the figures. Results are shown for white Gaussian input regressor data as well as colored input regressor data, with the correlation factor of 0.5. As can be seen from Figs. 2 and 3, the proposed

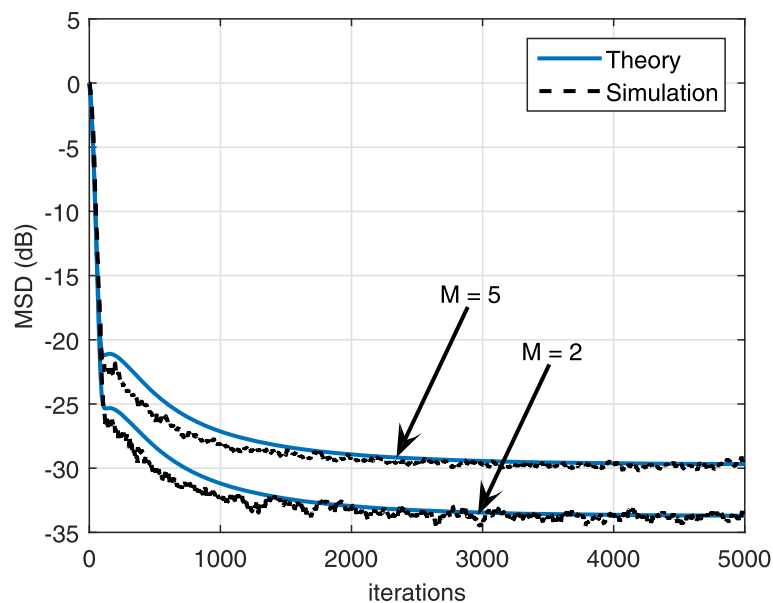


Fig. 7 Theory (20) vs simulation for a network of size $N = 5$

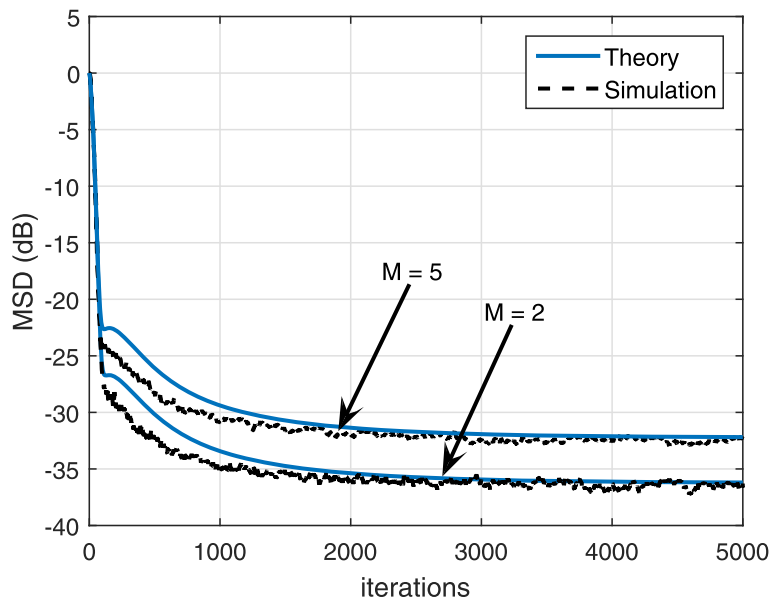


Fig. 8 Theory (20) vs simulation for a network of size $N = 10$

algorithm clearly outperforms the VSSDLMS algorithm from [6]. At low SNR, the performance of the proposed algorithm at 0 dB is even better than the performance of the VSSDLMS algorithm at 10 dB. The steady-state results for these experiments are given in Table 3. As can be seen, the proposed algorithm gives an improvement in performance of almost 10 dB for the case of white input and 5 dB for the case of colored input. This significant

improvement in performance clearly justifies the trade-off between complexity and performance.

