Telecommunication Systems (2018) 69:447–459
https://doi.org/10.1007/s11235-018-0447-z

A unified analytical framework for distributed variable step size LMS algorithms in sensor networks

Muhammad Omer Bin Saeed[1] · Waleed Ejaz[2] · Saad Rehman[1] · Azzedine Zerguine[3] · Alagan Anpalagan[2] · Houbing Song[4]

Published online: 28 March 2018

Abstract

Internet of Things (IoT) is helping to create a smart world by connecting sensors in a seamless fashion. With the forthcoming fifth generation (5G) wireless communication systems, IoT is becoming increasingly important since 5G will be an important enabler for the IoT. Sensor networks for IoT are increasingly used in diverse areas, e.g., in situational and location awareness, leading to proliferation of sensors at the edge of physical world. There exist several variable step-size strategies in literature to improve the performance of diffusion-based Least Mean Square (LMS) algorithm for estimation in wireless sensor networks. However, a major drawback is the complexity in the theoretical analysis of the resultant algorithms. Researchers use several assumptions to find closed-form analytical solutions. This work presents a unified analytical framework for distributed variable step-size LMS algorithms. This analysis is then extended to the case of diffusion based wireless sensor networks for estimating a compressible system and steady state analysis is carried out. The approach is applied to several variable step-size strategies for compressible systems. Theoretical and simulation results are presented and compared with the existing algorithms to show the superiority of proposed work.

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

1 Introduction

The Internet of Things (IoT) has grown immensely during the last decade with many emerging applications built around different types of sensors. With issues at device and protocol levels, there is a growing trend in integration of sensors and sensor based systems with cyber physical systems as well as device-to-device (D2D) communications [1–3]. Recently, research directions has been shifted towards 5G networks and one current active area of research in 5G is IoT [4]. One of the most important component of the IoT paradigm is wireless sensor networks (WSNs). WSNs can sense the environment, process the sensed data, and communicate for monitoring and diagnostics [6]. The sensor data originating from the future IoT is expected to be diverse and also grow manifold with each passing year. During the last few years, multi sensor data fusion attracted a lot of attention in

✉ Houbing Song
Houbing.Song@erau.edu

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

Waleed Ejaz
waleed.ejaz@ieee.org

Saad Rehman
saadrehman@ceme.nust.edu.pk

Azzedine Zerguine
azzedine@kfupm.edu.sa

Alagan Anpalagan
alagan@ee.ryerson.ca

[1] Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Rawalpindi, Pakistan

[2] Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada

[3] Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

[4] Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

many applications (such as radar and target detection, target tracking, smart grid, intelligent transportation systems, robot networks, smart cities, etc.) as they provide a cost-effective solution. Unknown parameters or phenomenon of interest can be estimated using centralized and distributed estimation systems. For centralized estimation systems, the sensor nodes collect data and transmit it to a central processor for signal processing. The central node is also responsible for communication with the individual nodes. The resulting estimate is as accurate as if each sensor had access to the complete information available across the network. However, this centralized system has several disadvantages including cost of communication overhead, power consumption, and single point of failure. In order to overcome these issues, the distributed estimation systems have been proposed in the literature. Each sensor node senses data and collaborates with its neighboring nodes in order to arrive at an estimate of the unknown parameter instead of transmitting the entire data to a central node. This approach offers less communication overhead and low processing load at the cost of added complexity at the nodes [7–11].

Various algorithms have been proposed in the literature for distributed estimation over WSNs [9–21]. Further, a detailed survey of distributed algorithms is available in [22]. The focus is primarily on least mean-square (LMS)-based algorithms due to simplicity and effectiveness [23]. However, an important limiting factor of LMS is the trade-off between convergence speed and steady-state miss-adjustment. One of the most popular and fully distributed algorithm is diffusion LMS [12]. To overcome the problem with LMS-based algorithms, several variable step-size (VSS) strategies have been proposed, which provide fast convergence speed initially while reducing the step-size with time in order to achieve a low error performance [24–30].

1.1 VSS strategies in literature

The work in [24] introduces a normalization factor to overcome the problem of slow convergence in the case high auto-correlation for the input data, thus making the step-size variable. A novel VSS algorithm is introduced in [25], which has become the most popular VSS algorithm in literature due to its low complexity and excellent performance. The algorithm uses the energy of the instantaneous error to update the step-size. However, the algorithm may suffer in performance if the noise power is high. To overcome this problem, another VSS strategy is proposed in [26], which uses the cross-correlation between the current and previous error values to update the step-size. The work in [27] proposes an improved version of [26] by incorporating the cross-correlation of the input signal in the update equation as well. However, despite these improvements, the VSS strategy of [25] generally outperforms the two algorithms as well as most other VSS algorithms. A noise-constrained VSS algorithm is proposed in [28]. The algorithm uses Lagrange multipliers with the constraint that the noise variance is known. The work in [29] proposes a VSS normalized LMS algorithm that provided a VSS strategy to the algorithm of [24]. The authors in [30] propose a sparsity-aware VSS strategy, which updates the step-size using the absolute value of the instantaneous error instead of the energy. The algorithm is shown to perform better for sparse systems compared with other VSS strategies.

1.2 VSS strategies for WSNs

The idea of varying the step-size has been extended to estimation in WSNs. In some cases, the authors have proposed to simply incorporate existing VSS strategies within the WSN framework [13–16]. The work in [13,14] incorporates the VSS strategy proposed in [25] directly into the distributed estimation framework. The same VSS strategy has been used to transform domain distributed estimation by the authors in [15]. The authors in [16] use a recently proposed sparse VSS technique of [30] for distributed estimation of compressible systems. While others have used the setup of the network to specifically derive new VSS strategies [17–21]. The work in [17] improves the strategy introduced in [13] by diffusing the error values into the network as well. The work in [18] proposes a noise constrained distributed algorithm, derived using Lagrangian multipliers. The authors in [19–21] derive VSS strategies from mean square deviation calculations.

1.3 Analysis for VSS strategies

In general, each VSS algorithm aims to improve performance at the cost of computational complexity. Generally, this is an acceptable trade-off as the improvement in performance is considerable. The drawback is that the additional complexity results in the analysis of the algorithm becoming tedious. This complexity is further increased for the case of WSNs as different sensor nodes collaborate with each other to improve performance. Authors have used various assumptions in order to perform the analysis of these algorithms. However, each algorithm has been dealt with independently in order to find a closed-form solution. An exact method of analysis has been recently proposed in [31,32]. Although the results are accurate, this method is mathematically complex as well as algorithm specific and cannot be generalized for all algorithms. A generic treatment of VSS algorithms has been presented in [33] but without taking into account a WSN. In a WSN, the data is shared between nodes and this fact needs to be taken into account while performing the analysis. Due to this very important factor, the analysis of [33] cannot be extended to the WSN scenario.

Table 1 Distributed VSS algorithms and analysis in literature

Ref.	M.A.[a]	M.S.A.[b]	S.S.A.[c]	Gen.[d]
[13]			✓	
[14]	✓	✓	✓	
[15]				
[16]	✓		✓	
[17]				
[18]	✓	✓	✓	
[19]				
[20]				
[21]	✓	✓	✓	
Proposed	✓	✓	✓	✓

[a] Mean analysis
[b] Mean-square analysis
[c] Steady-state analysis
[d] Generic analysis

Table 2 Description of the acronyms used in this paper

Acronym	Description
IoT	Internet of Things
VSS	Variable step-size
LMS	Least mean square
WSN	Wireless sensor network
MSD	Mean square deviation
EMSE	Excess mean square error
ATC	Adapt-then-combine
DLMS	Diffusion LMS
VSSDLMS	Variable step-size diffusion LMS
SNR	Signal-to-noise ratio
KJ	Algorithm proposed in [25] and used in [14]
NC	Algorithm proposed in [28] and used in [18]
Sp	Algorithm proposed in [30] and used in [16]

Table 1 lists which analysis has been performed for the VSS algorithms of [13–21] and compares them with the proposed work. As can be seen, most of the work includes either no analysis or only a part of the analysis. None of the authors have, however, performed the analysis in a generic way.

1.4 Contributions

This work presents a generalized analysis approach for LMS-based VSS techniques that have been applied to WSNs. The proposed generalized analysis can be applied to most existing as well as any forthcoming VSS approaches. Following are the main contributions of this work:

- Derivation of the step-size limit for stability of distributed VSSLMS algorithms in WSNs.
- Generalized transient analysis for distributed VSSLMS algorithms in WSNs, including derivation of iterative equations for mean square deviation (MSD) and excess mean square error (EMSE).
- Derivation of steady-state equations for MSD and EMSE for distributed VSSLMS algorithms in WSNs.
- Derivation of steady-state step-size terms for the VSSLMS algorithms being used as case study in this work.
- Validation of theoretical analysis through experimentation.

