Noname manuscript No.
(will be inserted by the editor)

On the Analysis of the Incremental ℓ_0-LMS Algorithm for Distributed Systems

Robson A. do Prado · Raphael M. Guedes · Felipe da R. Henriques · Felipe M. da Costa · Luís D. T. J. Tarrataca · Diego B. Haddad

Received: date / Accepted: date

Abstract Adaptive filtering algorithms implement an estimation of a set of parameters. Frequently, the system to be identified is sparse, in the sense that most of its energy is concentrated among a few coefficients. Adaptive algorithms, such as the ℓ_0-LMS, can incorporate this property in order to increase the convergence rate. In this work, a stochastic model is proposed to predict characteristics of transient, steady-state and tracking of the ℓ_0-LMS algorithm, implemented in a distributed way (*i.e.*, with the incremental strategy). Such a diffuse strategy is adequate in situations where the network energy is severely limited. The advanced analysis does not require neither white nor Gaussian input signals in order to predict the learning capabilities of the ℓ_0-LMS algorithm.

Keywords Adaptive Filtering · Adaptive Networks · Norm Regularization

Robson A. do Prado
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, RJ, Brazil
E-mail: robson_prado@hotmail.com

Raphael M. Guedes
Universidade Estácio de Sá, RJ, Brasil
E-mail: raphael.guedes@estacio.br

Felipe da R. Henriques
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Petrópolis-RJ, Brazil
E-mail: felipe.henriques@cefet-rj.br

Felipe M. da Costa
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Petrópolis-RJ, Brazil
E-mail: felmacedo@gmail.com

Luís D. T. J. Tarrataca
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Petrópolis-RJ, Brazil
E-mail: luis.tarrataca@cefet-rj.br

Diego B. Haddad
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Petrópolis-RJ, Brazil
E-mail: diego.haddad@cefet-rj.br

1 Introduction

Adaptive filter algorithms update the coefficients $w_i(k)$ ($i \in \{0, 1, \ldots, N-1\}$) in a recursive way. The new values of these coefficients are obtained by means of a correction term:

$$\begin{pmatrix} \text{new parameter} \\ \text{values} \end{pmatrix} = \begin{pmatrix} \text{old parameter} \\ \text{values} \end{pmatrix} + \begin{pmatrix} \text{correction} \\ \text{term} \end{pmatrix}, \tag{1}$$

in which the correction term, in supervised contexts, depends on a reference signal and an error signal, that roughly estimates (in a stochastic manner) the discrepancy among the adaptive filter and its ideal values [55].

In practice, several transfer functions present energy concentration in a few coefficients [51, 50]. This information allows for adaptive algorithms to perform a faster identification of the desired transfer function, which tends to be of great interest for researchers [50]. The advent of electronic circuit miniaturization, alongside the development of robust communication protocols, has given rise to a set of applications (such as sensor networks [2]) in which a collection of agents connected according to a certain topology can interact dynamically with each other. These interactions eventually created the *distributed adaptive filtering* area, which was notable for outperforming consensus strategies in terms of stability, convergence rate and tracking ability [23]. It is worth mentioning that centralized solutions can result in higher energy usage and increased communication resources. The existence of a critical point in the fusion center is also a disadvantage. Namely, this results in less network autonomy and is not scalable for large networks [32, 37]. Distributed adaptive filtering raises the need for modeling characteristics related to information aggregation, processing and diffusion across graphs, which are capable of modeling the topology and neighborhoods of agent networks. Amongst the applications in which diffuse adaptive networks are considered an efficient solution, it is worth noting: modeling of complex behaviors exhibited by socioeconomic or biological networks [23], targeting and tracking [57], environmental monitoring [56], spectral sensing in mobile networks [42] and distributed optimization problems [56]. Reaching consensus among agents is critical for successful inference on these issues [13].

1.1 Motivation

With severe energy and resource constraints on communications, cooperative incremental strategies for the linear estimation problem are preferable to diffuse schemes [32]. In incremental implementations, agents share data cyclically, with only one node at a time communicating with their immediate neighbor (to limit the waste of energy) [41]. This type of implementations does not require an intermediate aggregation step, which reduces the demand for computational resources [37]. The resulting algorithm is able to exploit the spatial dimension and respond in real time to environmental changes. It also imbues

the method with a distributed and cooperative behavior. As with classical adaptive filtering algorithms, no knowledge of data statistics is required for incremental strategies to work properly. The sequential flow of incremental techniques allows the distributed solution to respond to new data.

Employing an adaptive algorithm requires theoretical safeguards in order to guarantee its performance characteristics concerning, namely: (i) convergence rate; (ii) steady-state performance; (iii) upper limit for the learning factor to avoid divergence; and (iv) tracking ability and evolution of performance metrics (such as MSE and MSD) over long iterations. Moreover, performance analysis is a crucial task since: (i) it provides useful guidelines for adaptive filter design [49]; (ii) supplies compromises between bias and variance [11]; (iii) predicts the impact that the eigenvalue dispersion of the input signal autocorrelation matrix has on learning [12]; and (iv) reveals effects of a finite arithmetic precision [47].

1.2 Objectives and Contributions

This paper performs steady-state, transient and tracking analyses for the ℓ_0-LMS algorithm, in its incremental version. Therefore, a new stochastic model that predicts the learning capabilities of the considered adaptive network is proposed. We derive recursive equations aiming at predicting the MSD over the number of iterations. The proposed stochastic model is capable of estimating the transient behavior of the considered incremental algorithm (l_0-LMS). Moreover, the model allows the evaluation of the tracking capabilities of distributed adaptive filtering algorithms. The performance of the model was evaluated by performing a comparison between theoretical and simulation results, considering metrics such as the MSD and MSE. The results show the capabilities of the analysis to predict both transient and steady-state behaviors. Furthermore, the observed simulation data closely matches the theoretical performance.

1.3 Notation

This paper considers the following notation:

- **Scalars**: lower case letters (without bold) are used, such as x.
- **Vectors**: lower case letters (in bold) are used, such as $\boldsymbol{x}$. All vectors are column. Thus, $\boldsymbol{x}^T$ is a row vector, since $(\cdot)^T$ is the transpose operator.
- **Matrices**: upper case letters are used (in bold), such as $\boldsymbol{R}$.
- **Operators**: Statistical average operator is denoted by $\mathbb{E}[\cdot]$.

1.4 Structure of the Paper

The remainder of this work is organized as follows: Section 2 describes approaches for *sparsity-aware* algorithms. Section 3 details the ℓ_0-LMS algorithm

this paper focuses on. Section 4 presents some incremental implementations of distributed adaptive filtering algorithms. The proposed analyses for the l_0-LMS algorithm in its incremental format are presented in Section 5. Theoretical and computational results are discussed in Section 6. Section 7 lists our concluding remarks.

2 Sparse Systems Identification

In practice, many systems are sparse [31,9]. As a result, researchers verified that addressing sparsity in the design stage of algorithms [38] can lead to several advantages. For example, in problems such as the Compressive Sensing [8] this strategy resulted in a cost- and complexity- reduction of the data gathering process [54]. The remainder of this section presents, succinctly, some popular approaches for sparsity-aware adaptive filtering algorithms.

Our work assumes an adaptive transversal structure, with $x(k)$ denoting the input signal and N the number of elements of the adaptive filter. Furthermore

$$y(k) \triangleq \boldsymbol{w}^T(k)\boldsymbol{x}(k) \tag{2}$$

is the filter output at the k-th iteration, where

$$\boldsymbol{w}(k) \triangleq \left[w_0(k)\ w_1(k)\ \ldots\ w_{N-1}(k)\right]^T, \tag{3}$$

and

$$\boldsymbol{x}(k) \triangleq \left[x(k)\ x(k-1)\ \ldots\ x(k-N+1)\right]^T. \tag{4}$$

The work described in [21] proposed the PNLMS algorithm in the adaptive filtering context. The method is a a particular case of the adaptive algorithm of variable metric parallel projection [65] and gave rise the *proportionate adaptation* paradigm. In this strategy, coefficients with higher magnitudes receive larger update energy. This energy can be interpreted as the effect of a manager of scarce resources, which distributes the learning step and expedites the identification process [27]. Other algorithms which consider this paradigm are: PNLMS++ [25], IPNLMS [10], MPNLMS [18] and IMPNLMS [24].

Most of the proportionate algorithms can be derived from the following optimization problem [27]:

$$\min_{\boldsymbol{w}(k+1)} \|\boldsymbol{w}(k+1) - \boldsymbol{w}(k)\|^2_{\boldsymbol{\Lambda}_k^{-1}}$$

$$\text{subject to } e_p(k) = \left(1 - \beta \frac{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Lambda}_k}}{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Lambda}_k} + \delta}\right) e(k), \tag{5}$$

where $\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Lambda}_k^{-1}} \triangleq \boldsymbol{x}^T(k)\boldsymbol{\Lambda}_k^{-1}\boldsymbol{x}(k)$ is the norm of the vector $\boldsymbol{x}(k)$ weighted by $\boldsymbol{\Lambda}_k^{-1}$.

The solution for (5), obtained by means of the Lagrange multipliers technique, consists in:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta \frac{\boldsymbol{\Lambda}_k \boldsymbol{x}(k) e(k)}{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Lambda}_k} + \delta}, \tag{6}$$

where $\boldsymbol{\Lambda}_k$ is a diagonal matrix, whose main diagonal elements are chosen differently for distinct proportionate algorithms and δ is a regularization factor. Commonly, the sum of these elements is unitary; in this case, [28] suggests that the steady–state MSE is identical to the MSE of the NLMS algorithm (keeping the β parameter equal). Thus, the main advantage of the proportionate paradigm is the increased convergence rate. Recently, [33] proposed a more sophisticated prediction model for the steady–state MSE of these algorithms.

Another research line is inspired by the IPAPA algorithm [30] (*Improved Proportionate Affine Projection Algorithm*), which inherits the desirable convergence properties of the affine projection algorithm. However, affine projection based algorithms often require matrix inversions (which are avoided in some variants [66,6,5]), that can be ill posed. Commonly, one performs the matrices regularization to be inverted by adding a positive constant in their main diagonals. The optimal choice for this regularization parameter is dependent on the noise variation. Because of this reason, there are proposals for the adaptive choice of this parameter [48].

Although proportionate adaptation is still a very popular paradigm, sparsity–aware adaptive filtering algorithms that were proposed in the last years tend to employ some sparsity regularization, regarding the ℓ_1 or ℓ_0 norms.