5.2 Experiment II

In this experiment, the performance of the proposed algorithm is tested for different network sizes with a fixed unknown vector length, $M = 5$. The performance of the proposed algorithm is then tested for a fixed network size,

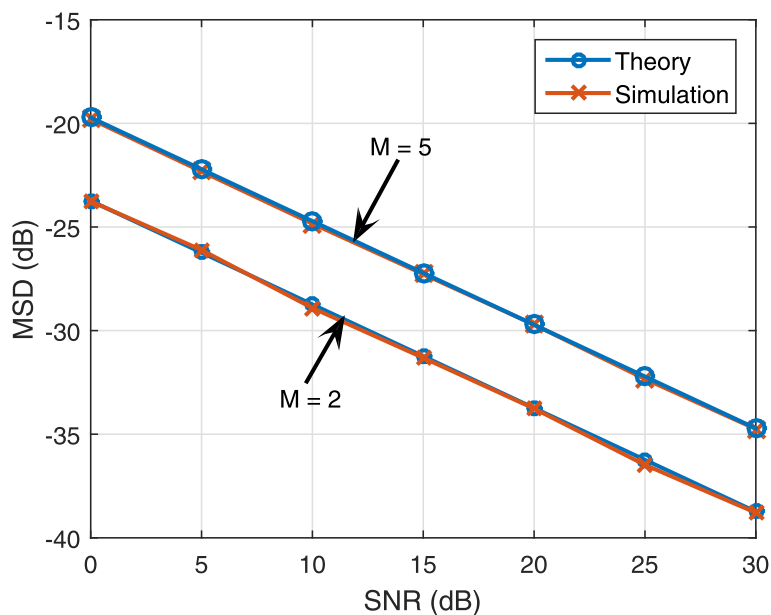


Fig. 9 Steady-state theory (26) vs simulation for a network of size $N = 5$

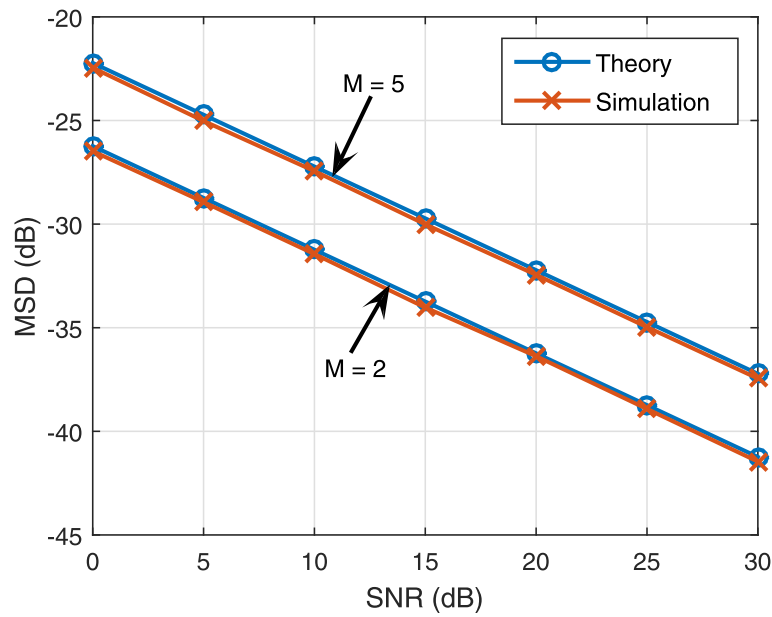


Fig. 10 Steady-state theory (26) vs simulation for a network of size $N = 10$

$N = 20$, while the length of the unknown vector, M is varied. Finally, the performance of the algorithm is compared for Gaussian and Uniform input data. The results are shown in Figs. 4, 5, and 6. For varying N and M , the steady-state MSD results are plotted. As expected, performance improves as N increases as each sensor has more

neighboring nodes for sharing data and improving its estimate. However, the performance degrades with increase in M as it becomes difficult to obtain an accurate estimate as the number of unknown parameters increases. The results shown use the same control parameters as in Experiment I.