A list of the acronyms being used within the paper are given below:

The rest of the paper is organized as follows. Section 2 presents a working system model and problem statement. Section 3 details the complete theoretical analysis for diffusion-based distributed VSS LMS algorithms. Simulation results are presented in Sect. 4. Section 5 concludes the paper (Table 2).

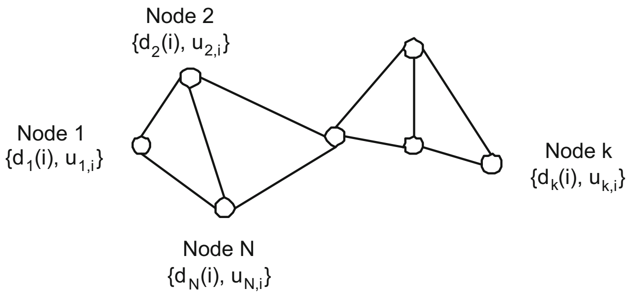

Fig. 1 An illustration of adaptive network of N nodes

2 System model

We consider an adaptive network consisting of N sensor nodes, as shown in Fig. reffigspsans, deployed over a geographical area to estimate an M dimensional unknown parameter vector, whose optimum value is denoted by $\mathbf{w}^o \in \mathbb{R}^M$. We denote the neighborhood of any node k by $\mathcal{N}_k$ and number of neighbors of node k by n_k. The neighborhood of a node k is a set of nodes in close vicinity such that they have a single-hop communication link with node k, i.e., for $l = 1, 2, \ldots, N$ and $l \neq k$, nodes l and k have single-hop communication between them if $l \in \mathcal{N}_k$.

Each node k has access to a time realization of a known regressor row vector $\mathbf{u}_k(i)$ of length M and a scalar measurement $d_k(i)$, which are related by

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

where $k = 1, \ldots, N$, $v_k(i)$ is zero-mean additive white Gaussian noise with variance $\sigma^2_{v_k}$, which is spatially uncorrelated, and i is the time index. At each node, these measurements are used to generate an estimate of the unknown vector $\mathbf{w}^o$, denoted by $\mathbf{w}_k(i)$.

The fully distributed *Adapt-Then-Combine (ATC)* DLMS algorithm proposed in [12] is summarized below.

$$e_k(i) = d_k(i) - \mathbf{u}_k(i)\mathbf{w}_k(i) \tag{2}$$

$$\boldsymbol{\Psi}_k(i) = \mathbf{w}_k(i) + \mu_k \mathbf{u}_k^T(i) e_k(i) \tag{3}$$

$$\mathbf{w}_k(i+1) = \sum_{l \in \mathcal{N}_k} c_{lk} \boldsymbol{\Psi}_l(i), \tag{4}$$

where $\boldsymbol{\Psi}_k(i)$ is the intermediate update, $e_k(i)$ is the instantaneous error and c_{lk} are the combination weights [12].

The step-size μ_k can be varied using any one of several available strategies [13–21]. Thus, the generic VSSDLMS algorithm is described by (2), (4) and the following set of equations

$$\boldsymbol{\Psi}_k(i) = \mathbf{w}_k(i) + \mu_k(i) \mathbf{u}_k^T(i) e_k(i) \tag{5}$$

$$\mu_k(i+1) = f\{\mu_k(i)\}, \tag{6}$$

where $f\{\mu_k(i)\}$ is a function that defines the update equation for the step-size and varies for each VSS strategy.

While performing the analysis of the LMS algorithm, the input regressor vector is assumed to be independent of the estimated vector. For the VSS algorithms, it is generally assumed that control parameters are chosen such that the step-size and the input regressor vector are asymptotically independent of each other, resulting in a closed-form steady-state solution that closely matches with the simulation results. For some VSS algorithms, the analytical and simulation results are closely matched during the transition 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 diffusion-based VSSLMS algorithms, in lieu with the assumptions mentioned above. A list of symbols used in this paper is presented in Table 3.

Table 3 Description of the symbols used in the model

Symbol	Description
N	Number of nodes in the network
M	Length of the unknown parameter vector
k	Denotes Node number
i	Denotes time instant
d_k	Scalar measurement for node k
$\mathbf{u}_k$	Input regressor row vector for node k
$\mathbf{w}^o$	Unknown vector of length M
v_k	Additive random noise for node k
σ^2_v	Noise power
e_k	Measurement error for node k
$\boldsymbol{\Psi}_k$	Intermediate estimate vector for node k
$\mathbf{w}_k$	Final estimate vector for node k
μ_k	Step-size for node k
c_{lk}	Combination weight, node l to node k
$f\{\mu_k(i)\}$	Step-size update function
$\mathbf{d}$	Vector of measurements for the network
$\mathbf{U}$	Input regressor matrix for the network
$\mathbf{v}$	Additive random noise vector for the network
$\mathbf{e}$	Measurement error vector for the network
$\boldsymbol{\Psi}$	Intermediate estimate vector for the network
$\mathbf{w}$	Final estimate vector for the network
$\mathbf{D}$	Step-size matrix for the network
$\mathbf{C}$	Combination weight matrix, size $N \times N$
$\mathbf{w}^{(o)}$	Unknown vector of length MN
$\mathbf{G}$	Combination weight matrix, size $MN \times MN$
$f\{\mathbf{D}(i)\}$	Step-size update function matrix
$\mathbf{Q}$	$MN \times M$ sized matrix of N identity matrices of size M
$\mathbf{I}_M$	Identity matrix of size M
$\tilde{\mathbf{w}}$	Weight-error matrix for the network
$\mathbf{R}_\mathbf{U}$	Autocorrelation matrix for $\mathbf{U}$
$\mathbf{R}_{\mathbf{u},k}$	Autocorrelation matrix for $\mathbf{u}_k$
λ	Eigenvalue for $\mathbf{R}_{\mathbf{u},k}$
$\boldsymbol{\Sigma}$	Weighting matrix
$\mathbf{T}$	Eigenvector matrix for $\mathbf{R}_\mathbf{U}$
$\boldsymbol{\Lambda}$	Diagonal eigenvalue matrix for $\mathbf{R}_\mathbf{U}$
$\mathbf{R}_\mathbf{v}$	Autocorrelation matrix for network noise
ss	Denotes steady-state values

3 Proposed unified analysis framework

In a distributed network, the nodes exchange data, as can be seen from (4). As a result, there is going to exist a correlation between the data of the entire network. To account for this inter-node dependence, the performance of the whole network needs to be studied. Some new variables are introduced, transforming the local ones into global variables as follows:

$$\begin{aligned}
\mathbf{w}(i) &= \text{col}\left\{\mathbf{w}_1(i), \ldots, \mathbf{w}_N(i)\right\}, \\
\boldsymbol{\Psi}(i) &= \text{col}\left\{\boldsymbol{\Psi}_1(i), \ldots, \boldsymbol{\Psi}_N(i)\right\}, \\
\mathbf{U}(i) &= \text{diag}\left\{\mathbf{u}_1(i), \ldots, \mathbf{u}_N(i)\right\}, \\
\mathbf{D}(i) &= \text{diag}\left\{\mu_1(i)\mathbf{I}_M, \ldots, \mu_N(i)\mathbf{I}_M\right\}, \\
\mathbf{d}(i) &= \text{col}\left\{d_1(i), \ldots, d_N(i)\right\}, \\
\mathbf{v}(i) &= \text{col}\left\{v_1(i), \ldots, v_N(i)\right\}.
\end{aligned}$$

Using the new variables, the set of equations representing the entire network is given by

$$
\begin{aligned}
\mathbf{d}(i) &= \mathbf{U}(i)\mathbf{w}^{(o)} + \mathbf{v}(i),\\
\boldsymbol{\Psi}(i+1) &= \mathbf{w}(i) + \mathbf{D}(i)\mathbf{U}^T(i)\left(\mathbf{d}(i) - \mathbf{U}(i)\mathbf{w}(i)\right),\\
\mathbf{w}(i+1) &= \mathbf{G9}(i+1),\\
\mathbf{D}(i+1) &= f\{\mathbf{D}(i)\},
\end{aligned}
\tag{7}
$$

where $\mathbf{w}^{(o)} = \mathbf{Q}\mathbf{w}^o$, $\mathbf{Q} = col\{\mathbf{I}_\mathrm{M}, \mathbf{I}_\mathrm{M}, \ldots, \mathbf{I}_\mathrm{M}\}$ is a $MN \times M$ matrix, $\mathbf{G} = \mathbf{C} \otimes \mathbf{I}_M$, $\mathbf{C}$ is an $N \times N$ weighting matrix, $\{\mathbf{C}\}_{lk} = c_{lk}$ and $\otimes$ is the Kronecker product.