The zero-attracting LMS algorithm (ZA–LMS) is one of the most popular adaptive algorithms with sparsity regularization, which employs the ℓ_1 norm. Transient analyses of ZA–LMS can be seen in [60,14]. The ℓ_1 norm is commonly employed as substitute for the ℓ_0 norm, which is non–convex [14]. The ZA–LMS algorithm can be obtained by the (stochastic gradient) minimization of the following cost function:

$$\mathcal{F}_{\ell_1}[\boldsymbol{w}(k+1)] = e^2(k) + \gamma \|\boldsymbol{w}(k)\|_1, \tag{7}$$

which generates the following update equation:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta e(k) \boldsymbol{x}(k) - \kappa \boldsymbol{s}(k), \tag{8}$$

where:

$$\kappa \triangleq \beta\gamma$$

and

$$\boldsymbol{s}(k) \triangleq \left[\text{sign}[w_0(k)]\ \text{sign}[w_1(k)]\ \ldots\ \text{sign}[w_{N-1}(k)]\right]^T$$

and $\text{sign}(\cdot)$ being the signal function, defined by:

$$\text{sign}(x) = \begin{cases} 1, & \text{for } x > 0 \\ 0, & \text{for } x = 0 \\ -1, & \text{for } x < 0 \end{cases}. \tag{9}$$

3 The ℓ_0-LMS Algorithm

Unfortunately, the zero–attraction term ($-\kappa \boldsymbol{s}(k)$) of the ZA–LMS updating equation (8) intends to shrink to zero both the high magnitude coefficients (those having values not close to zero) and the low magnitude coefficients (those, as presumed by the sparsity assumption, close to zero). This property of the ZA–LMS algorithm tends to degenerate its steady–state performance, since there is a bias for the high magnitude coefficients estimates. The ℓ_0–LMS and ℓ_0–NLMS, proposed in [26], have received great attention [61,63,62,52], because they do not have the aforementioned disadvantage. Of these, the first (ℓ_0–LMS) can be derived by means of the stochastic gradient technique from an approximation of the following cost function:

$$\mathcal{F}_{\ell_0}[\boldsymbol{w}(k+1)] = e^2(k) + \gamma \|\boldsymbol{w}(k)\|_0, \tag{10}$$

where $\|\boldsymbol{w}(k)\|_0$ denotes the ℓ_0 norm (a pseudo–norm), which reflects the number of non–zero elements of $\boldsymbol{w}(k)$ and γ is a control parameter that penalizes solutions that are not sparse. However, the ℓ_0 norm is not derivable, and its minimization leads to a NP–hard problem [26]. Thus, the most common approach to minimize (10) is to approximate the ℓ_0 norm by a derivable function. Among several employed approximations, the most popular is [26]:

$$\|\boldsymbol{w}(k)\|_0 \approx \sum_{i=0}^{N} F_\rho[w_i(k)] = \sum_{i=0}^{N-1} \left(1 - e^{-\rho|w_i(k)|}\right), \tag{11}$$

where $\rho \in \mathbb{R}_+$ is an adjustable parameter. From (11), the cost function in (10) can be approximated by:

$$\mathcal{F}_{\ell_0}[\boldsymbol{w}(k+1)] \approx e^2(k) + \gamma \sum_{i=0}^{N-1} \left(1 - e^{-\rho|w_i(k)|}\right), \tag{12}$$

which, at the stochastic gradient minimization paradigm, give rises to the following algorithm (in scalar terms):

$$w_i(k+1) = w_i(k) + \beta e(k) x(k-i) - \kappa \text{sign}\,[w_i(k)]\, e^{-\rho|w_i(k)|}, \tag{13}$$

for $0 \leq i \leq N$, where $\kappa \triangleq \beta\gamma$.

Equation (13) refers to an algorithm that is not normalized by the energy of the input signal. A possible normalized update equation could be:

$$w_i(k+1) = w_i(k) + \beta \frac{e(k)x(k-i)}{\|\boldsymbol{x}(k)\|^2} - \kappa \text{sign}\,[w_i(k)]\, e^{-\rho|w_i(k)|}, \tag{14}$$

which has the advantage of having upper bounds for β that are independent on the statistics of the input signal [55].

Although the equations (13) and (14) allow the implementation of sparsity–aware algorithms, they usually are deprecated by computationally simpler versions, that try to avoid the exponential term $e^{-\rho|w_i(k)|}$, which is replaced by

simpler linear approximations. Expanding the exponential in a first order Taylor series around zero, leads to [26]:

$$e^{-\rho|w_i(k)|} \approx \begin{cases} 1-\rho|w_i(k)|, & |w_i(k)| \leq \frac{1}{\rho} \\ 0, & \text{for the rest} \end{cases}, \tag{15}$$

and the update equations for the ℓ_0-LMS and ℓ_0-NLMS can be described by:

$$w_i(k+1) = w_i(k) + \tilde{\beta}(k)e(k)x(k-i) + \kappa f_\rho\left[w_i(k)\right], \tag{16}$$

where $f_\rho\left[w_i(k)\right]$ is defined as:

$$f_\rho\left[w_i(k)\right] = \begin{cases} \rho^2 w_i(k) + \rho, & -\frac{1}{\rho} \leq w_i(k) < 0 \\ \rho^2 w_i(k) - \rho, & 0 < w_i(k) \leq \frac{1}{\rho} \\ 0, & \text{for the rest} \end{cases}, \tag{17}$$

and

$$\tilde{\beta}(k) = \begin{cases} \beta, & \text{for } \ell_0\text{-LMS}, \\ \frac{\beta}{\boldsymbol{x}^T(k)\boldsymbol{x}(k)+\delta}, & \text{for } \ell_0\text{-NLMS} \end{cases}. \tag{18}$$

Reference [15] compared both proportionate and regularization approaches. It concluded that the regularization method has a better trade-off between convergence rate and steady-state performance than the proportionate approach. As a result, our work intends to evaluate the transient and the tracking of algorithms that employ norm regularization, specifically the ℓ_0-LMS, in a distributed implementation. The next section describes the distributed adaptive estimation scenario this paper is interested in.

4 Distributed Adaptive Filtering in the Incremental Modality

Consider a connected network composed of M local agents with some computational capacity. Their interconnections can be modeled by graphs, in which vertices are the agents and the arcs represent the information exchange capacity between any two agents. The neighborhood $\mathcal{N}_l$ of the l-th agent is the set of agents that are connected to it by arcs; Suppose that this set always contains the l-th agent. An undirected graph is assumed to be used. Accordingly, if the l-th agent is a neighbor of the m-th agent then this implies that the m-th agent is also a neighbor of the l-th one [59].

In a distributed context, it is up to the network of M agents to estimate a parameter vector $\boldsymbol{w}^\star \in \mathbb{R}^N$ in order to minimize an objective function of the network, such as [59]:

$$\min_{\boldsymbol{w}} \sum_{l=1}^{M} J_l(\boldsymbol{w}), \tag{19}$$

which associates with the l-th agent a cost $J_l(\boldsymbol{w}) \in \mathbb{R}$. We will assume that all costs $\{J_l(\boldsymbol{w})\}$ $(l = 1, 2, \ldots, M)$ reach their minimum when their argument is $\boldsymbol{w}^\star$; that is, the minimum is the same for the various agents. In practice, this is

a common situation whilst also sufficiently accommodating for the main ideas of distributed adaptive filtering [59].

We are interested in how to perform this estimation process in a distributed manner, where multiple agents, connected by a topology, aim to estimate an optimal parameter vector. Such an architecture, relying solely on local interactions, disregards sink nodes, which makes network processing more reliable and with added resilience to nodes and links whilst also being scalable and more resource efficient [4].

In this paper, we will assume that the l-th agent has access to a moving-average model:

$$d_l(k) = \sum_{n=0}^{N-1} w^\star(n) x_l(k-n) + \nu_l(k) = \left(\boldsymbol{w}^\star\right)^T \boldsymbol{x}_l(k) + \nu_l(k), \tag{20}$$

where $\boldsymbol{x}_i(k) \triangleq \left[x_l(k)\ x_l(k-1)\ \ldots\ x_l(k-N+1)\right]$ contains N consecutive samples of the input signal for the l-th node[1] and $\nu_l(k)$ represents the k-th noise sample (possibly from measurement) associated with the l-th agent.

Reference [59] provides an example of model (20) in a situation where M agents intend to estimate the coefficients of a communication channel. If we assume that agents are able to independently test the unknown model - and observe its response to their respective excitements - then the system dynamics for each agent matches the formulation of (20), which also models the presence of additive-specific noise for each node.

For generality purposes, we will not assume that the statistical properties of input signals and noise are constant throughout the nodes. Thus, the noise variance of one node may be greater than that of another. This will enable modeling of cases where measurements at some nodes are noisier than at others.

The derivation of the incremental modality is presented next (as described in [40]). Consider the minimization problem of (19). Then, assuming a stationary environment and minimizing the mean square error, each $J_l(\boldsymbol{w})$ can be written as:

$$J_l(\boldsymbol{w}) \triangleq \mathbb{E}\left\{\left[d_l(k) - \boldsymbol{w}^T \boldsymbol{x}_l(k)\right]^2\right\} \tag{21}$$

$$= \sigma_{d,l}^2 - 2\boldsymbol{R}_{dx,l}\boldsymbol{w} + \boldsymbol{w}^T \boldsymbol{R}_{x,l}\boldsymbol{w}, \tag{22}$$

in which second order statistics are defined by:

$$\sigma_{d,l}^2 \triangleq \mathbb{E}\left[d_l^2(k)\right], \tag{23}$$

$$\boldsymbol{R}_{x,l} \triangleq \mathbb{E}\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right], \tag{24}$$

$$\boldsymbol{R}_{dx,l} \triangleq \mathbb{E}\left[d_l(k)\boldsymbol{x}_l(k)\right], \tag{25}$$

[1] In this paper we use the terms node and agent indistinctly.