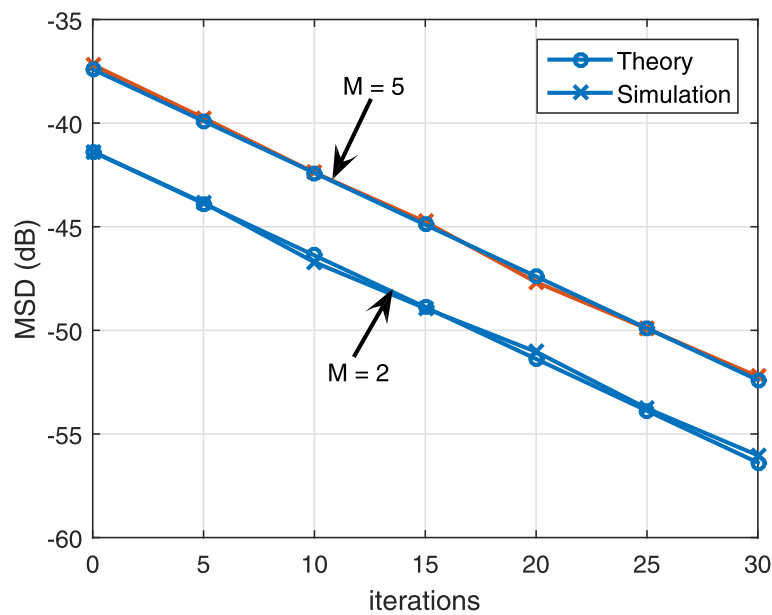


Fig. 11 Steady-state theory (26) vs simulation for a network of size $N = 15$

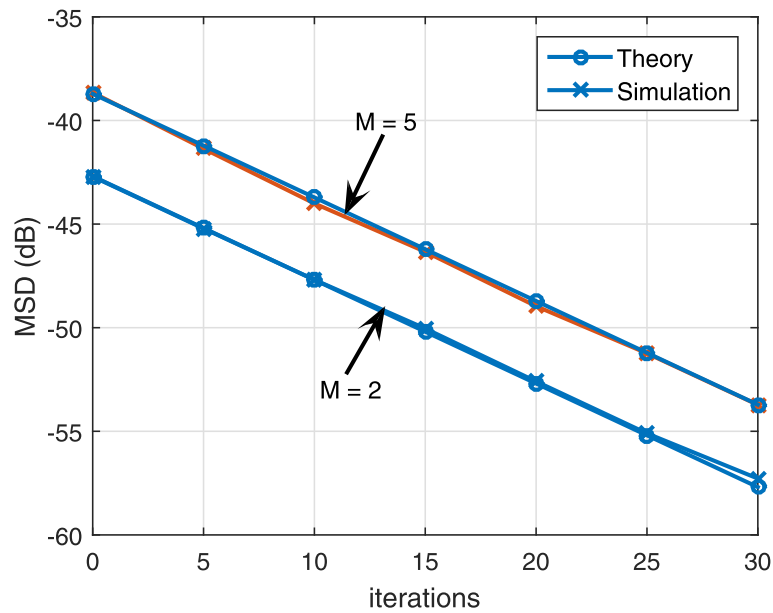


Fig. 12 Steady-state theory (26) vs simulation for a network of size $N = 20$

5.3 Experiment III

The next experiment compares the transient behavior of the proposed algorithm obtained from (20) with the simulation results. For this experiment, the network size is chosen as $N = 5$ and $N = 10$. The length of the unknown parameter vector is taken to be $M = 2$ and $M = 5$. The SNR is fixed at 20 dB. The step-size control parameters are set as $\alpha = 0.99$, $\gamma = 10^{-3}$ and $b = 1 - 10^{-3}$. The results are shown in Figs. 7 and 8. As can be seen, the theoretical results closely match the simulation results. The mismatch during the transient stage is a result of the assumptions from [16]. However, this is expected, as shown in the results of [16], and is acceptable as the two curves are closely matched at steady-state.

5.4 Experiment IV

Next, we compare the steady-state results obtained using (26) with the steady-state results obtained through

simulations. The SNR values are varied from 0 dB to 30 dB. The network size is varied from $N = 5$ to $N = 20$. The results are shown in Figs. 9, 10, 11, and 12. There is a very close match for all results. For an exact comparison, some of these results have been tabulated in Tables 4 and 5. As can be seen, the results for the theory match closely with those obtained through simulation.