3.1 Mean analysis

The analysis is now carried out using the new set of equations. The global weight-error vector is given by

$$
\tilde{\mathbf{w}}(i) = \mathbf{w}^{(o)} - \mathbf{w}(i).
\tag{8}
$$

Since $\mathbf{G}\mathbf{w}^{(o)} \triangleq \mathbf{w}^{(o)}$, by incorporating the global weight-error vector into (7), we get

$$
\begin{aligned}
\tilde{\mathbf{w}}(i+1) &= \mathbf{G}\tilde{\boldsymbol{\Psi}}(i+1)\\
&= \mathbf{G}\tilde{\mathbf{w}}(i) - \mathbf{G}\mathbf{D}(i)\mathbf{U}^T(i)\left(\mathbf{U}(i)\tilde{\mathbf{w}}(i) + \mathbf{v}(i)\right)\\
&= \mathbf{G}\left(\mathbf{I}_{MN} - \mathbf{D}(i)\mathbf{U}^T(i)\mathbf{U}(i)\right)\tilde{\mathbf{w}}(i)\\
&\quad - \mathbf{G}\mathbf{D}(i)\mathbf{U}^T(i)\mathbf{v}(i).
\end{aligned}
\tag{9}
$$

Using the assumption that the step-size matrix $\mathbf{D}(i)$ is independent of the regressor matrix $\mathbf{U}(i)$ [14,16,18,25–29], the following relation holds true asymptotically

$$
\mathbb{E}\left[\mathbf{D}(i)\mathbf{U}^T(i)\mathbf{U}(i)\right] \approx \mathbb{E}\left[\mathbf{D}(i)\right]\mathbb{E}\left[\mathbf{U}^T(i)\mathbf{U}(i)\right],
\tag{10}
$$

where $\mathbb{E}\left[\mathbf{U}^T(i)\mathbf{U}(i)\right] = \mathbf{R}_\mathrm{U}$ is the auto-correlation matrix of $\mathbf{U}(i)$. Now, taking the expectation on both sides of (9) and simplifying gives

$$
\mathbb{E}\left[\tilde{\mathbf{w}}(i+1)\right] = \mathbf{G}\left(\mathbf{I}_{MN} - \mathbb{E}\left[\mathbf{D}(i)\right]\mathbf{R}_\mathrm{U}\right)\mathbb{E}\left[\tilde{\mathbf{w}}(i)\right].
\tag{11}
$$

The expectation of the second term of the right-hand side of (9) is zero due to the measurement noise being zero-mean as well as spatially uncorrelated with the input regressor.

It can be seen from (11) that the term defining the stability of the algorithm is $(\mathbf{I}_{MN} - \mathbb{E}\left[\mathbf{D}(i)\right]\mathbf{R}_\mathrm{U})$. As the matrix $\mathbf{D}(i)$ is diagonal, this term can be further simplified to the node-level as $\left(\mathbf{I} - \mathbb{E}\left[\mu_k(i)\right]\mathbf{R}_{\mathbf{u},k}\right)$. Thus, the stability condition is given by

$$
\prod_{i=0}^{n}\left(\mathbf{I} - \mathbb{E}\left[\mu_k(i)\right]\mathbf{R}_{\mathbf{u},k}\right) \rightarrow 0, \quad as\, n \rightarrow \infty
\tag{12}
$$

which holds true if the mean of the step-size is governed by

$$
0 < \mathbb{E}\left[\mu_k(i)\right] < \frac{2}{\lambda_{\max}\left(\mathbf{R}_{\mathbf{u},k}\right)}, \quad 1 \le k \le N,
\tag{13}
$$

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

3.2 Mean-square analysis

Following the analysis procedure of [23], we take the weighted norm of (9) and then apply the expectation operator. After simplifying we get

$$
\begin{aligned}
\mathbb{E}\left[\|\tilde{\mathbf{w}}(i+1)\|^2_{\boldsymbol{\Sigma}}\right] &= \mathbb{E}\left[\|\tilde{\mathbf{w}}(i)\|^2_{\hat{\boldsymbol{\Sigma}}}\right]\\
&\quad + \mathbb{E}\left[\mathbf{v}^T(i)\mathbf{Y}^T(i)\boldsymbol{\Sigma}\mathbf{Y}(i)\mathbf{v}(i)\right],
\end{aligned}
\tag{14}
$$

where

$$
\mathbf{Y}(i) = \mathbf{G}\mathbf{D}(i)\mathbf{U}^T(i)
\tag{15}
$$

$$
\begin{aligned}
\hat{\boldsymbol{\Sigma}} &= \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G} - \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{Y}(i)\mathbf{U}(i)\\
&\quad - \mathbf{U}^T(i)\mathbf{Y}^T(i)\boldsymbol{\Sigma}\mathbf{G} + \mathbf{U}^T(i)\mathbf{Y}^T(i)\boldsymbol{\Sigma}\mathbf{Y}(i)\mathbf{U}(i).
\end{aligned}
\tag{16}
$$

The analysis becomes quite tedious for non-Gaussian data. Therefore, the data is assumed to be Gaussian, without loss of generality [23]. The auto-correlation matrix is decomposed as $\mathbf{R}_\mathrm{U} = \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^T$, where $\boldsymbol{\Lambda}$ is an eigenvalues diagonal matrix and $\mathbf{T}$ is a matrix of eigenvectors corresponding to the eigenvalues, such that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$. Using $\mathbf{T}$, the variables are redefined as

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

where the input regressors are considered independent for each node and the step-size matrix $\mathbf{D}(i)$ is block-diagonal so it remains the same.

Using the data independence assumption [23] and simplifying, we arrive at the final recursive update equation

$$
\begin{aligned}
\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T(i)\bar{\boldsymbol{\sigma}}\\
&\quad + \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(i)\left[\mathbf{F}(i) - \mathbf{I}_{M^2N^2}\right]\bar{\boldsymbol{\sigma}}}\\
&\quad + \mathcal{B}(i)\left[\mathbf{F}(i) - \mathbf{I}_{M^2N^2}\right]\bar{\boldsymbol{\sigma}},
\end{aligned}
\tag{17}
$$

where

$$
\begin{aligned}
\mathbf{F}(i) = \big[&\mathbf{I}_{M^2N^2} - \left(\mathbf{I}_{MN} \odot \boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(i)\right]\right)\\
&- \left(\boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(i)\right] \odot \mathbf{I}_{MN}\right) + \left(\mathbb{E}\left[\mathbf{D}(i) \odot \mathbf{D}(i)\right]\right)\mathbf{A}\big]
\end{aligned}
$$

$$\cdot \left(\mathbf{G}^T \odot \mathbf{G}^T \right). \tag{18}$$

$$\mathcal{A}(i+1) = \mathcal{A}(i)\mathbf{F}(i). \tag{19}$$

$$\mathcal{B}(i+1) = \mathcal{B}(i)\mathbf{F}(i) + \mathbf{b}^T(i)\mathbf{I}_{M^2N^2}. \tag{20}$$

Finally, taking the weighting matrix $\mathbf{\Sigma} = \mathbf{I}_{M^2N^2}$ gives the mean-square-deviation (MSD) while $\mathbf{\Sigma} = \mathbf{\Lambda}$ gives the EMSE. The detailed analysis and description of the variables are given in Appendix A.

3.3 Steady-state analysis

At steady-state, (31) and (18) become

$$\mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\bar{\boldsymbol{\sigma}}}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\mathbf{F}_{ss}\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T_{ss}\bar{\boldsymbol{\sigma}}, \tag{21}$$

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

where $\mathbf{D}_{ss} = \text{diag}\left\{\mu_{ss,k}\mathbf{I}_M\right\}, \mathbf{b}_{ss} = \mathbf{R_v}\mathbf{D}^2_{ss}\mathbf{\Lambda}$ and $\mathbf{D}^2_{ss} = \text{diag}\left\{\mu^2_{ss,k}\mathbf{I}_M\right\}$. Simplifying (21), we get

$$\mathbb{E}\left[\|\bar{\mathbf{w}}_{ss}\|^2_{\bar{\boldsymbol{\sigma}}}\right] = \mathbf{b}^T_{ss}\left[\mathbf{I}_{M^2N^2} - \mathbf{F}_{ss}\right]^{-1}\bar{\boldsymbol{\sigma}}. \tag{23}$$

This equation gives the steady-state performance measure for the entire network. In order to solve for the steady-state values of MSD and EMSE, we take $\bar{\boldsymbol{\sigma}} = \text{bvec}\{\mathbf{I}_{M^2N^2}\}$ and $\bar{\boldsymbol{\sigma}} = \text{bvec}\{\mathbf{\Lambda}\}$, respectively.