where we assume that the reference signals have zero mean on all nodes. Thus, the minimization of (19) by the *steepest-descent* method occurs by:

$$\begin{aligned}\boldsymbol{w}(k+1) &= \boldsymbol{w}(k) - \beta' \left[\nabla J(\boldsymbol{w}(k))\right], \\ &= \boldsymbol{w}(k) - \beta' \sum_{l=1}^{M} \left[\nabla J_l(\boldsymbol{w}(k))\right], \\ &= \boldsymbol{w}(k) + \overbrace{2\beta'}^{=\beta} \sum_{l=1}^{M} \left(\boldsymbol{R}_{dx,l} - \boldsymbol{R}_{x,l}\boldsymbol{w}(k)\right) \\ &= \boldsymbol{w}(k) + \beta \sum_{l=1}^{M} \left(\boldsymbol{R}_{dx,l} - \boldsymbol{R}_{x,l}\boldsymbol{w}(k)\right), \end{aligned} \tag{26}$$

where $\beta \in \mathbb{R}_+$ is a learning step defined by the project designer and $\nabla J(\boldsymbol{w}(k))$ is the gradient vector of $J(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ evaluated at the point $\boldsymbol{w} = \boldsymbol{w}(k)$. The optimization method in (26) is not incremental, as this technique requires that each node has access to only its immediate neighbor in the same *cycle*. Let $\boldsymbol{\psi}_l(k)$ be a *local estimate* of $\boldsymbol{w}^\star$ at node l at time k. Assume that the l-th node has access to $\boldsymbol{\psi}_{l-1}(k)$ (that is, the estimate of $\boldsymbol{w}^\star$ from its immediately neighbour node in the cycle). Therefore, an *incremental gradient* algorithm must calculate the required gradient $\nabla J_l(\cdot)$ in the local estimate $\boldsymbol{\psi}_{l-1}(k)$ obtained from the index node $l-1$. Thus, we have a cooperative scheme and an *incremental* distributed solution of the *steepest-descent* algorithm (26) defined by:

$$\begin{cases} \boldsymbol{\psi}_0(k) &= \boldsymbol{w}(k), \\ \boldsymbol{\psi}_l(k) &= \boldsymbol{\psi}_{l-1}(k) - \beta_l \left[\nabla J_l(\boldsymbol{\psi})\right], l = 1, \ldots, M \\ \boldsymbol{w}(k+1) &= \boldsymbol{\psi}_M(k). \end{cases} \tag{27}$$

Although solution (27) relies only on locally available information, it requires the knowledge of the second order moments $\boldsymbol{R}_{dx,l}$, $\boldsymbol{R}_{x,l}$, necessary for the calculation of local gradients $\nabla J_l(\boldsymbol{\psi})$. By substituting these statistics for their stochastic approximations (*i.e.*, $\boldsymbol{R}_{dx,l} \approx d_l(k)\boldsymbol{u}_l(k)$ and $\boldsymbol{R}_{x,l} \approx \boldsymbol{u}_l(k)\boldsymbol{u}_l^T(k)$), the *distributed incremental LMS* algorithm is obtained:

$$\begin{cases} \boldsymbol{\psi}_l(k) &= \boldsymbol{\psi}_{l-1}(k) + \beta_k \boldsymbol{x}_l(k) \left[d_l(k) - \boldsymbol{\psi}_{l-1}^T(k)\boldsymbol{x}_l(k)\right], \\ & \qquad\qquad l = 1, \ldots, M \\ \boldsymbol{w}(k+1) = \boldsymbol{\psi}_M(k) \end{cases} \tag{28}$$

whose operation is described in Figure 1. A sparsity-aware variant (*i.e.*, the ℓ_0-LMS incremental algorithm) of algorithm (28) will be the focus of this paper. The analysis proposed in the following section is not simple, as each node cooperates with an adjacent node to explore the spatial dimension while performing local calculations on the temporal dimension [37]. As shall be seen, each node converges to different MSE levels, depending on data statistical diversity and the different noise levels [40].

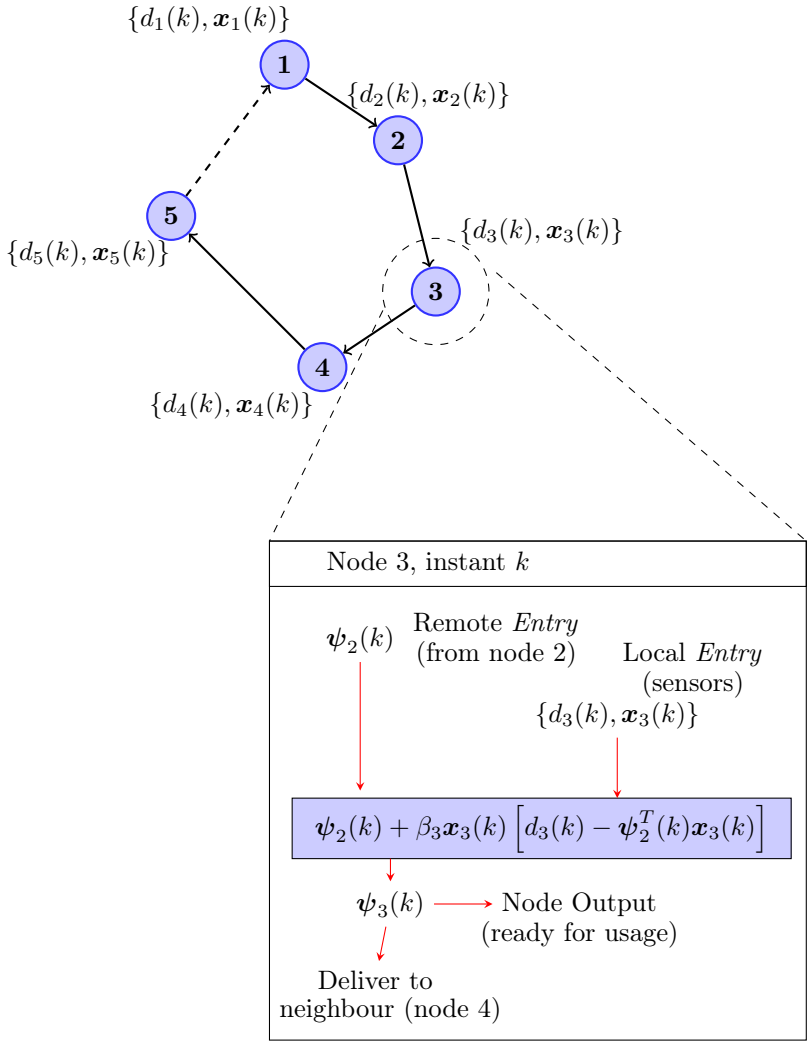

Fig. 1 Data processing in a distributed adaptive structure with incremental collaboration and M = 5 agents. Adapted from [40].

5 Analysis of the incremental ℓ_0-LMS algorithm

5.1 Main theoretical analyses

In this work, we are essentially focused in examining the incremental ℓ_0-LMS algorithm in terms of: (i) transient analysis; (ii) stability; and (iii) tracking. Doing so requires building a stochastic model capable of predicting performance throughout the iterations for each network agent. When the number of iterations is high, such a model should be able to provide to the designer an

indication of algorithmic performance in the steady-state. In order to avoid unnecessary restrictions in the theoretical predictions, each agent will have specific β, ρ and κ parameters. In addition, we will also allow that the measurement noise variance and the statistical properties of each agent to be different for each agent. This type of modelling will allow for *heterogeneous networks*, in which some nodes will employ norm-regularized algorithms, whilst others will not[2]. Configurations such as these are capable of reducing the computational cost, without necessarily implying a diminishing of the overall performance of the distributed learning [17]. Work [16] provides guidelines for the optimal choice of the nodes that should employ norm-regularization.

In the absence of noise, the analysis of adaptive algorithms usually falls in an equation using homogeneous difference, whose divergence can be analyzed. In practice, there will exist a measurement noise, which implies that in strict terms the convergence is no longer possible. For this reason, the literature usually focuses on studying: (i) convergence on the mean and (ii) second order statistics. In the former, the expected deviations between the adaptive coefficients and the optimal ones are evaluated to see if they approach zero when the number of iterations approaches infinity. In the latter, the elements of the autocorrelation matrix of the deviations are evaluated to see if they are finite, in particular, in the steady-state regime [53].

Different transient analysis techniques have been proposed in the literature [19]. Most of which invokes the *independence hypothesis* **H1**, which can be stated as follows:

H1. The filters $\boldsymbol{\psi}_i(k)$ are independent of $\boldsymbol{x}_j(k)$, for $i, j \in \{1, \ldots, M\}$.

When the analysis is performed in non-distributed algorithms (the most common scenario) it is possible to replace $\boldsymbol{\psi}_i(k)$ by $\boldsymbol{w}(k)$ and $\boldsymbol{x}_j(k)$ by $\boldsymbol{x}(k)$. It is also common to consider the hypothesis of *spatial independence*, which assumes statistically independent input signals in different nodes [46]. Equivalently, the independence hypothesis can also assume that $\boldsymbol{x}_i(k_1)$ is statistically independent of $\boldsymbol{x}_i(k_2)$ for whatever $k_1 \neq k_2$ and $i, j \in \{1, \ldots, M\}$. Even though this is clearly violated when the adaptive filter consists of a transversal structure. Furthermore, this hypothesis provides consistent results, in particular when the learning factor β is small [55]. Reference [44] argues that the analyzes employing **H1** consider a first-order evaluation of the adaptive algorithm. Consequently, it is implied that these analyzes predict real behaviour only when β does not assume large values [36]. It should be noted that the input signals of different nodes can present statistical correlation, depending on the chosen topology, among other factors. In this case, the proposed stochastic model can be less accurate, especially when large step sizes are adopted, which increases the stochastic coupling between the excitation data and the adaptive weights [34].

The most sophisticated transient analysis in the literature can be called of exact expectation analysis [20, 34–36], which can also predict with great accu-

[2] By setting the κ parameter of a certain node to zero, its regularization term will be automatically deactivated.

racy the convergence, stability and performance characteristics in the steady-state. Even though the main objective is to perform an "exact" analysis, sometimes divergences are observed between the experimental and the theoretical curves obtained, which was partially explained in [45]. The main issue resides in the complexity and the lack of clear intuitive results. The analysis expresses the evolution of the expected values of interest contained in state vector $\boldsymbol{y}_k$. This can be expressed through the following state equation that is time invariant:

$$\boldsymbol{y}_{k+1} = \boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{b}, \tag{29}$$

where the dimension of the square matrix $\boldsymbol{A}$ is 28181 for the simple case in which the LMS algorithm presents just six adaptive coefficients with a white input signal. Given the extreme complexity of this analysis, we opted to not undertake it in this work.

Another transient analysis technique is the flow energy approach [3]. The method does not assume that the input signals are Gaussian and is based on a fundamental relationship of energy conservation. This relationship was originally developed for the deterministic analysis of adaptive filtering algorithms robustness [58]. Additionally, the method also revealed to provide an adequate stochastic analysis of the steady-state of these algorithm [64].

Transient analysis via energy conservation also gives rise to a time-invariant state equation which is simpler than the one derived by the exact expectation analysis technique [3]. However, the application of this methodology in algorithms with sparsity regularization (the main purpose of this article) is a challenging process due to some terms of difficult analytical solution [29]. For this reason, we are not aware of any application example of this approach to algorithms with sparsity regularization (such as ℓ_0-LMS). This happens despite the fact that it is possible to apply this approach to proportional techniques.