5.5 Experiment V

In the last experiment, the robustness of the proposed algorithm is tested in the case that the exact number of unknown parameters are not known. This means that the exact value for M is unknown. In such a case, there are three possible scenarios. First, an underdetermined case, where the value of M is chosen as less than the actual value. Second, an overdetermined case, where the value of M is chosen as more than the actual value. Finally, the case where the value of M is chosen

Table 4 Steady-state results for $N = 5$ and $N = 10$

M	SNR	$N = 5$		$N = 10$	
		Eq. (26)	Sim.	Eq. (26)	Sim.
2	0	-23.74	-23.73	-26.24	-26.48
	10	-28.74	-28.83	-31.24	-31.44
	20	-33.74	-33.79	-36.24	-36.37
5	0	-19.72	-19.79	-22.22	-22.46
	10	-24.72	-24.84	-27.22	-27.44
	20	-29.72	-29.72	-32.22	-32.45

Table 5 Steady-state results for $N = 15$ and $N = 20$

M	SNR	$N = 15$		$N = 20$	
		Eq. (26)	Sim.	Eq. (26)	Sim.
2	0	-41.38	-41.37	-42.69	-42.71
	10	-46.38	-46.71	-47.69	-47.69
	20	-51.38	-51.04	-52.69	-52.59
5	0	-37.40	-37.19	-38.71	-38.66
	10	-42.40	-42.39	-43.71	-44.00
	20	-47.40	-47.68	-48.71	-48.96

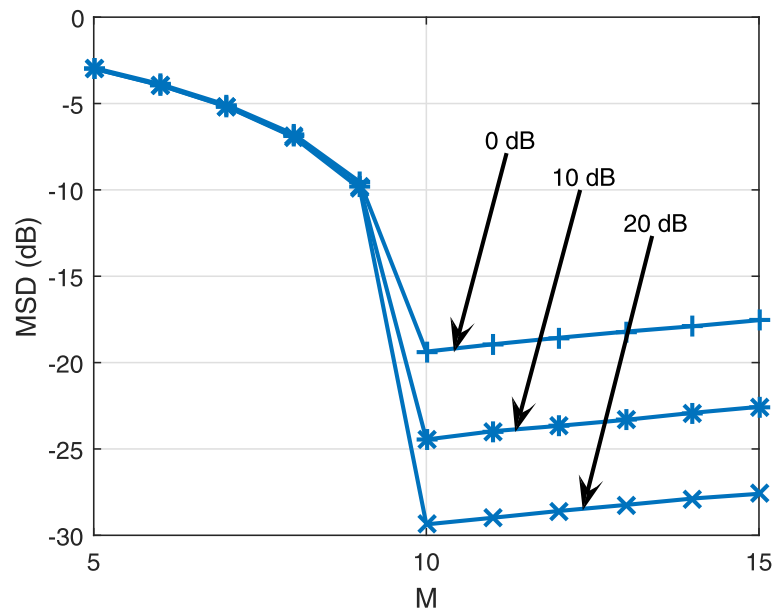


Fig. 13 Steady-state results for a mismatch in the value of M for a network of size $N = 10$

correctly. For this experiment, the network size is chosen as $N = 10$ and the unknown parameter vector length as $M = 10$. The range for M is varied between 5 and 15, and the steady-state results are compared for SNR values of 0 dB, 10 dB, and 20 dB. The results are shown in Fig. 13. It can be seen that for the underdetermined case, the value of SNR does not matter, as the algorithm converges very quickly to give a large error. The best result is achieved for an exact match, as expected. However, the results for the overdetermined case are very close to the best result. As the value of M increases, the value of the steady-state error also increases. However, it is clear from these results that in case of a mismatch, it is better to have an overdetermined system rather than an underdetermined system.

6 Conclusion

This work proposes a new variable step-size diffusion least mean square algorithm, based on the quotient form, for distributed estimation in adaptive networks. The mean, mean-square, and steady-state analyses have presented. The proposed algorithm has been compared with the algorithm from [6], and a computational complexity analysis has been shown to justify this comparison. It has been shown that the proposed algorithm easily outperforms the algorithm from [6] and provides a much better trade-off between complexity and performance. Results show an improvement in steady-state MSD of almost 10 dB for white input data and almost 5 dB for correlated input. Results are also shown for different network sizes as well as for different sizes of the unknown vector. A comparison between theoretical and

simulation results has been shown and the two were found to be closely matched, as shown by the figures as well as the tables. Finally, the performance of the algorithm has been tested in case of a mismatch in the value of M . The proposed algorithm has been shown to perform well in a Gaussian noise environment. However, for non Gaussian environments, a variable step-size algorithm based on the least mean fourth algorithm be developed.