3.4 Steady-state step-size analysis

The analysis presented in the above section has been generic for any VSS algorithm. In this section, 3 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 (23) and (22). The 3 different VSS algorithms and their step-size update equations are given in Table 4. The first algorithm, denoted **KJ** is the work of Kwong and Johnston [25], as used in [14]. The **NC** algorithm refers to the noise-constrained LMS algorithm [28], as used in [18]. Finally, **Sp** refers to the Sparse VSSLMS algorithm of [30], as used in [16]. It should be noted that the step-size matrix for the network, **D**, is diagonal. Therefore, the step-size for each node can be studied independently.

Applying the expectation operator and simplifying results in the equations presented in Table 5, where $\zeta_k(i)$ is the value of the EMSE for node k.

At steady-state, the step-size is given by $\mu_{k,ss}$ and the approximate steady-state equations are given in Table 6.

Table 4 Step-size update equations for the VSSLMS algorithms

Algo.	Step-size update equation
KJ [14]	$\mu_k(i+1) = \alpha_{kj}\mu_k(i) + \gamma_{kj}e^2_k(i)$
NC [18]	$\theta_{k,nc}(i+1) = (1-\alpha_{nc})\theta_{k,nc}(i) + \frac{\alpha_{nc}}{2}\left(e^2_k(i) - \sigma^2_{v,k}\right)$ $\mu_k(i+1) = \mu_0\left(1 + \gamma_{nc}\theta_{k,nc}(i+1)\right)$
Sp [16]	$\mu_k(i+1) = \alpha_{sp}\mu_k(i) + \gamma_{sp}\left\|e_k(i)\right\|$

Table 5 Expectations of the update equations from Table 4

Algo.	Expectation of update equation
KJ [14]	$\mathbb{E}[\mu_k(i+1)] = \alpha_{kj}\mathbb{E}[\mu_k(i)] + \gamma_{kj}\left[\zeta_k(i) + \sigma^2_{v,k}\right]$
NC [18]	$\mathbb{E}\left[\theta_{k,nc}(i+1)\right] = (1-\alpha_{nc})\mathbb{E}\left[\theta_{k,nc}(i)\right] + \frac{\alpha_{nc}}{2}\zeta_k(i)$ $\mathbb{E}[\mu_k(i+1)] = \mu_0\left(1 + \gamma_{nc}\mathbb{E}\left[\theta_{k,nc}(i+1)\right)\right]$
Sp [16]	$\mathbb{E}[\mu_k(i+1)] = \alpha_{sp}\mathbb{E}[\mu_k(i)] + \gamma_{sp}\sqrt{2\sigma^2_{v,k}/\pi}$

Table 6 Steady-state step-size values for equations from Table 4

Algo.	Steady-state step-size value
KJ [14]	$\mu_{k,ss} \approx \frac{\gamma_{kj}}{1-\alpha_{kj}}\sigma^2_{v,k}.$
NC [18]	$\mu_{k,ss} \approx \mu_0.$
Sp [16]	$\mu_{k,ss} \approx \frac{\gamma_{sp}}{1-\alpha_{sp}}\sqrt{2\sigma^2_{v,k}/\pi}.$

It should be noted that the steady-state EMSE values are assumed small enough to be ignored.

4 Results and discussion

In this section, the analysis presented above will be tested upon three VSS algorithms listed in Table 4 when they are incorporated within the DLMS framework. The analysis is verified through three different experiments. The first experiment plots the theoretical transient MSD using (17) and compares with simulation results. Further, steady-state results are tabulated to see the effects of the assumptions. Simulation steady-state results are compared to theoretical results attained using (17) as well as (23). In the second experiment, steady-state MSD results are compared for different network sizes while the signal-to-noise ratio (SNR) is varied. Finally, a tabular comparison has been shown to ascertain the difference between the results obtained from (23) with those from [14,18] and [16].

4.1 Experiment 1

For the first experiment, the network size, $N = 10$, the size of the unknown parameter vector, $M = 4$ and the SNR is varied between 0, 10 and 20 dB. The step-size control parameters

Table 7 Step-size control parameters for experiment 1

Algo.	SNR (dB)	Parameters
KJ [14]	All	$\mu_0 = 0.01, \gamma_{kj} = 1e - 3$
	0	$\alpha_{kj} = 0.95$
	10	$\alpha_{kj} = 0.98$
	20	$\alpha_{kj} = 0.99$
NC [18]	All	$\mu_0 = 0.01, \gamma_{nc} = 1, \alpha_{nc} = 1e - 4$
Sp [16]	All	$\mu_0 = 0.01, \gamma_{sp} = 1e - 3$
	0	$\alpha_{sp} = 0.97$
	10	$\alpha_{sp} = 0.98$
	20	$\alpha_{sp} = 0.99$

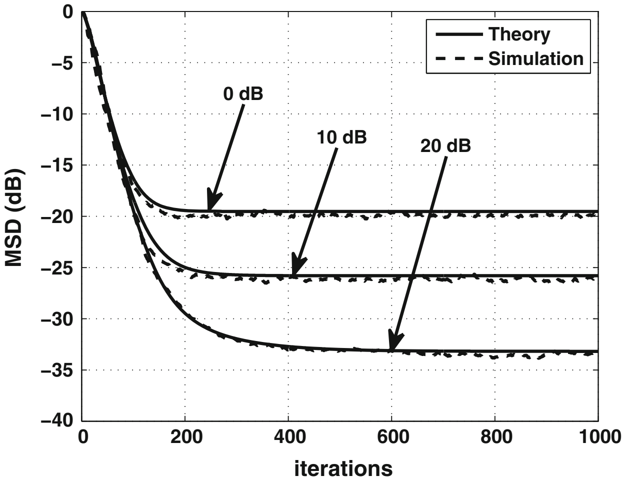

Fig. 2 Theory (17) v simulation MSD comparison for the *KJ* algorithm [14]

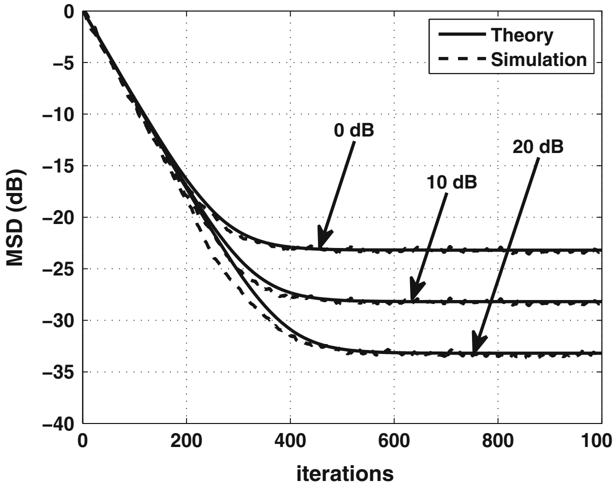

Fig. 3 Theory (17) v simulation MSD comparison for the *NC* algorithm [18]

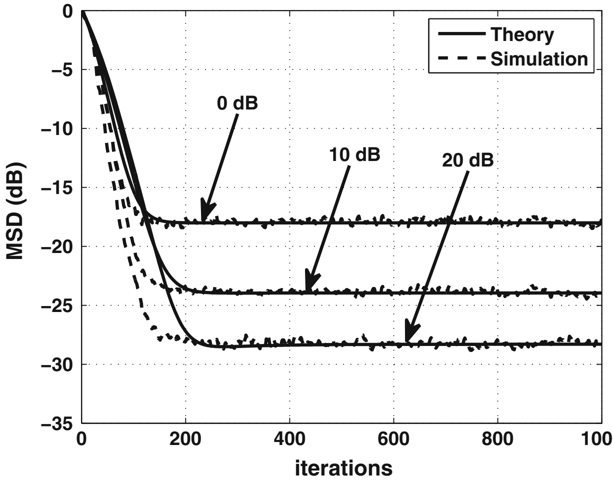

Fig. 4 Theory (17) v simulation MSD comparison for the *Sp* algorithm [16]

chosen for this experiment are given in Table 7. The values are slightly different in some cases in order to maintain similar convergence speed. The results are shown in Figs. 2, 3, 4.

The comparison for the **KJ** algorithm are shown in Fig. 2. There is a slight discrepancy due to the assumptions that have been made. However, this discrepancy is acceptable as the steady-state results match very closely. The results for the **NC** algorithm are shown in Fig. 3. There is a mismatch in the transient stage in this case as well but the results match very closely at steady-state again. Finally, the comparison for the **Sp** algorithm is shown in Fig. 4. The mismatch during the transient stage is greater for this algorithm compared with the previous two cases. However, since the step-size is assumed to be asymptotically independent of the regressor data and there is an excellent match at steady-state, these results are completely acceptable. The steady-state results are tabulated in Table 8. Due to the assumption that the EMSE is small enough to be ignored at steady-state, there exists a slight mismatch between the results of (17) and (23). However, the results are overall closely matched and the results are acceptable.