A third technique, the most popular in the literature, consists in determining difference equations that describe in a recursive manner the evolution of the autocorrelation matrix $\boldsymbol{R}_{\tilde{\boldsymbol{\psi}}_i}(k) \triangleq \mathbb{E}\left[\tilde{\boldsymbol{\psi}}_i(k)\tilde{\boldsymbol{\psi}}_i^T(k)\right]$ from the deviation $\tilde{\boldsymbol{\psi}}_i(k)$, which can be defined as

$$\tilde{\boldsymbol{\psi}}_i(k) \triangleq \boldsymbol{w}^\star - \boldsymbol{\psi}_i(k). \tag{30}$$

Determining $\boldsymbol{R}_{\tilde{\boldsymbol{\psi}}_i}(k)$, which can consist of second order statistics, is very useful in case we abdicate of using hypothesis **H2**, namely:

H2. The additive noise $\nu_i(k)$ is i.i.d. of null average and independent of the signals $\nu_j(k)$ (for $j \neq i$) and $x_j(k)$ (for $j \in \{1, \ldots, M\}$).

Hypothesis **H2** is not an exclusive attribute of the difference equations technique. If one assumes **H2** to be true, as well as **H1**, it is possible to predict the MSE for the k-th iteration and for the i-th node as [19]

$$\xi_i(k) = \mathrm{Tr}\left\{\boldsymbol{R}_{\boldsymbol{x},i}\mathbb{E}\left[\tilde{\boldsymbol{\psi}}_i(k)\tilde{\boldsymbol{\psi}}_i^T(k)\right]\right\} + \sigma_{\nu_i}^2(k), \tag{31}$$

where $\mathrm{Tr}\{\boldsymbol{X}\}$ represents the trace of the matrix $\boldsymbol{X}$ and $\sigma^2_{\nu_i}(k)$ denotes the additive noise variance for the k-th iteration and for the i-th node.

Similarly, the MSD for the i-th node in the k-th iteration can be inferred through:

$$\mathrm{MSD}_i(k) = \mathrm{Tr}\left\{\mathbb{E}\left[\tilde{\boldsymbol{\psi}}_i(k)\tilde{\boldsymbol{\psi}}_i^T(k)\right]\right\}. \tag{32}$$

This work will adopt the last analysis technique in order to find recursive equations capable of estimating the MSD at each iteration (through (32)).

5.2 Recursive Equation Development

As we saw, the recursive determination of $\boldsymbol{R}_{\tilde{\boldsymbol{\psi}}_i}(k) \triangleq \mathbb{E}\left[\tilde{\boldsymbol{\psi}}_i(k)\tilde{\boldsymbol{\psi}}_i^T(k)\right]$ is essential for the development of the intended analysis (*i.e.*, via difference equations). For the distributed case, the ℓ_0-LMS algorithm update equation for the i-th node can be written as [52, 39]:

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_l(k+1) &= \boldsymbol{\psi}_l(k) + \beta_l[d_l(k) - \boldsymbol{\psi}_{l-1}^T(k)\boldsymbol{x}_l(k)]\boldsymbol{x}_l(k) + \\ &\quad + \kappa_l \boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right], \\ &= \boldsymbol{\psi}_l(k) + \beta_l e_l(k)\boldsymbol{x}_l(k) + \kappa_l \boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\end{aligned} \tag{33}$$

where $e_l(k) \triangleq d_l(k) - \boldsymbol{\psi}_{l-1}^T(k)\boldsymbol{x}_l(k)$. Parameters β_l, κ_l and ρ_l can be specific to each node and

$$\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right] \triangleq \left[f_{\rho_l}\left[\psi_{l,0}(k)\right]\, f_{\rho_l}\left[\psi_{l,1}(k)\right] ... f_{\rho_l}\left[\psi_{l,N-1}(k)\right]\right]^T,$$

where $\psi_{l,i}(k)$ is the i-th coefficient of the vector $\boldsymbol{\psi}_l(k)$.

Equation (33) needs to be manipulated in order to represent a recursion in the deviations (see (30)). Therefore[3]

$$\tilde{\boldsymbol{\psi}}_l(k+1) = \tilde{\boldsymbol{\psi}}_l(k) - \beta_l e_l(k)\boldsymbol{x}_l(k) - \kappa_l \boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]. \tag{34}$$

The next stage consists of rewriting the error $e_l(k)$ as a function of the deviations $\tilde{\boldsymbol{\psi}}_l(k)$, which can be done through:

$$\begin{aligned}e_l(k) &= d_l(k) - \boldsymbol{\psi}_l^T(k)\boldsymbol{x}_l(k), \\ &= \left(\boldsymbol{w}^o\right)^T \boldsymbol{x}_l(k) + \nu_l(k) - \boldsymbol{\psi}_l^T(k)\boldsymbol{x}_l(k), \\ &= \tilde{\boldsymbol{\psi}}_l^T(k)\boldsymbol{x}_l(k) + \nu_l(k).\end{aligned} \tag{35}$$

Inserting (35) in (34) produces:

[3] We will not express the argument of $\boldsymbol{f}_{\rho_l}[\cdot]$ in terms of deviations $\tilde{\boldsymbol{\psi}}_l(k)$ to reduce the size of the equations. Such a substitution can be easily performed after the mathematical deductions.

$$\tilde{\boldsymbol{\psi}}_l(k+1) = \left[\boldsymbol{I}_N - \beta_l \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\tilde{\boldsymbol{\psi}}_l(k) + \\ -\beta_l \boldsymbol{x}_l(k)\nu_l(k) - \kappa_l \boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right], \tag{36}$$

where $\boldsymbol{I}_N$ is the identity matrix of dimension $N \times N$.

Expression (36) consists of an exact recursive and deterministic equation, which is not capable of reflecting the average behaviour of the algorithm. In order to do that, we need to employ the expectation operator, resulting in:

$$\mathbb{E}\left\{\tilde{\boldsymbol{\psi}}_l(k+1)\right\} = \mathbb{E}\left\{\left[\boldsymbol{I}_N - \beta_l \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\tilde{\boldsymbol{\psi}}_l(k)\right\} + \\ -\kappa_l \mathbb{E}\left\{\boldsymbol{f}_{\rho_l}\left[\tilde{\boldsymbol{\psi}}_l(k)\right]\right\}, \tag{37}$$

where we employ **H2** to eliminate the term $\mathbb{E}\{\beta_l \boldsymbol{x}_l(k)\nu_l(k)\}$.

Calculating the right-hand terms of (37) is difficult since it requires determining the combined probability densities of several random variables. The adoption of **H1** can approximate (37), which results in

$$\mathbb{E}\left\{\tilde{\boldsymbol{\psi}}_l(k+1)\right\} = \left[\boldsymbol{I}_N - \beta_l \boldsymbol{R}_{x,l}\right]\mathbb{E}\left\{\tilde{\boldsymbol{\psi}}_l(k)\right\} + \\ -\kappa_l \mathbb{E}\left\{\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\right\}, \tag{38}$$

where matrix $\boldsymbol{R}_{x,l}$ does not depend on k, since the input signals are assumed to be stationary.

By only incorporating first-order statistics, Equation (38) is not sufficient to characterize algorithmic performance since it is incapable of predicting instability. The latter of which only occurs when deviation variance (a second-order statistic), grows without limit. Furthermore, performance measurement metrics such as MSE and MSD depend on second-order statistics. Therefore, it is necessary to devise a second-order stochastic recursive equation describing the expected value $\mathbb{E}\left\{\tilde{\boldsymbol{\psi}}_l(k+1)\tilde{\boldsymbol{\psi}}_l^T(k+1)\right\}$ in terms of its values on the k-th iteration. Additionally, the expression should also be derived from an equation similar to (36). Doing that requires first transposing (36):

$$\tilde{\boldsymbol{\psi}}_l^T(k+1) = \tilde{\boldsymbol{\psi}}_l^T(k)\left[\boldsymbol{I}_N - \beta_l \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right] + \\ -\beta_l \boldsymbol{x}_l^T(k)\nu_l(k) - \kappa_l \boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]. \tag{39}$$

Multiplying (36) by (39) produces:

$$\begin{aligned}\tilde{\boldsymbol{\Psi}}(k+1) = \tilde{\boldsymbol{\Psi}}(k) & \\ -\beta_l\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k)\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k) & \\ -\kappa_l\tilde{\boldsymbol{\psi}}_l(k)\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right] + & \\ -\beta_l\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k) & \\ +\beta_l^2\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k)\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k) & \\ +\beta_l\kappa_l\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\tilde{\boldsymbol{\psi}}_l(k)\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right], & \\ +\beta_l^2\boldsymbol{x}_l^T(k)\nu_l^2(k) - \kappa_l\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\tilde{\boldsymbol{\psi}}_l^T(k) & \\ +\beta_l\kappa_l\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\tilde{\boldsymbol{\psi}}_l^T(k)\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k) & \\ +\kappa_l^2\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right] & \\ +\mathcal{O}\left[\nu_l(k)\right] & \end{aligned} \tag{40}$$

where

$$\tilde{\boldsymbol{\Psi}}(k) \triangleq \tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k) \tag{41}$$

and the component $\mathcal{O}\left[\nu_l(k)\right]$ incorporates the first-order terms of the noise measurements of the l-th node $\nu_l(k)$. These, as will later be shown, will have no impact on the final result of this analysis.