Acknowledgments

The authors acknowledge the support provided by the Deanship of Scientific Research at KFUPM under Research Grant SB181016. Also, the authors would like to acknowledge the support provided by the National Plan for Science, Technology and Innovation (MAARIFAH)—King Abdulaziz City for Science and Technology through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM), the Kingdom of Saudi Arabia under award 12-ELE3006-04.

Authors' contributions

The authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Computer Engineering, College of E. & M. E., National University of Sciences & Technology, Islamabad, Pakistan. ²Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia.

Received: 4 March 2019 Accepted: 2 March 2020

Published online: 24 March 2020

References

1. A. H. Sayed, Adaptive Networks. *Proc. IEEE*. **102**(4), 460–497 (2014)
2. C. G. Lopes, A. H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Trans. Sig. Process.* **56**(7), 3122–3136 (2008)

3. I. D. Schizas, G. Mateos, G. B. Giannakis, Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. Sig. Process.* **57**(6), 2365–2382 (2009)
4. F. Cattivelli, A. H. Sayed, Diffusion LMS strategies for distributed estimation. *IEEE Trans. Sig. Process.* **58**(3), 1035–1048 (2010)
5. M. O. B. Saeed, A. Zerguine, S. A. Zummo, Variable step-size least mean square algorithms over adaptive networks, in *Proc. Of 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA)*, 381–384 (2010)
6. M. O. B. Saeed, A. Zerguine, S. A. Zummo, A variable step-size strategy for distributed estimation over adaptive networks. *EURASIP J. Adv. Sig. Process.* **2013**, 2013:135 (2013)
7. M. O. B. Saeed, A. Zerguine, A new variable step-size strategy for adaptive networks, in *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers*, 312–315 (2011)
8. M. O. B. Saeed, A. Zerguine, S. A. Zummo, A noise-constrained algorithm for estimation over distributed networks. *Intl. J. Adap. Cont. Sig. Process.* **27**(10), 827–845 (2013)
9. S. Ghazanfari-Rad, F. Labeau, Optimal variable step-size diffusion LMS algorithms, in *Proc. of IEEE Workshop on Statistical Signal Processing (SSP)*, 464–467 (2014)
10. S. M. Jung, J.-H. Sea, P. G. Park, A variable step-size diffusion normalized least-mean-square algorithm with a combination method based on mean-square deviation. *Circ. Syst. Sig. Process.* **34**(10), 3291–3304 (2015)
11. H. S. Lee, S. E. Kim, J. W. Lee, W. J. Song, A variable step-size diffusion LMS algorithm for distributed estimation. *IEEE Trans. Sig. Process.* **63**(7), 1808–1820 (2015)
12. M. O. B. Saeed, A. Zerguine, M. S. Sohail, S. Rehman, W. Ejaz, A. Anpalagan, A variable step-size strategy for distributed estimation of compressible systems in wireless sensor networks, in *Proc. of IEEE CAMAD*, 1–5 (2016)
13. R. H. Kwong, E. W. Johnston, A variable step-size LMS algorithm. *IEEE Trans. Sig. Process.* **40**(7), 1633–1642 (1992)
14. S. Zhao, Z. Man, S. Khoo, H. R. Wu, Variable step-size LMS algorithm with a quotient form. *Sig. Process.* **89**(1), 67–76 (2009)
15. M. O. B. Saeed, LMS-based variable step-size algorithms: a unified analysis approach. *Arab. J. Sci. Eng.* **42**(7), 2809–2816 (2017)
16. M. O. B. Saeed, W. Ejaz, S. Rehman, A. Zerguine, A. Anpalagan, H. Song, A unified analytical framework for distributed variable step size LMS algorithms in sensor networks. *Telecom. Syst.* (2018). <https://doi.org/10.1007/s11235-018-0447-z>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