4.2 Experiment 2

For the second experiment, the SNR is varied from 0 dB to 40 dB and the steady-state results are plotted. These are compared with the theoretical results obtained from (23). The network size is varied between $N = 10$ and $N = 20$. The step-size controlling parameters used for this experiment are given in Table 9. The initial step-size is kept small for the **NC** algorithm as the initial step-size effects the steady-state value as shown in Table 6. However, there is no relation between the steady-state step-size value and the initial step-size value for the other algorithms. This has also been shown for the case of the **KJ** algorithm in [14]. The results are shown in Fig. 5

Table 8 Theory versus simulation comparison for steady-state MSD for the different VSS algorithms

Algo.	SNR	MSD eq. (17)	MSD eq. (23)	MSD sim.
KJ [14]	0	– 19.53	–19.59	– 19.95
	10	– 25.80	–25.85	– 26.15
	20	– 33.20	–33.20	– 33.55
NC [18]	0	– 23.19	–23.20	23.29
	10	– 28.19	–28.20	– 28.31
	20	– 33.19	–33.20	– 33.31
Sp [16]	0	– 18.01	–18.01	– 17.97
	10	– 23.96	–23.96	– 23.92
	20	– 28.30	–28.30	– 28.25

Table 9 Step-size control parameters for experiment 2

Algo.	SNR	Parameters
KJ [14]	All	$\mu_0 = 0.01, \gamma_{kj} = 1e - 3$
	0	$\alpha_{kj} = 0.95$
	10	$\alpha_{kj} = 0.97$
	20	$\alpha_{kj} = 0.99$
	30	$\alpha_{kj} = 0.99$
	40	$\alpha_{kj} = 0.995$
NC [18]	All	$\mu_0 = 0.003, \gamma_{nc} = 1, \alpha_{nc} = 1e - 4$
Sp [16]	All	$\mu_0 = 0.01, \gamma_{sp} = 1e - 3$
	0	$\alpha_{sp} = 0.95$
	10	$\alpha_{sp} = 0.97$
	20	$\alpha_{sp} = 0.99$
	30	$\alpha_{sp} = 0.99$
	40	$\alpha_{sp} = 0.995$

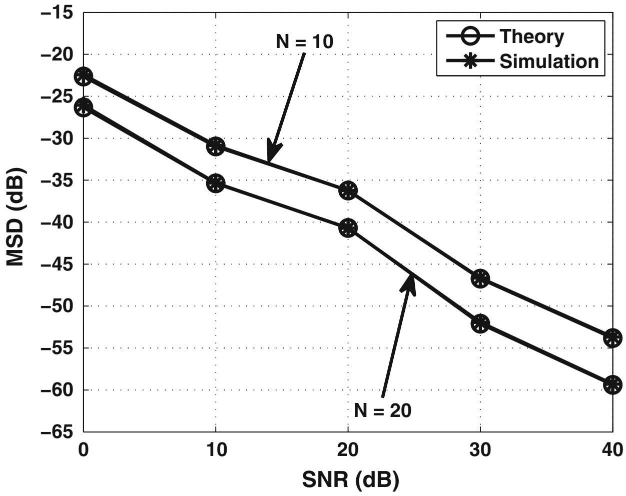

Fig. 5 Theory (23) v simulation MSD comparison for the *KJ* algorithm [14] for different network sizes and varying SNR

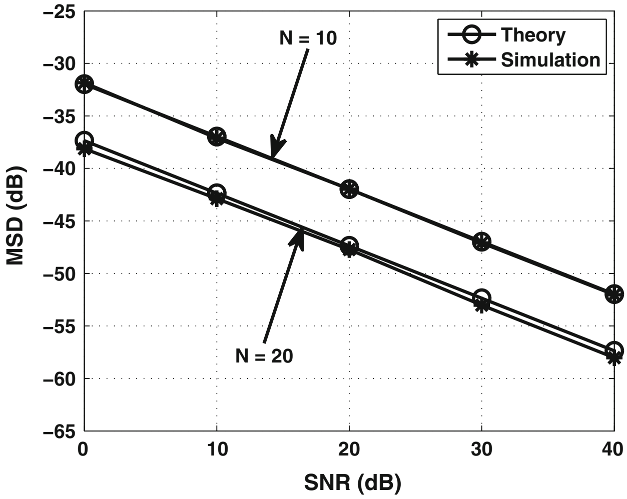

Fig. 6 Theory (23) v simulation MSD comparison for the *NC* algorithm [18] for different network sizes and varying SNR

for the **KJ** algorithm. There is a close match and a steady downward trend with increase in SNR. The results for the **NC** algorithm are shown in Fig. 6. There is a slight mismatch but this is not significant and can be ignored. Finally, the results for the **Sp** algorithm are shown in Fig. 7. There is an excellent match and a steady improvement in performance with increase in SNR.

4.3 Experiment 3

In this final experiment, the steady-state MSD theoretical results from [14,18] and [16] are compared with the results obtained from (23). The analysis performed in [14,18] and [16] gives exact expressions for $\mu_{k,ss}$ as well as $\mu^2_{k,ss}$ for evaluating steady-state MSD. In the present work, a generic expression has been proposed. This experiment shows that the results from the exact analysis and those obtained from (23) match closely. The unknown vector length is varied between $M = 2$, $M = 4$ and $M = 6$. The SNR is varied between 0, 10 and 20 dB. The step-size control parameters are given in Table 10. The results are shown in Table 10. As can be seen, there are slight mismatches among certain values. However, the values are very closely matched for all cases. Thus, the proposed analysis procedure has been verified as generic and applicable to different VSS strategies in WSNs (Table 11).

5 Conclusion

Parameter estimation is an important aspect for WSNs that form an integral part of the IoT paradigm. This work presents a generalized approach for the theoretical analysis of LMS-based VSS algorithms being employed for estimation in diffusion-based WSNs. The analysis provided here can prove to be an excellent tool for the analysis of any VSS approach

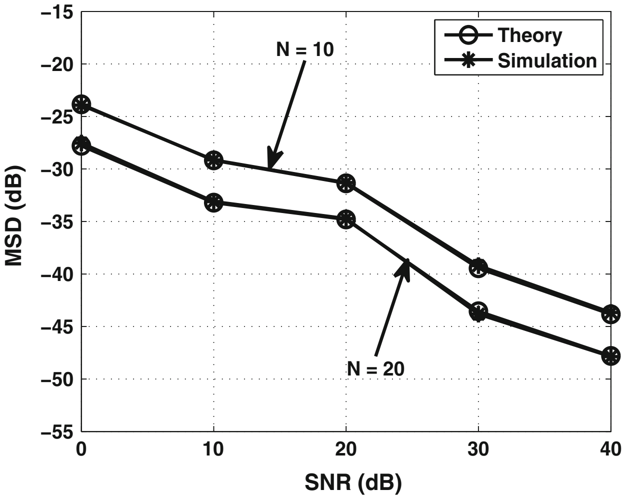

Fig. 7 Theory (23) v simulation MSD comparison for the *Sp* algorithm [16] for different network sizes and varying SNR

Table 10 Step-size control parameters for experiment 3

Algo.	Parameters
KJ	$\mu_0 = 0.01, \alpha = 0.95, \gamma_{kj} = 1e-4$
NC	$\mu_0 = 0.001, \gamma_{nc} = 1, \alpha_{nc} = 1e-4$
Sp	$\mu_0 = 0.01, \alpha = 0.97, \gamma_{sp} = 1e-4$

applied to the WSN framework in future. Various existing algorithms have been tested thoroughly to verify the results of the presented work under different conditions. Simulation results confirm the generic behavior for the proposed analysis for both the transient state as well as steady-state. Furthermore, a comparison has been performed between the proposed analysis results and results obtained from the analysis that already exists in the literature. Results were found to be closely matched. further strengthening our claim that the proposed analysis is generic for VSS strategies applied to WSNs.