Isolating the terms $\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k)$, requires performing a mathematical manipulation that starts with the employment of operator $\mathrm{vec}(\boldsymbol{A})$. This operator, when applied to a matrix $\mathrm{vec}(\boldsymbol{A})$, returns a column vector, generated by concatenating the columns of $\boldsymbol{A}$. The symbol $\otimes$ is used to represent the Kronecker product, which allows us to obtain the following relation [67]:

$$\mathrm{vec}\left[\boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z}\right] = \left(\boldsymbol{Z}^T \otimes \boldsymbol{X}\right)\mathrm{vec}(\boldsymbol{Y}). \tag{42}$$

By applying operator $\mathrm{vec}(\cdot)$ to (40) and being careful with expression (42), we obtain the identity (43).

$$\begin{aligned}\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k+1)\tilde{\boldsymbol{\psi}}_l^T(k+1)\right] = & \\ \{\boldsymbol{I}_{N^2} - \beta_l\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\otimes\boldsymbol{I}_N\right] & \\ -\beta_l\left[\boldsymbol{I}_N\otimes\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right] + \beta_l^2\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\otimes\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\}\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k)\right] & \\ +\kappa_l\{\beta_l\left[\boldsymbol{I}_N\otimes\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right] - \boldsymbol{I}_{N^2}\}\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k)\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right] & \\ +\kappa_l\{\beta_l\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\otimes\boldsymbol{I}_N\right] - \boldsymbol{I}_{N^2}\}\mathrm{vec}\left[\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\tilde{\boldsymbol{\psi}}_l^T(k)\right] & \\ \beta_l^2\nu_l^2(k)\mathrm{vec}\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right] + \kappa_l^2\mathrm{vec}\left\{\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right\} + \mathrm{vec}\left\{\mathcal{O}[\nu_l(k)]\right\}. & \end{aligned} \tag{43}$$

The deterministic equation (43) can be converted into a stochastic equation through the application of the expectation operator. By defining $\boldsymbol{v}_l(k) \triangleq \mathbb{E}\left\{\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k)\tilde{\boldsymbol{\psi}}_l^T(k)\right]\right\}$, it is possible to obtain:

$$\boldsymbol{v}_l(k+1) = \overbrace{\left\{\boldsymbol{I}_{N^2} - \beta_l \boldsymbol{A}_l + \beta_l^2 \boldsymbol{B}_l\right\}}^{\triangleq \boldsymbol{D}_l} \boldsymbol{v}_l(k) + \beta_l^2 \sigma_{\nu,l}^2(k) \boldsymbol{c}_l + \kappa_l \boldsymbol{Z}_l(k), \tag{44}$$

where the term $\mathbb{E}\{\mathcal{O}(\nu_l(k))\}$ was removed (through the **H2** hypothesis) and some expected values of the products of random variables were converted into product of expected values, by means of **H1**. The term $\sigma_{\nu,l}^2(k) \triangleq \mathbb{E}\left[\nu_l^2(k)\right]$ can be singled out by the application of **H2**. Equation (44) presents some matrices (or vectors) not yet mentioned, which can be defined by:

$$\boldsymbol{A}_l \triangleq \mathbb{E}\left\{\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k) \otimes \boldsymbol{I}_N\right] + \left[\boldsymbol{I}_N \otimes \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\right\} \tag{45}$$

$$\boldsymbol{B}_l \triangleq \mathbb{E}\left\{\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k) \otimes \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\right\} \tag{46}$$

$$\boldsymbol{C}_l \triangleq \mathbb{E}\left\{\mathrm{vec}\left[\boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\right\} \tag{47}$$

$$\boldsymbol{D}_l \triangleq \mathbb{E}\left\{\boldsymbol{I}_{N^2} - \beta_l \boldsymbol{A}_l + \beta_l^2 \boldsymbol{B}_l\right\} \tag{48}$$

$$\begin{aligned}\boldsymbol{Z}_l(k) \triangleq\ & \mathbb{E}\left\{\beta_l\left[\boldsymbol{I}_N \otimes \boldsymbol{R}_{\boldsymbol{x},l}\right] - \boldsymbol{I}_{N^2}\right\} \\ & + \mathbb{E}\left\{\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k)\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right]\right\} \\ & + \kappa_l \mathbb{E}\left\{\mathrm{vec}\left\{\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right\}\right\} \\ & + \mathbb{E}\left\{\beta_l\left[\boldsymbol{R}_{\boldsymbol{x},l} \otimes \boldsymbol{I}_N\right] - \boldsymbol{I}_{N^2}\right\} \\ & + \mathbb{E}\left\{\mathrm{vec}\left[\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\tilde{\boldsymbol{\psi}}_l^T(k)\right]\right\}.\end{aligned} \tag{49}$$

Equations (37) and (44) are capable of providing a set of diverse information about the learning process of the ℓ_0-LMS incremental algorithm. Matrices $\boldsymbol{A}_l$, $\boldsymbol{B}_l$, $\boldsymbol{D}_l$, $\boldsymbol{R}_{x,l}$ and the vector $\boldsymbol{c}_l$ present a simple structure, that only depends on statistics of the input signal. On the other hand, calculating vector $\boldsymbol{Z}_l(k)$ is more challenging, and it will be respectively our main focus next.

5.3 Evaluation of the Expected value of functions of the deviations

The terms

$$\mathbb{E}\left\{\mathrm{vec}\left[\tilde{\boldsymbol{\psi}}_l(k)\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right]\right\}$$

$$\mathbb{E}\left\{\mathrm{vec}\left\{\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\boldsymbol{f}_{\rho_l}^T\left[\boldsymbol{\psi}_l(k)\right]\right\}\right\}$$

and

$$\mathbb{E}\left\{\mathrm{vec}\left[\boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right]\tilde{\boldsymbol{\psi}}_l^T(k)\right]\right\}$$

can be obtained if we have analytical expressions capable of calculating the following expectation values:

$$\mathbb{E}\left\{\tilde{\psi}_{l,i}(k) f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\}, \tag{50}$$

$$\mathbb{E}\left\{f_{\rho_l}\left[\tilde{\psi}_{l,i}(k)\right] f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\}, \tag{51}$$

for $i, j \in \{0, 2, \ldots, N-1\}$. Calculating the terms (50) and (51) is made easier if hypothesis **H3** is considered, namely:

H3. The adaptive coefficients $\psi_{l,i}(k)$ (as well as the deviations $\tilde{\psi}_{l,i}(k)$) are guassian random variables.

Hypothesis **H3** is very popular in the literature [67], having been the object of empirical studies in [52]. Though important, this hypothesis is insufficient for obtaining analytical expressions. This is due to the terms (50) and (51) requiring the resolution of bidimensional integrals, which do not presented closed formulations. In the literature, the solution to this deadlock resides in the following additional hypothesis [52,67]:

H4. The approximations

$$\begin{aligned}\mathbb{E}\left\{\tilde{\psi}_{l,i}(k) f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\} &\approx \mathbb{E}\left\{\tilde{\psi}_{l,i}(k)\right\}\mathbb{E}\left\{f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\}\\ \mathbb{E}\left\{f_{\rho_l}\left[\psi_{l,i}(k)\right] f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\} &\approx \mathbb{E}\left\{f_{\rho_l}\left[\psi_{l,i}(k)\right]\right\}\\ &\quad \mathbb{E}\left\{f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\}\end{aligned} \tag{52}$$

are reasonably accurate. By employing **H3**-**H4**, the term $\mathbb{E}\left\{f_{\rho_l}\left[\psi_{l,j}(k)\right]\right\}$ can be calculated by [52]:

$$\begin{aligned}&\mathbb{E}\left\{f_{\rho_l}\left(\psi_{l,j}(k)\right)\right\}\\ &= \frac{1}{\sqrt{2\pi}\sigma_{i,j,k}}\int_{-\infty}^{\infty} f_{\rho_l}\left[\psi_{l,j}(k)\right] e^{-\frac{\left(\psi_{l,j}(k)-\mu_{i,k}\right)^2}{2\sigma_{i,j,k}^2}} d\psi_{l,j}(k)\\ &= \frac{\rho_l^2\sigma_{i,j,k}}{\sqrt{2\pi}}\left[e^{\frac{-\left(\mu_{i,j,k}+\frac{1}{\rho_l}\right)^2}{2\sigma_{i,j,k}^2}} - e^{\frac{-\left(\mu_{i,j,k}-\frac{1}{\rho_l}\right)^2}{2\sigma_{i,j,k}^2}}\right]\\ &\quad + \frac{\rho_l^2\mu_{i,j,k}}{2}\left[\mathrm{erf}\left(\frac{\mu_{i,j,k}+\frac{1}{\rho_l}}{\sqrt{2}\sigma_{i,j,k}}\right) - \mathrm{erf}\left(\frac{\mu_{i,j,k}-\frac{1}{\rho_l}}{\sqrt{2}\sigma_{i,j,k}}\right)\right]\\ &\quad + \frac{\rho_l}{2}\left[\mathrm{erf}\left(\frac{\mu_{i,j,k}+\frac{1}{\rho_l}}{\sqrt{2}\sigma_{i,j,k}}\right) + \mathrm{erf}\left(\frac{\mu_{i,j,k}-\frac{1}{\rho_l}}{\sqrt{2}\sigma_{i,j,k}}\right)\right]\\ &\quad - \rho_l\mathrm{erf}\left(\frac{\mu_{i,j,k}}{\sqrt{2}\sigma_{i,j,k}}\right),\end{aligned}$$

where $\mathrm{erf}(x) \triangleq \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$, $\mu_{i,j,k} \triangleq \mathbb{E}\left[\psi_{l,j}(k)\right]$ and

$$\sigma_{i,j,k} \triangleq \sqrt{\mathbb{E}\left[\psi_{l,j}^2(k)\right] - \mu_{i,j,k}^2}. \tag{53}$$

By calculating the expected values (50) and (51) (and by employing the recursive equations (37) and (44)), we are able to establish a stochastical model capable of predicting the transient behavior of the ℓ_0-LMS incremental algorithm.

5.4 Tracking analysis

One of the most important abilities of an adaptive filtering algorithm consists in tracking modifications of the transfer function to be identified. A stochastic model capable of predicting the transient, or the steady-state, of an adaptive filtering algorithm in a non-stationary environment provides guarantees concerning the tracking capabilities of the algorithm. Normally, such evaluations are performed for the case in which the ideal transfer function varies over time [64]. Furthermore, *random walk* model [7] is used for evaluation. In this mode, the ideal function to be identified varies over time in accordance with:

$$\boldsymbol{w}^{\star}(k+1) = \boldsymbol{w}(k) + \boldsymbol{q}(k), \tag{54}$$

where $\boldsymbol{q}(k)$ denotes a random perturbation. Accordingly, Eq. (36), in a tracking context, can be rewritten as:

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_l(k+1) = \left[\boldsymbol{I}_N - \beta_l \boldsymbol{x}_l(k)\boldsymbol{x}_l^T(k)\right]\tilde{\boldsymbol{\psi}}_l(k) + \\ -\beta_l \boldsymbol{x}_l(k)\nu_l(k) - \kappa_l \boldsymbol{f}_{\rho_l}\left[\boldsymbol{\psi}_l(k)\right] + \boldsymbol{q}(k).\end{aligned} \tag{55}$$

The first and second order statistical analysis of (55) require an additional hypothesis, namely [64]:

H5. The vector sequence $\{\boldsymbol{q}(k)\}$ is stationary presenting zero mean independent values. Furthermore, these vectors are independent from the $x_l(k)$ and $\nu_l(k)$ signals.

Hypothesis **H5** gives rise to a first-order Markov model (see [43] for an alternative modelling example, which assumes a component that periodically varies over time for an ideal transfer function). The application of the expectation operator in (55) (accompanied by hypotheses **H1** and **H5**) results in a first-order update equation identical to the one presented in (38). Meanwhile, in case of the second-order statistics, there is an additional term regarding (44) which produces [22]:

$$\boldsymbol{v}_l(k+1) \triangleq \boldsymbol{D}_l\boldsymbol{v}_l(k) + \beta_l^2\sigma_{\nu,l}^2(k)\boldsymbol{c}_l + \kappa_l\boldsymbol{Z}_l(k) + \boldsymbol{\vartheta}, \tag{56}$$

where $\boldsymbol{\vartheta} \triangleq \mathbb{E}\left\{\text{vec}\left[\boldsymbol{q}(k)\boldsymbol{q}^T(k)\right]\right\}$. Assuming **H5**, identity (56) reveals that a small modification of (44) allows the stochastic model to be extended for the tracking case.

Normally, a reduced learning factor β_l implies a small misadjustment in the steady-state regime [36]. However, in a tracking context, very small learning factors can imply performance degradation in the permanent regime since

tracking capacity is reduced. Excessive coefficient oscillations related to the choice of the elevated learning factors, allows us to conclude that raising them inordinately does not solve the problem. Also, we did not take into consideration here the real possibility that a significant increase of these factors may result in a divergence of the algorithm. This means that it is very common to have an optimal value for the learning factors, which is: (i) not very small in order to not harm the tracking ability; (ii) nor to high, to avoid excessive oscillations.

6 Results

Firstly, we should emphasize that it is possible to obtain MSE and MSD evolution estimates metrics for each agent both in the transient and the permanent regime. Other evaluation metrics of network global performance consist of global MSE and MSD, respectively defined as:

$$\mathrm{MSE}_{\mathrm{global}}(k) \triangleq \frac{1}{M}\sum_{m=1}^{M}\mathrm{MSE}_m(k), \tag{57}$$

$$\mathrm{MSD}_{\mathrm{global}} \triangleq \frac{1}{M}\sum_{m=1}^{M}\mathrm{MSD}_m(k), \tag{58}$$

which can be inferred from the local MSE and MSD values.

This section also compares the performance metrics predicted and simulated for the case of identifying a transfer function varying over time, in accordance with a first-order Markov model. This was done in order to evaluate the ℓ_0-LMS incremental algorithm tracking properties.

In all of the following scenarios, the transfer functions to be identified consist of the first N samples of the different models presented in [1], scaled by an α factor. This was done to avoid adaptive coefficients with small magnitude. Unless explicitly stated, the following items are uniformly distributed over the respectively stated range: (i) the learning factors over a predetermined range $[\beta_{\min}, \beta_{\max}]$; (ii) the coefficients of the zero attractors over range $[\kappa_{\min}, \kappa_{\max}]$; (iii) parameters ρ_l over range $[\rho_{\min}, \rho_{\max}]$; (iv) the additive measurement noise variances (gaussian and white) over range $[\sigma^2_{\nu,\min}, \sigma^2_{\nu,\max}]$. The independent Monte Carlo trials in each scenario is represented by E_{MC}.

6.1 Scenario 1

This first scenario employs $N = 8$ and the first model of [1] with $\alpha = 100$ and $E_{\mathrm{MC}} = 1000$. The ideal transfer function is depicted in Fig. 2. The network consists of $M = 4$ agents, with the ranges for parameters β, ρ, κ e σ^2_ν described in Tab. 1.

This scenario focuses in comparing the transient behaviour and the steady-state of the theoretical analysis with the results obtained via simulation. In

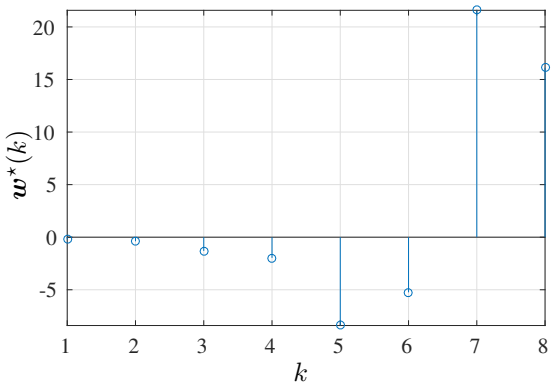

Fig. 2 Ideal plant adopted in Scenario 1.

Table 1 Simulation parameters for Scenario 1

Parameter	Minimum Value	Maximum Value
β	10^{-3}	10^{-4}
κ	10^{-10}	10^{-6}
ρ	2	5
σ_{ν}^2	10^{-6}	1

Fig. 3 the blue filled lines describe empirical MSE evolution for each agent whilst the dotted red lines present theoretical MSE evolution. A comparison of both curves shows good model adherence to the Monte Carlo trials.

6.2 Scenario 2

Equation (38), which focuses on first order statistics, can be employed to obtain the theoretical evolution for each adaptive coefficient through the iterations. Evaluating such predictions, when compared against the results obtained via simulation, is one of the objectives of the second scenario, which employs $N = 10$ and the fourth model of [1] with $\alpha = 100$ (see Fig. 4). The network consists of $M = 10$ agents, with the ranges for parameters ρ, κ e σ_{ν}^2 presented in Tab. 2. All the agents use the same β value. Fig. 5 presents the results for some randomly chosen coefficients (arbitrarily chosen) through the network for $\beta = 10^{-3}$. Once more it is possible to observe the quality of the theoretical predictions. The greater variability of the adaptive coefficients regarding the first nodes was expected. This is due to these nodes having a measurement noise whose variance is greater. By changing β for both nodes, it is possible to compare theoretical estimates and the simulation data for global steady-state MSD. A comparison performed in Fig. 6, which again reflects the accuracy of the proposed stochastic model.

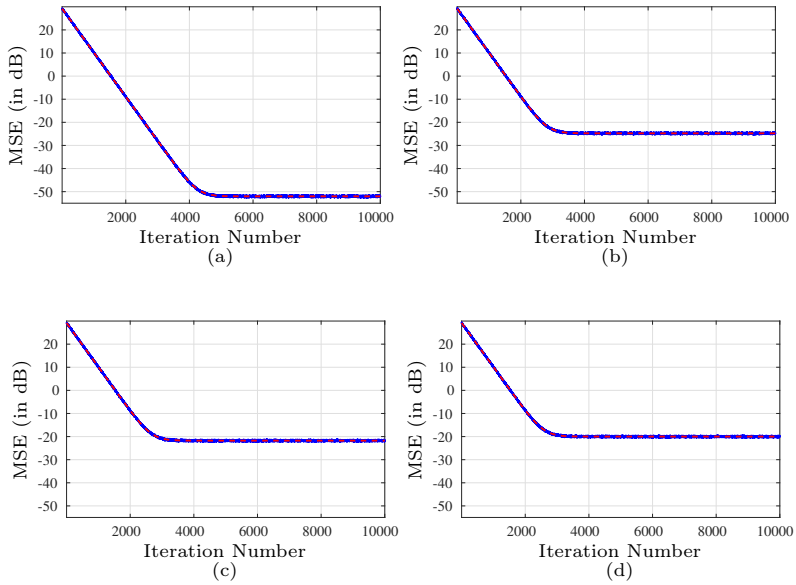

Fig. 3 Empirical MSE evolution (blue-filled lines) and theoretical (red-dashed lines) for Scenario 1. (a) First node; (b) second node; (c) third node; (d) fourth node. All figures are presented in the same scale.

Table 2 Simulation parameters for Scenario 2.

Parameter	Value for the first node	Value for the last node
κ	10^{-8}	10^{-7}
ρ	1	10
σ_ν^2	10^{-2}	10^{-3}

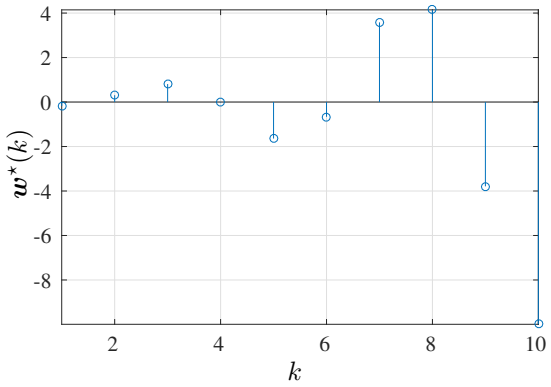

Fig. 4 Ideal plant adopted in Scenario 2.

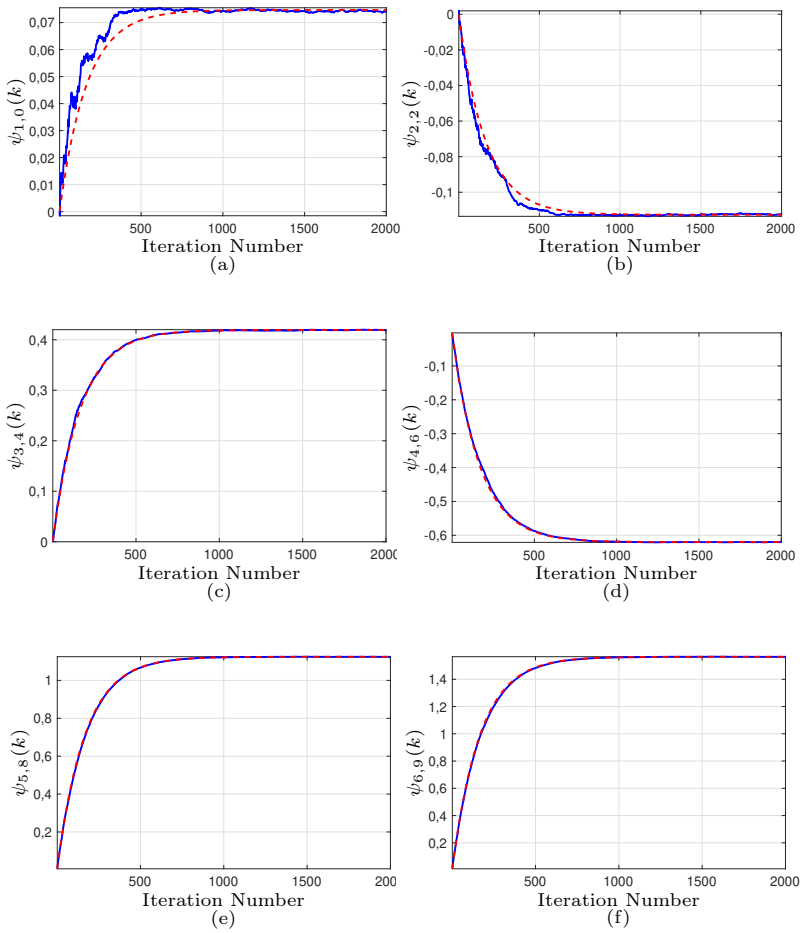

Fig. 5 Theoretical evolution (red-dashed lines) and experimental (blue-filled lines) for the second scenario and the i-th adaptive coefficient of the l-th node $\psi_{l,i}(k)$ ($E_{\mathrm{MC}} = 10$). (a) first node and $i = 0$; (b) second node and $i = 2$; (c) third node and $i = 4$; (d) fourth node and $i = 6$; (e) fifth node and $i = 8$ and (f) sixth node and $i = 9$.