Appendix A

Here, we present the detailed mean-square analysis. Applying the expectation operator to the weighting matrix of (14) gives

Table 11 Comparison between proposed and existing analyses

SNR (db)	Algo.	Ref.	MSD		
			$M = 2$	$M = 4$	$M = 6$
0	KJ	Eq. (23)	– 33.81	– 30.80	– 29.03
		[14]	– 33.80	– 30.78	– 29.00
	NC	Eq. (23)	– 36.90	– 33.89	– 32.13
		[18]	– 36.87	– 33.83	– 32.03
	Sp	Eq. (23)	– 32.52	– 29.50	– 27.74
		[16]	– 32.48	– 29.47	– 27.70
10	KJ	Eq. (23)	– 43.92	– 40.91	– 39.15
		[14]	– 43.92	– 40.91	– 39.14
	NC	Eq. (23)	– 41.90	– 38.89	– 37.13
		[18]	– 41.90	– 38.88	– 37.10
	Sp	Eq. (23)	– 40.11	– 37.10	– 35.34
		[16]	– 40.08	– 37.06	– 35.30
20	KJ	Eq. (23)	– 53.96	– 50.95	– 49.19
		[14]	– 53.96	– 50.95	– 49.19
	NC	Eq. (23)	– 46.90	– 43.89	– 42.13
		[18]	– 46.90	– 43.89	– 42.12
	Sp	Eq. (23)	– 47.67	– 44.66	– 42.89
		[16]	– 47.63	– 44.62	– 42.86

$$\begin{aligned}\mathbb{E}\left[\hat{\boldsymbol{\Sigma}}\right] &= \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G} - \mathbf{G}^T\boldsymbol{\Sigma}\mathbb{E}\left[\mathbf{Y}(i)\mathbf{U}(i)\right] \\ &\quad - \mathbb{E}\left[\mathbf{U}^T(i)\mathbf{Y}^T(i)\right]\boldsymbol{\Sigma}\mathbf{G} \\ &\quad + \mathbb{E}\left[\mathbf{U}^T(i)\mathbf{Y}^T(i)\boldsymbol{\Sigma}\mathbf{Y}(i)\mathbf{U}(i)\right] \\ &= \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G} - \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G}\mathbb{E}\left[\mathbf{D}(i)\right]\mathbb{E}\left[\mathbf{U}^T(i)\mathbf{U}(i)\right] \\ &\quad - \mathbb{E}\left[\mathbf{U}^T(i)\mathbf{U}(i)\right]\mathbb{E}\left[\mathbf{D}(i)\right]\mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G} \\ &\quad + \mathbb{E}\left[\mathbf{U}^T(i)\mathbf{Y}^T(i)\boldsymbol{\Sigma}\mathbf{Y}(i)\mathbf{U}(i)\right], \end{aligned} \tag{24}$$

For ease of notation, we denote $\mathbb{E}\left[\hat{\boldsymbol{\Sigma}}\right] = \boldsymbol{\Sigma}'$ for the remaining analysis.

Next, using the Gaussian transformed variables as gives in Sect. 3.2, (14) and (24) are rewritten, respectively, as

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\bar{\boldsymbol{\Sigma}}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\bar{\boldsymbol{\Sigma}}'}\right] \\ &\quad + \mathbb{E}\left[\mathbf{v}^T(i)\bar{\mathbf{Y}}^T(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\mathbf{v}(i)\right], \end{aligned} \tag{25}$$

and

$$\begin{aligned}\bar{\boldsymbol{\Sigma}}' &= \bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}} - \bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}}\mathbb{E}\left[\mathbf{D}(i)\right]\mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{U}}(i)\right] \\ &\quad - \mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{U}}(i)\right]\mathbb{E}\left[\mathbf{D}(i)\right]\bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}} \\ &\quad + \mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{Y}}(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\bar{\mathbf{U}}(i)\right] \\ &= \bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}} - \bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}}\mathbb{E}\left[\mathbf{D}(i)\right]\boldsymbol{\Lambda} \\ &\quad - \boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(i)\right]\bar{\mathbf{G}}^T\bar{\boldsymbol{\Sigma}}\bar{\mathbf{G}} \\ &\quad + \mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{Y}}(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\bar{\mathbf{U}}(i)\right], \end{aligned} \tag{26}$$

where $\bar{\mathbf{Y}}(i) = \bar{\mathbf{G}}D(i)\bar{\mathbf{U}}^T(i)$ and $\mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{U}}(i)\right] = \boldsymbol{\Lambda}$.

The two terms that need to be solved are $\mathbb{E}\left[\mathbf{v}^T(i)\bar{\mathbf{Y}}^T(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\mathbf{v}(i)\right]$ and $\mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{Y}}(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\bar{\mathbf{U}}(i)\right]$. Using the bvec$\{.\}$ operator and the block Kronecker product, denoted by $\odot$ [34], the two terms are simplified as

$$\mathbb{E}\left[\mathbf{v}^T(i)\bar{\mathbf{Y}}^T(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\mathbf{v}(i)\right] = \mathbf{b}^T(i)\bar{\boldsymbol{\sigma}}, \tag{27}$$

and

$$\begin{aligned}&\text{bvec}\left\{\mathbb{E}\left[\bar{\mathbf{U}}^T(i)\bar{\mathbf{Y}}^T(i)\bar{\boldsymbol{\Sigma}}\bar{\mathbf{Y}}(i)\bar{\mathbf{U}}(i)\right]\right\}\\ &\quad= \left(\mathbb{E}\left[\mathbf{D}(i)\odot\mathbf{D}(i)\right]\right)\mathbf{A}\left(\mathbf{G}^T\odot\mathbf{G}^T\right)\bar{\boldsymbol{\sigma}},\end{aligned} \tag{28}$$

where $\bar{\boldsymbol{\sigma}} = \text{bvec}\left\{\bar{\boldsymbol{\Sigma}}\right\}$, $\mathbf{b}(i) = \text{bvec}\left\{\mathbf{R}_\mathbf{v}\mathbb{E}\left[\mathbf{D}^2(i)\right]\boldsymbol{\Lambda}\right\}$, $\mathbf{R}_\mathbf{v} = \boldsymbol{\Lambda}_\mathbf{v}\odot\mathbf{I}_M$, $\boldsymbol{\Lambda}_\mathbf{v}$ is a diagonal noise variance matrix for the network and $\mathbf{A} = \text{diag}\left\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N\right\}$ [10], with each matrix $\mathbf{A}_k$ defined as

$$\begin{aligned}\mathbf{A}_k = \text{diag}&\left\{\boldsymbol{\Lambda}_1\otimes\boldsymbol{\Lambda}_k, \ldots, \boldsymbol{\lambda}_k\boldsymbol{\lambda}_k^T\right.\\ &\left.+2\boldsymbol{\Lambda}_k\otimes\boldsymbol{\Lambda}_k, \ldots, \boldsymbol{\Lambda}_N\otimes\boldsymbol{\Lambda}_k\right\},\end{aligned} \tag{29}$$

where $\boldsymbol{\Lambda}_k$ is the diagonal eigenvalue matrix and λ_k is the corresponding eigenvalue vector for node k. Applying the bvec$\{.\}$ operator to (26) and simplifying gives

$$\text{bvec}\left\{\bar{\boldsymbol{\Sigma}}'\right\} = \bar{\boldsymbol{\sigma}}' = \mathbf{F}(i)\bar{\boldsymbol{\sigma}}, \tag{30}$$

where $\mathbf{F}(i)$ is given by (18). Thus, (14) is rewritten as

$$\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\bar{\boldsymbol{\sigma}}}\right] = \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\mathbf{F}(i)\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T(i)\bar{\boldsymbol{\sigma}}, \tag{31}$$

which characterizes the transient behavior of the network. Although not explicitly visible from (31), (18) clearly shows the effect of the VSS algorithm on the performance of the algorithm through the presence of the diagonal step-size matrix $\mathbf{D}(i)$.

Now, using (31) and (18), the analysis iterates as

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(0)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\bar{\boldsymbol{\sigma}}},\\ \mathbf{F}(0) &= \left[\mathbf{I}_{M^2N^2} - \left(\mathbf{I}_{MN}\odot\boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(0)\right]\right)\right.\\ &\quad-\left(\boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(0)\right]\odot\mathbf{I}_{MN}\right)\\ &\quad\left.+\left(\mathbb{E}\left[\mathbf{D}(0)\odot\mathbf{D}(0)\right]\right)\mathbf{A}\right]\\ &\quad.\left(\mathbf{G}^T\odot\mathbf{G}^T\right),\end{aligned}$$

where $\mathbb{E}\left[\mathbf{D}(0)\right] = \text{diag}\left\{\mu_1(0)\mathbf{I}_M, \ldots, \mu_N(0)\mathbf{I}_M\right\}$ as these are the initial step-size values. The first iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(1)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(0)\|^2_{\mathbf{F}(0)\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T(0)\bar{\boldsymbol{\sigma}}\\ &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathbf{F}(0)\bar{\boldsymbol{\sigma}}} + \mathbf{b}^T(0)\bar{\boldsymbol{\sigma}}\\ \mathbf{F}(1) &= \left[\mathbf{I}_{M^2N^2} - \left(\mathbf{I}_{MN}\odot\boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(1)\right]\right)\right.\\ &\quad-\left(\boldsymbol{\Lambda}\mathbb{E}\left[\mathbf{D}(1)\right]\odot\mathbf{I}_{MN}\right)\\ &\quad\left.+\left(\mathbb{E}\left[\mathbf{D}(1)\odot\mathbf{D}(1)\right]\right)\mathbf{A}\right]\\ &\quad.\left(\mathbf{G}^T\odot\mathbf{G}^T\right),\end{aligned}$$