6.3 Scenario 3

The third scenario evaluates model adherence when MSE in the steady-state is compared as a function of κ. Model 4 of [1] was used with $N = 12$, $M = 4$ agents and $E_{\mathrm{MC}} = 100$. Fig. 7 depicts the employed ideal transfer function. The remaining parameters were distributed along the nodes in accordance with Tab. 3. As Fig. 8 shows, it is possible to observe good curve adherence whilst κ varies by two orders of magnitude.

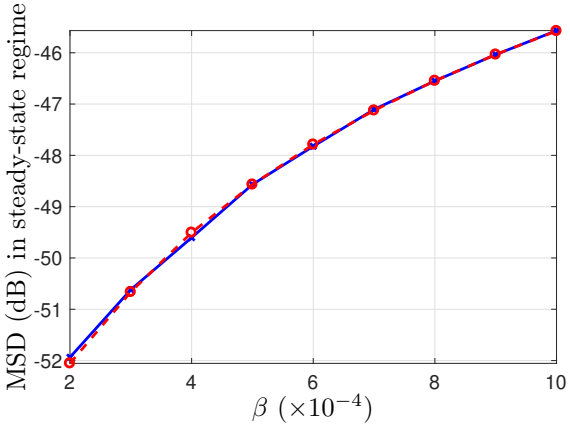

Fig. 6 Theoretical steady-state MSD (red-dashed line) and experimental (blue-filled) line for the second scenario, as a function of β ($E_{\mathrm{MC}} = 1000$). All nodes employ the same learning factor.

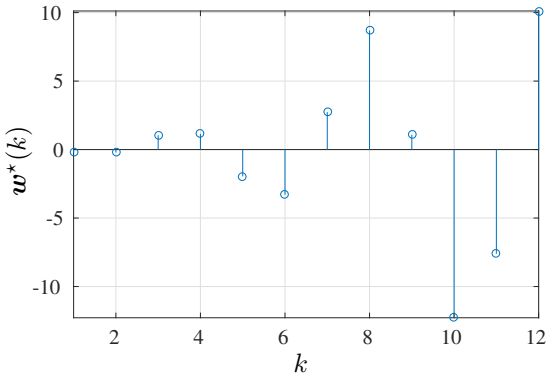

Fig. 7 Ideal plant adopted in Scenario 3.

Table 3 Simulation Parameters for Scenario 3.

Parameter	Value for the first node	Value for the last node
β	10^{-3}	5×10^{-2}
ρ	$1,25$	$1,25$
σ_{ν}^2	10^{-2}	10^{-3}

6.4 Scenario 4

The fourth scenario evaluates the precision of the proposed model for the challenging tracking case. This scenario makes use of the third model of [1] with $N = 8$ (see Fig. 9), $M = 10$ agents and $E_{\mathrm{MC}} = 100$. The remaining parameters were distributed along the nodes in accordance with Tab. 4.

A $\sigma_q^2 = 10^{-11}$ variance was employed for the elements of the random perturbation vector $\boldsymbol{q}(k)$. Fig. 10 presents the MSD in the steady-state as a function

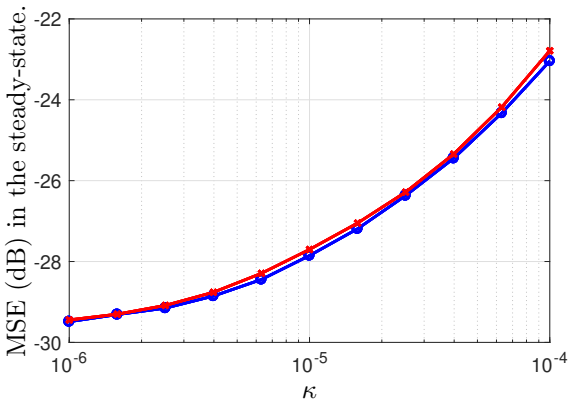

Fig. 8 Theoretical MSE in the steady-state regime (red-filled line) and experimental (blue-filled line) for the third scenario as a function of κ, ($E_{\mathrm{MC}} = 100$).

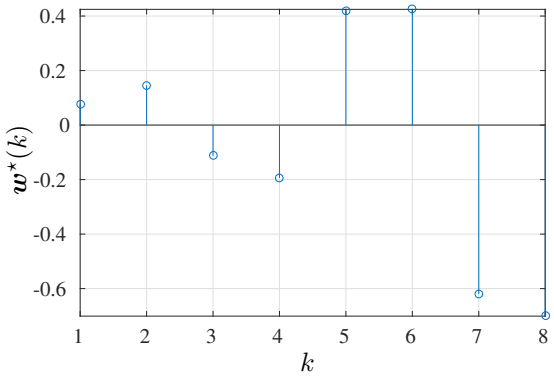

Fig. 9 Ideal plant adopted in Scenario 4.

Table 4 Simulation parameters for Scenario 4.

Parameter	Minimum Value	Maximum Value
κ	10^{-9}	10^{-8}
ρ	1	10
σ_ν^2	10^{-6}	10^{-4}

of parameter β, which was employed for all the agents. This figure shows that in a non-stationary scenario the MSD in the steady-state is not monotonically increasing as β increases. As previously stated, a small β value harms the algorithms tracking ability, which gives rise to an optimal β value that minimizes the MSD in the steady-state. This value is accurately predicted by the theoretical analysis, as is illustrated by the simulated and theoretical curves of Fig. 10. The MSE evolution of the first node for a β value equal to 5×10^{-5} is presented in Fig. 11. Again, the plot shows a good correlation between the predicted results and the simulated ones.

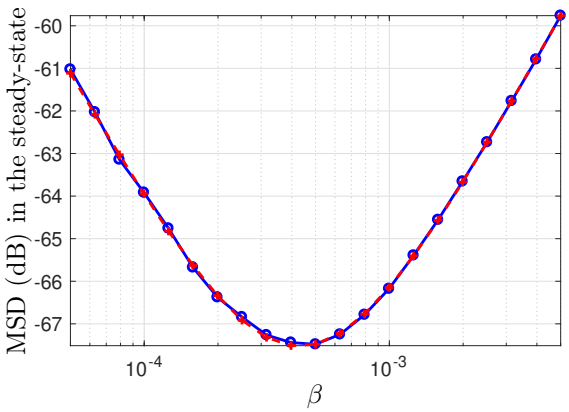

Fig. 10 Global MSD in the theoretical steady-state (red-dashed line) and experimental (blue-filled line) for the fourth scenario as a function of β. All the nodes employ the same learning factor value. The variance of the random perturbation applied to each one of the adaptive coefficients was $\sigma_q^2 = 10^{-11}$.

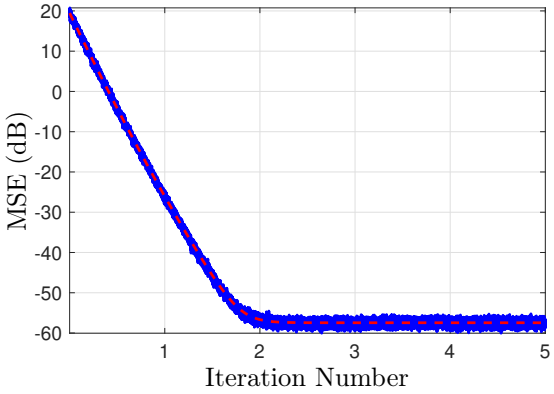

Fig. 11 First node MSE evolution for the fourth scenario with $\beta = 5\times10^{-5}$ and $\sigma_q^2 = 10^{-11}$. All the nodes employ the same learning factor value. Theoretical values are presented in red-dashed lines, whilst experimental results are presented in blue.

7 Conclusions

The ℓ_0-LMS is one of the most popular sparsity-aware adaptive filtering algorithms, which also lends itself to be implemented in a distributed manner. This article focused on the case where the algorithm is employed in an adaptive network, operating in an incremental manner, that aims to identify a system in a diffuse manner.

A stochastic model was proposed in order to model the empirical results concerning the learning of the algorithm. Doing this required considering classical or popular approaches in the literature, such as the independence hypothesis. The model was constructed in a general manner as to allow for: (i) variation in the number of agents (each of which with a specific learning fac-

tor); (ii) variation of the penalization factor; (iii) zero-attraction strength; and (iv) statistical properties of the input signal.

The model is capable of accurately predicting the average evolution: (i) of the coefficients; (ii) the MSE and MSD metrics that depend on second order statistics; and (iii) the coefficient update probability, which impacts the computational complexity of the algorithm. Furthermore, it is possible to predict steady-state performance. These allow for performance observation as a function of a selected parameter. For the case in which the transfer function is time variant (depending on a first-order Markov random walk) the algorithm's learning capability is also predicted.

Acknowledgment

This work has been supported by CNPq, FAPERJ and CAPES.

References

1. 15, I.T.S.G.: Digital network echo cancellers (recommendation). Tech. Rep. G.168, ITU-T (2004)
2. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks **38**(4), 393 – 422 (2002)
3. Al-Naffouri, T.Y., Sayed, A.H.: Transient analysis of adaptive filters with error nonlinearities. IEEE Transactions on Signal Processing **51**(3), 653–663 (2003)
4. Al-Sayed, S., Zoubir, A.M., Sayed, A.H.: Robust distributed estimation by networked agents. IEEE Transactions on Signal Processing **65**(15), 3909–3921 (2017)
5. Albu, F., Kwan, H.K.: Combined echo and noise cancellation based on gauss-seidel pseudo affine projection algorithm. In: 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512), vol. 3, pp. III–505 (2004)
6. Albu, F., Kwan, H.K.: Fast block exact gauss-seidel pseudo affine projection algorithm. Electronics Letters **40**(22), 1451–1453 (2004)
7. Arablouei, R., Dofancay, K.: Tracking performance analysis of the set-membership nlms adaptive filtering algorithm. In: Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1–6 (2012)
8. Baraniuk, R.G.: Compressive sensing. IEEE Signal Processing Magazine **24**(4), 118–121 (2007)
9. Benesty, J., Huang, Y., Chen, J.: An exponentiated gradient adaptive algorithm for blind identification of sparse simo systems. Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP) **2**, 829–832 (2004)
10. Benesty, J., S. L. Gay, S.L.: An improved pnlms algorithm. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. 1881–1884. IEEE (2002)
11. Chan, S.C., Chu, Y.J., Zhang, Z.G.: A new variable regularized transform domain nlms adaptive filtering algorithm - acoustic applications and performance analysis. IEEE Transactions on Audio, Speech, and Language Processing **21**(4), 868–878 (2013)
12. Chang, S., Ogunfunmi, T.: Performance analysis of nonlinear adaptive filter based on lms algorithm. In: Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 107–110 (1997)
13. Chen, J., Richard, C., Sayed, A.H.: Multitask diffusion adaptation over networks with common latent representations. IEEE Journal of Selected Topics in Signal Processing **11**(3), 563–579 (2017)
14. Chen, J., Richard, C., Song, Y., Brie, D.: Transient performance analysis of zero-attracting lms. IEEE Signal Processing Letters **23**(12), 1786–1790 (2016)

15. Das, B.K., Azpicueta-Ruiz, L.A., Chakraborty, M., Arenas-García, J.: A comparative study of two popular families of sparsity-aware adaptive filters. In: 4th International Workshop on Cognitive Information Processing (CIP), pp. 1–6. IEEE (2014)
16. Das, B.K., Chakraborty, M., Arenas-García, J.: Sparse distributed learning via heterogeneous diffusion adaptive networks. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 437–440 (2015)
17. Das, B.K., Chakraborty, M., Arenas-García, J.: Sparse distributed estimation via heterogeneous diffusion adaptive networks. IEEE Transactions on Circuits and Systems II: Express Briefs **63**(11), 1079–1083 (2016)
18. Deng, H., Doroslovacki, M.: Proportionate adaptive algorithms for network echo cancellation. IEEE Transactions on Signal Processing **54**(5), 1794–1803 (2006)
19. Diniz, P.S.R.: Adaptive filtering: algorithms and practical implementation, vol. 694. Springer Verlag (2008)
20. Douglas, S.C., Pan, W.: Exact expectation analysis of the lms adaptive filter. IEEE Transactions on Signal Processing **43**(12), 2863–2871 (1995)
21. Duttweiler, D.L.: Proportionate normalized least-mean-squares adaptation in echo cancelers. IEEE Transactions on Speech and Audio Processing **8**(5), 508–518 (2000)
22. Eweda, E.: Tracking analysis of the sign-sign algorithm for nonstationary adaptive filtering with gaussian data. IEEE Transactions on Signal Processing **45**(5), 1375–1378 (1997)
23. Fernandez-Bes, J., Arenas-García, J., Silva, M.T.M., Azpicueta-Ruiz, L.A.: Adaptive diffusion schemes for heterogeneous networks. IEEE Transactions on Signal Processing **65**(21), 5661–5674 (2017)
24. Fukumoto, M., Sachio, S.S.: An improved mu-law proportionate nlms algorithm. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3797–3800. IEEE (2008)
25. Gay, S.L.: An efficient, fast converging adaptive filter for network echo cancellation. In: Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 394–398. IEEE (1998)
26. Gu, Y., Jin, J., Mei, S.: Norm constraint lms algorithm for sparse system identification. IEEE Signal Processing Letters **16**(9), 774–777 (2009)
27. Haddad, D.B.: Estruturas em subbandas para filtragem adaptativa e separação cega e semi-cega de sinais de voz. Ph.D. thesis, UFRJ (2013)
28. Haddad, D.B., Petraglia, M.R.: Transient and steady-state mse analysis of the impnlms algorithm. Digital Signal Processing **33**, 50–59 (2014)
29. Haddad, D.B., Petraglia, M.R., Petraglia, A.: A unified approach for sparsity-aware and maximum correntropy adaptive filters. In: 2016 24th European Signal Processing Conference (EUSIPCO), pp. 170–174 (2016)
30. Hoshuyama, O., Goubran, R.A., Sugiyama, A.: A generalized proportionate variable step-size algorithm for fast changing acoustic environments. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. iv–161–iv–164 vol.4 (2004)
31. Jelfs, B., Mandic, D.P., Benesty, J.: A class of adaptively regularised pnlms algorithms. Proceedings of the 15th International Conference on Digital Signal Processing (DSP) pp. 19–22 (2007)
32. Khalili, A., Tinati, M.A., Rastegarnia, A.: Steady-state analysis of incremental lms adaptive networks with noisy links. IEEE Transactions on Signal Processing **59**(5), 2416–2421 (2011)
33. Kuhn, E.V., das C. de Souza, F., Seara, R., Morgan, D.R.: On the steady-state analysis of pnlms-type algorithms for correlated gaussian input data. IEEE Signal Processing Letters **21**(11), 1433–1437 (2014)
34. Lara, P., Igreja, F., Tarrataca, L.D.T.J., Haddad, D.B., Petraglia, M.R.: Exact expectation evaluation and design of variable step-size adaptive algorithms. IEEE Signal Processing Letters **26**(1), 74–78 (2019)
35. Lara, P., d. S. Olinto, K., Petraglia, F.R., Haddad, D.B.: Exact analysis of the least-mean-square algorithm with coloured measurement noise. Electronics Letters **54**(24), 1401–1403 (2018)
36. Lara, P., Tarrataca, L.D.T.J., Haddad, D.B.: Exact expectation analysis of the deficient-length lms algorithm. Signal Processing **162**, 54 – 64 (2019)

37. Li, L., Chambers, J.A., Lopes, C.G., Sayed, A.H.: Distributed estimation over an adaptive incremental network based on the affine projection algorithm. IEEE Transactions on Signal Processing **58**(1), 151–164 (2010)
38. Lima, M.V.S., Ferreira, T.N., Martins, W.A., Diniz, P.S.R.: Sparsity-aware data-selective adaptive filters. IEEE Transactions on Signal Processing **62**(17), 4557–4572 (2014)
39. Liu, Y., .Li, C., Zhang, Z.: Diffusion sparse least-mean squares over networks. IEEE Transactions on Signal Processing **60**(8), 4480–4485 (2012)
40. Lopes, C.G., Sayed, A.H.: Incremental adaptive strategies over distributed networks. IEEE Transactions on Signal Processing **55**(8), 4064–4077 (2007)
41. Lopes, C.G., Sayed, A.H.: Randomized incremental protocols over adaptive networks. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3514–3517 (2010)
42. Lorenzo, P.D., Barbarossa, S., Sayed, A.H.: Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks. IEEE Transactions on Signal Processing **61**(12), 3183–3197 (2013)
43. Matsuo, M.V.: Modelagem estocástica de algoritmos adaptativos para equalização ativa de ruído e identificação de sistemas. Ph.D. thesis, UFSC, Florianópolis (2016)
44. Mazo, J.E.: On the independence theory of equalizer convergence. Bell Sys. Tech. J. **58**(3), 963–966 (1979)
45. Nascimento, V.H., Sayed, A.H.: On the learning mechanism of adaptive filters. IEEE Transactions on Signal Processing **48**(6), 1609–1625 (2000)
46. Nassif, R., Richard, C., Ferrari, A., Sayed, A.H.: Diffusion lms for multitask problems with local linear equality constraints. IEEE Transactions on Signal Processing **65**(19), 4979–4993 (2017)
47. North, R.C., Zeidler, J.R., Ku, W.H., Albert, T.R.: A floating-point arithmetic error analysis of direct and indirect coefficient updating techniques for adaptive lattice filters. IEEE Transactions on Signal Processing **41**(5), 1809–1823 (1993)
48. Paleologu, C., Benesty, J., Albu, F.: Regularization of the improved proportionate affine projection algorithm. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 169–172 (2012)
49. Park, P., Lee, C.H., Ko, J.W.: Mean-square deviation analysis of affine projection algorithm. IEEE Transactions on Signal Processing **59**(12), 5789–5799 (2011)
50. Petraglia, M.R., Haddad, D.B.: New adaptive algorithms for identification of sparse impulse responses — analysis and comparisons. In: 2010 7th International Symposium on Wireless Communication Systems, pp. 384–388 (2010)
51. d. Prado, R.A., d. R. Henriques, F., Haddad, D.B.: Sparsity-aware distributed adaptive filtering algorithms for nonlinear system identification. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2018)
52. K. da S. Olinto, D.B.H., Petraglia, M.R.: Transient analysis of ℓ_0-lms and ℓ_0-nlms algorithms. Signal Processing **127**, 217–226 (2016)
53. Sankaran, S.G., Beex, A.A.L.: Convergence behavior of affine projection algorithms. IEEE Transactions on Signal Processing **48**(4), 1086–1096 (2000)
54. Sankaranarayanan, A.C., Turaga, P., Herman, M.A., Kelly, K.F.: Enhanced compressive imaging using model-based acquisition. IEEE Signal Processing Magazine **33**(5), 81–94 (2016)
55. Sayed, A.H.: Adaptive filters. John Wiley & Sons (2011)
56. Sayed, A.H.: Adaptive networks. Proceedings of the IEEE **102**(4), 460–497 (2014)
57. Sayed, A.H.: Diffusion adaptation over networks. In: S. Theodoridis, R. Chellappa (eds.) Academic Press Library in Signal Processing: Array and Statistical Signal Processing, chap. 9, pp. 323–456. Academic Press, New York, NY, USA (2014)
58. Sayed, A.H., Rupp, M.: A time-domain feedback analysis of adaptive gradient algorithms via the small gain theorem. In: Proceedings of SPIE - Conference on Advanced Signal Processing: Algorithms, Architectures, and Implementations, vol. 2563, pp. 458–469. IEEE (1995)
59. Sayed, A.H., Tu, S.Y., Chen, J., Zhao, X., Towfic, Z.J.: Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior. IEEE Signal Processing Magazine **30**(3), 155–171 (2013)

60. Shi, K., Shi, P.: Convergence analysis of sparse lms algorithms with l1-norm penalty based on white input signal. Signal Processing **90**(12), 3289–3293 (2010)
61. Su, G., Jin, J., Gu, Y., Wang, J.: Performance analysis of ℓ_0-norm constraint least mean square algorithm. IEEE Transactions on Signal Processing **60**(5), 2223–2235 (2012)
62. Weruaga, L., Jimaa, S.: Exact nlms algorithm with ℓ_p-norm constraint. IEEE Signal Processing Letters **22**(3), 366–370 (2015)
63. Wu, F.Y., Tong, F.: Non-uniform norm constraint lms algorithm for sparse system identification. IEEE communications letters **17**(2), 385–388 (2013)
64. Yousef, N.R., Sayed, A.H.: A unified approach to the steady-state and tracking analyzes of adaptive filters. IEEE Transactions on Signal Processing **49**(2), 314–324 (2001)
65. Yukawa, M., Yamada, I.: Adaptive parallel variable-metric projection algorithm - an application to acoustic echo cancellation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 3, pp. III–1353. IEEE (2007)
66. Zakharov, Y.V.: Low-complexity implementation of the affine projection algorithm. IEEE Signal Processing Letters **15**, 557–560 (2008)
67. Zhang, S., Zhang, J.: Transient analysis of zero attracting nlms algorithm without gaussian inputs assumption. Signal processing **97**, 100–109 (2014)