where $\mathbf{b}(0) = \text{bvec}\left\{\mathbf{R}_\mathbf{v}\mathbb{E}\left[\mathbf{D}^2(0)\right]\boldsymbol{\Lambda}\right\}$, $\mathbb{E}\left[\mathbf{D}(1)\right]$ is the first step-size update. The matrix $\mathbf{F}(i)$ is updated with (18) using the ith update for the step-size matrix $\mathbb{E}\left[\mathbf{D}(i)\right]$, which is updated using the VSS approach that is being applied to the algorithm. The second iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(2)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(1)\|^2_{\mathbf{F}(1)\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T(1)\bar{\boldsymbol{\sigma}}\\ &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathbf{F}(0)\mathbf{F}(1)\bar{\boldsymbol{\sigma}}}\\ &\quad+\mathbf{b}^T(0)\mathbf{F}(1)\bar{\boldsymbol{\sigma}}\\ &\quad+\mathbf{b}^T(1)\bar{\boldsymbol{\sigma}}.\end{aligned}$$

Continuing, the third iterative update is given by

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(3)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(2)\|^2_{\mathbf{F}(2)\bar{\boldsymbol{\sigma}}}\right] + \mathbf{b}^T(2)\bar{\boldsymbol{\sigma}}\\ &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathbf{F}(0)\mathbf{F}(1)\mathbf{F}(2)\bar{\boldsymbol{\sigma}}} + \mathbf{b}^T(2)\bar{\boldsymbol{\sigma}}\\ &\quad+\mathbf{b}^T(0)\mathbf{F}(1)\mathbf{F}(2)\bar{\boldsymbol{\sigma}} + \mathbf{b}^T(1)\mathbf{F}(2)\bar{\boldsymbol{\sigma}}\\ &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(2)\mathbf{F}(2)\bar{\boldsymbol{\sigma}}} + \mathbf{b}^T(2)\bar{\boldsymbol{\sigma}}\\ &\quad+\left[\sum_{k=0}^{1}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{2}\mathbf{F}(m)\right\}\right]\boldsymbol{\sigma},\end{aligned}$$

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

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(4)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(3)\mathbf{F}(3)\bar{\boldsymbol{\sigma}}} + \mathbf{b}^T(3)\bar{\boldsymbol{\sigma}}\\ &\quad+\left[\sum_{k=0}^{2}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{3}\mathbf{F}(m)\right\}\right]\boldsymbol{\sigma},\end{aligned}$$

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

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\bar{\boldsymbol{\sigma}}}\right] &= \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(i-1)\mathbf{F}(i-1)\bar{\boldsymbol{\sigma}}} + \mathbf{b}^T(i-1)\bar{\boldsymbol{\sigma}}\\ &\quad+\left[\sum_{k=0}^{i-2}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{i-1}\mathbf{F}(m)\right\}\right]\boldsymbol{\sigma}.\end{aligned} \tag{32}$$

where $\mathcal{A}(i-1) = \mathcal{A}(i-2)\mathbf{F}(i-2)$. The recursion for the $(i+1)$th update is given by

$$\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\bar{\sigma}}\right] = \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(i)\mathbf{F}(i)\bar{\sigma}} + \mathbf{b}^T(i)\bar{\boldsymbol{\sigma}} + \left[\sum_{k=0}^{i-1}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{i}\mathbf{F}(m)\right\}\right]\boldsymbol{\sigma}, \tag{33}$$

where $\mathcal{A}(i) = \mathcal{A}(i-1)\mathbf{F}(i-1)$. Subtracting (32) from (33) and simplifying gives the overall recursive update equation

$$\begin{aligned}\mathbb{E}\left[\|\bar{\mathbf{w}}(i+1)\|^2_{\bar{\sigma}}\right] &= \mathbb{E}\left[\|\bar{\mathbf{w}}(i)\|^2_{\bar{\sigma}}\right] \\ &+ \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(i)\mathbf{F}(i)\bar{\sigma}} \\ &- \left\|\bar{\mathbf{w}}^{(o)}\right\|^2_{\mathcal{A}(i-1)\mathbf{F}(i-1)\bar{\sigma}} \\ &+ \mathbf{b}^T(i)\bar{\boldsymbol{\sigma}} - \mathbf{b}^T(i-1)\bar{\boldsymbol{\sigma}} \\ &+ \left[\sum_{k=0}^{i-1}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{i}\mathbf{F}(m)\right\}\right]\bar{\boldsymbol{\sigma}} \\ &- \left[\sum_{k=0}^{i-2}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{i-1}\mathbf{F}(m)\right\}\right]\bar{\boldsymbol{\sigma}},\end{aligned} \tag{34}$$

Simplifying (34) and rearranging the terms gives the final recursive update equation (17), where

$$\begin{aligned}\mathcal{B}(i) = &\sum_{k=0}^{i-2}\left\{\mathbf{b}^T(k)\prod_{m=k+1}^{i-1}\mathbf{F}(m)\right\} \\ &+ \mathbf{b}^T(i-1)\mathbf{I}_{M^2N^2}.\end{aligned} \tag{35}$$

The final set of iterative equations for the mean-square learning curve are given by (17), (18), (19) and (20).

References

1. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805.
2. Stankovic, J. (2014). Research directions for the Internet of Things. *IEEE Internet Things Journal*, *1*(1), 3–9.
3. Ejaz, W., & Ibnkahla, M. (2015). Machine-to-machine communications in cognitive cellular systems. In *Proceedings of 15th IEEE international conference on ubiquitous wireless broadband (ICUWB)*, Montreal, Canada (pp. 1–5).
4. Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., et al. (2016). Internet of Things in the 5G era: Enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communicatios*, *34*(3), 510–527.
5. ul Hasan, N., Ejaz, W., Baig, I., Zghaibeh, M., Anpalagan, A. (2016). QoS-aware channel assignment for IoT-enabled smart building in 5G systems. In *Proceedings of 8th IEEE international conference on ubiquitous and future networks (ICUFN)*, Vienna, Austria (pp. 924–928).
6. Ejaz, W., Naeem, M., Basharat, M., Anpalagan, A., & Kandeepan, S. (2016). Efficient wireless power transfer in software-defined wireless sensor networks. *IEEE Sensors Journal*, *16*(20), 7409–7420.
7. Culler, D., Estrin, D., & Srivastava, M. (2004). Overview of sensor networks. *IEEE Computer*, *37*(8), 41–49.
8. Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.
9. Lopes, C. G., & Sayed, A. H. (2007). Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, *55*(8), 4064–4077.
10. Lopes, C. G., & Sayed, A. H. (2008). Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, *56*(7), 3122–3136.
11. Schizas, I. D., Mateos, G., & Giannakis, G. B. (2009). Distributed LMS for consensus-based in-network adaptive processing. *IEEE Transactions on Signal Processing*, *57*(6), 2365–2382.
12. Cattivelli, F., & Sayed, A. H. (2010). Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, *58*(3), 1035–1048.
13. Bin Saeed, M. O., Zerguine, A., Zummo, S. A. (2010). Variable step-size least mean square algorithms over adaptive networks. In *Proceedings of 10th international conference on information sciences signal processing and their applications (ISSPA)*, Kuala Lumpur, Malaysia (pp. 381–384).
14. Bin Saeed, M. O., Zerguine, A., & Zummo, S. A. (2013). A variable step-size strategy for distributed estimation over adaptive networks. *EURASIP Journal on Advances in Signal Processing*, *2013*(2013), 135.
15. Almohammedi, A. & Deriche, M. (2015). Variable step-size transform domain ILMS and DLMS algorithms with system identification over adaptive networks. In *Proceedings of IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, Amman, Jordan (pp. 1–6).
16. Bin Saeed, M. O., Zerguine, A., Sohail, M. S., Rehman, S., Ejaz, W., & Anpalagan, A. (2016). A variable step-size strategy for distributed estimation of compressible systems in wireless sensor networks. In *Proceedings of IEEE CAMAD*, Toronto, Canada (pp. 1–5).
17. Bin Saeed, M. O., & Zerguine, A. (2011). A new variable step-size strategy for adaptive networks. In *Proceedings of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, Pacific Grove, CA (pp. 312–315).
18. Bin Saeed, M. O., Zerguine, A., & Zummo, S. A. (2013). A noise-constrained algorithm for estimation over distributed networks. *International Journal of Adaptive Control and Signal Processing*, *27*(10), 827–845.
19. Ghazanfari-Rad, S., & Labeau, F. (2014). Optimal variable step-size diffusion LMS algorithms. In *Proceedings of IEEE workshop on statistical signal processing (SSP)*, Gold Coast, VIC (pp. 464–467).
20. Jung, S. M., Sea, J.-H., & Park, P. G. (2015). A variable step-size diffusion normalized least-mean-square algorithm with a combination method based on mean-square deviation. *Circuits, Systems, and Signal Process.*, *34*(10), 3291–3304.
21. Lee, H. S., Kim, S. E., Lee, J. W., & Song, W. J. (2015). A variable step-size diffusion LMS algorithm for distributed estimation. *IEEE Transactions on Signal Processing*, *63*(7), 1808–1820.
22. Sayed, A. H. (2014). Adaptive networks. *Proceedings of the IEEE*, *102*(4), 460–497.
23. Sayed, A. H. (2003). *Fundamentals of adaptive filtering*. New York: Wiley.

24. Nagumo, J., & Noda, A. (1967). A learning method for system identification. *IEEE Transactions on Automatic Control*, *12*(3), 282–287.
25. Kwong, R. H., & Johnston, E. W. (1992). A variable step-size LMS algorithm. *IEEE Transactions on Signal Processing*, *40*(7), 1633–1642.
26. Aboulnasr, T., & Mayyas, K. (1997). A robust variable step size LMS-type algorithm: Analysis and simulations. *IEEE Transactions on Signal Processing*, *45*(3), 631–639.
27. Costa, M. H., & Bermudez, J. C. M. (2008). A noise resilient variable step-size LMS algorithm. *Signal Processing*, *88*(3), 733–748.
28. Wei, Y., Gelfand, S. B., & Krogmeier, J. V. (2001). Noise-constrained least mean squares algorithm. *IEEE Transactions on Signal Processing*, *49*(9), 1961–1970.
29. Sulyman, A. I., & Zerguine, A. (2003). Convergence and steady-state analysis of a variable step-size NLMS algorithm. *Signal Processing*, *83*(6), 1255–1273.
30. Bin Saeed, M. O., & Zerguine, A. (2013). A variable step-size strategy for sparse system identification. In *Proceedings of 10th international multi-conference on systems, signals & devices (SSD)*, Hammamet (pp. 1–4).
31. Al-Naffouri, T. Y., & Moinuddin, M. (2010). Exact performance analysis of the ϵ-NLMS algorithm for colored circular Gaussian inputs. *IEEE Transactions on Signal Processing*, *58*(10), 5080–5090.
32. Al-Naffouri, T. Y., Moinuddin, M., & Sohail, M. S. (2011). Mean weight behavior of the NLMS algorithm for correlated Gaussian inputs. *IEEE Signal Processing Letters*, *18*(1), 7–10.
33. Bin Saeed, M. O. (2017). LMS-based variable step-size algorithms: A unified analysis approach. *Arabian Journal of Science & Engineering*, *42*(7), 2809–2816.
34. Koning, R. H., Neudecker, H., & Wansbeek, T. (1990). Block Kronecker products and the vecb operator. Economics Deptartment, Institute of Economics Research, Univ. of Groningen, Groningen, The Netherlands, Research Memo. No. 351.

Muhammad Omer Bin Saeed received the B.E. and M.S. degrees from National University of Sciences & Technology (NUST), Islamabad, Pakistan, in 2003 and 2005, respectively, and the Ph.D. degree from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia, in 2011, all in electrical engineering. He joined the Department of Computer & Software Engineering at the College of E&ME, NUST, in May 2016, where he is presently an Assistant Professor, working in the area of Signal Processing and Communications. His research interests include signal processing, adaptive filtering, channel coding theory and digital communications.

Waleed Ejaz (S'12–M'14–SM'16) received the B.Sc. degree in computer engineering from the University of Engineering and Technology at Taxila, Pakistan, the M.Sc. degree in computer engineering from the National University of Sciences and Technology, Islamabad, Pakistan, and the Ph.D. degree in information and communication engineering from Sejong University, Seoul, South Korea, in 2014. Since 2014, he held academic and research positions with Carleton University, Ottawa, ON, Canada, Ryerson University, Toronto, ON, and Queen's University, Kingston, ON. He has completed certificate courses on teaching and learning in higher education from the Chang School, Ryerson University. His current research interests include Internet of Things, energy harvesting, 5G cellular networks, and mobile cloud computing. Dr. Ejaz's professional activities include being an Associate Editor of IEEE Communications Magazine, the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING, and IEEE ACCESS. In addition, he is handling several special issues in prestigious journals, as well as in organizing committees of several major IEEE conferences and workshops. He is a Registered Professional Engineer in the Province of Ontario, Canada.

Saad Rehman received the M.S. and Ph.D. degrees from the University of Sussex, Brighton, U.K. He is currently an Associate Head of Department with the College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad. His research interests include correlation pattern recognition, and image and signal processing.

Azzedine Zerguine received the B.Sc. degree from Case Western Reserve University, Cleveland, OH, in 1981, the M.Sc. degree from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1990, and the Ph.D. degree from Loughborough University, Loughborough, U.K., in 1996, all in electrical engineering. From 1981 to 1987, he was working with different Algerian state-owned companies. From 1987 to 1990, he was a Research and Teaching Assistant, Electrical Engineering Department, KFUPM. In 1990, he joined the Physics Department at KFUPM as a Lecturer, and then from 1997 to 1998 as an Assistant Professor. In 1998, he joined the Department of Electrical Engineering at KFUPM where he is presently a Professor working in the area of Signal Processing and Communications. His research interests include signal processing, adaptive filtering, neural networks, and digital communications.

Alagan Anpalagan (S'98–M'01–SM'04) is currently a Professor with the Department of Electrical and Computer Engineering, Ryerson University, where he directs a Research Group working on radio resource management and radio access and networking areas within the WINCORE Laboratory. He served as an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIAL (2012–2014), the IEEE COMMUNICATIONS LETTERS (2010–2013), the *Springer Wireless Personal Communications* (2011–2013), and the *EURASIP Journal of Wireless Communications and Networking* (2004–2009). He co-authored three edited books, *Design and Deployment of Small Cell Networks*, (Cambridge University Press, 2014), *Routing in Opportunistic Networks*, (Springer, 2013), *Handbook on Green Information and Communication Systems*, (Academic Press 2012). He received B.A.Sc., M.A.Sc., and Ph.D. degrees from the University of Toronto all in electrical engineering. He is currently a registered Professional Engineer in the province of Ontario, Canada. He is a Fellow of the Institution of Engineering and Technology. He was the Vic-Chair of the IEEE Canada Central Area Chair (2012–2014), the IEEE Toronto Section Chair (2006–2007), the ComSoc Toronto Chapter Chair (2004 -2005), and the IEEE Canada Professional Activities Committee Chair (2009–2011). He currently serves as the TPC Vice-Chair of the IEEE VTC 2017, and served as the TPC Co-Chair of the IEEE Globecom'15 on SAC Green Communication and Computing, the IEEE WPMC'12 Wireless Networks, the IEEE PIMRC'11 Cognitive Radio and Spectrum Management, and the IEEE CCECE'04/08. He serves as the Vice Chair of the IEEE SIG on Green and Sustainable Networking and Computing with Cognition and Cooperation. He was a recipient of the Dean's Teaching Award in 2011, the Faculty Scholastic, Research and Creativity Award in 2010, 2014, and 2017, the Faculty Service Award in 2011and 2013 in Ryerson University. He also received the Exemplary Editor Award from the IEEE ComSoc in 2013 and the Editor-in-Chief Top10 Choice Award in Transactions on Emerging Telecommunications Technology in 2012.

Houbing Song received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in August 2012. In August 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). He served on the faculty of West Virginia University from August 2012 to August 2017. In 2007 he was an Engineering Research Associate with the Texas A&M Transportation Institute. He serves as an Associate Technical Editor for IEEE Communications Magazine. He is the editor of four books, including Smart Cities: Foundations, Principles and Applications, Hoboken, NJ: Wiley, 2017, Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications, Chichester, UK: Wiley-IEEE Press, 2017, Cyber-Physical Systems: Foundations, Principles and Applications, Boston, MA: Academic Press, 2016, and Industrial Internet of Things: Cybermanufacturing Systems, Cham, Switzerland: Springer, 2016. He is the author of more than 100 articles. His research interests include cyber-physical systems, cybersecurity and privacy, internet of things, edge computing, big data analytics, unmanned aircraft systems, connected vehicle, smart and connected health, and wireless communications and networking. Dr. Song is a senior member of both IEEE and ACM. Dr. Song was the very first recipient of the Golden Bear Scholar Award, the highest campus-wide recognition for research excellence at West Virginia University Institute of Technology (WVU Tech), in 2016.