Signal Processing 128 (2016) 131–141

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Reweighted nonnegative least-mean-square algorithm

Jie Chen [a,*], Cédric Richard [b], José Carlos M. Bermudez [c]

[a] Center of Intelligent Acoustics and Immersive Communications (CIAIC), School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

[b] The Université de Nice Sophia-Antipolis, UMR CNRS 7293, Observatoire de la Côte d'azur, Laboratoire Lagrange, Parc Valrose, 06102 Nice, France

[c] The Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900 Florianópolis, SC, Brazil

ARTICLE INFO

Article history:
Received 10 October 2015
Received in revised form
22 February 2016
Accepted 26 March 2016
Available online 1 April 2016

Keywords:
Online system identification
Nonnegativity constraints
Behavior analysis
Sparse system identification

ABSTRACT

Statistical inference subject to nonnegativity constraints is a frequently occurring problem in learning problems. The nonnegative least-mean-square (NNLMS) algorithm was derived to address such problems in an online way. This algorithm builds on a fixed-point iteration strategy driven by the Karush–Kuhn–Tucker conditions. It was shown to provide low variance estimates, but it however suffers from unbalanced convergence rates of these estimates. In this paper, we address this problem by introducing a variant of the NNLMS algorithm. We provide a theoretical analysis of its behavior in terms of transient learning curve, steady-state and tracking performance. We also introduce an extension of the algorithm for online sparse system identification. Monte-Carlo simulations are conducted to illustrate the performance of the algorithm and to validate the theoretical results.

1. Introduction

Online learning aims at determining a mapping from a dataset to the corresponding labels when the data are available in a sequential fashion. In particular, algorithms such as the Least-Mean-Square (LMS) and the Recursive Least-Square (RLS) algorithms minimize the mean square-error cost function in an online manner based on input/output measurement sequences [1,2]. In practice, rather than leaving the parameters to be estimated totally free and relying on data, it is often desirable to introduce some constraints on the parameter space. These constraints are usually introduced to impose some specific structures, or to incorporate prior knowledge, so as to improve the estimation accuracy and the interpretability of results in learning systems [3,4]. The nonnegativity constraint is one of the most frequently used constraints among several popular ones [5]. It can be imposed to avoid physically unreasonable solutions and to comply with physical characteristics. For example, quantities such as intensities [6,7], chemical concentrations [8], and material fractions of abundance [9] must naturally fulfill nonnegativity constraints. Nonnegativity constraints may also enhance the physical interpretability of some analyzed results. For instance, Nonnegative Matrix Factorization leads to more meaningful image decompositions than Principle Component Analysis (PCA) [10,11]. PCA and neural networks can also be conducted subject to nonnegativity constraints in order to enhance result interpretability [12,13]. Finally, there are important problems in signal processing that can be cast as optimization problems under nonnegativity constraints [14]. Other applications of learning systems related to nonnegativity constraints can be found in [15–17,5,18–20].

Nonnegativity constrained problems can be solved in a batch mode via active set methods [21,22], gradient projection methods [23,24], and multiplicative methods [25], to cite a few. Online system identification methods subject to nonnegativity constraints can also be of great interest in applications that require to adaptively identify a dynamic system. An LMS-type algorithm, called the non-negative least-mean-square (NNLMS) algorithm, was proposed in [26] to address the least-mean-square problem under nonnegativity constraints. It was derived based on a stochastic gradient descent approach combined with a fixed-point iteration strategy that ensures convergence toward a solution satisfying the Karush–Kuhn–Tucker (KKT) conditions. In [27], several variants of the NNLMS were derived to improve its convergence behavior in some sense. The steady-state performance of these algorithms was analyzed in [28]. It was observed that one limitation of the NNLMS algorithm is that the filter coefficients suffer from unbalanced convergence rates. In particular, convergence of small coefficients in the active set (the set of zero-valued optimum weights), that is, those tending to zero at steady-state, progressively slows down with time and almost stalls when approaching the steady-state (see [27] and also Fig. 3(a)). Another limitation of the NNLMS algorithm is its vulnerability to the occurrence of a large coefficient value spread. A large spread of coefficient values in NNLMS lead to a large spread of the weight updates, increasing the coefficient variances and complicating the choice of an adequate step-size.

[*] Corresponding author.
E-mail addresses: dr.jie.chen@ieee.org (J. Chen), cedric.richard@unice.fr (C. Richard), j.bermudez@ieee.org (J.C.M. Bermudez).

http://dx.doi.org/10.1016/j.sigpro.2016.03.017

The Exponential NNLMS algorithm was proposed in [27] to alleviate the first limitation. This algorithm applies a Gamma scaling function to each component of the NNLMS update. Although this NNLMS variant leads to more balanced coefficient convergence rates, it does not completely solve large coefficient update spread problem, as the scaling function is still unbounded on the coefficient values. Moreover, the exponential scaling operation tends to be computationally expensive for real-time implementations requiring a large number of coefficients. Thus, it is of interest to investigate alternative algorithms that may simultaneously address these two NNLMS limitations.

In this paper, we propose a variant of the NNLMS algorithm that balances more efficiently the convergence rate of the different filter coefficients. The entries of the gradient correction term are reweighted by a bounded differentiable sign-like function at each time instant. This gives the filter coefficients balanced convergence rates and largely reduces the sensitivity of the algorithm to large coefficient spreads. The stochastic behavior of the algorithm is then studied in detail. A statistical analysis of its transient and steady-state behavior leads to analytical models that are able to accurately predict the algorithm performance. In particular, contrary to a previous analysis [27] the algorithm tracking behavior is studied using a nonstationarity model that allows for a bounded covariance matrix of the optimal solution, a more practical scenario. The accuracy of the derived analytical models is verified through Monte Carlo simulations. Finally, the applicability of the proposed algorithm to problems whose definition does not specify nonnegativity constraints on the coefficients is illustrated through an example of identification of sparse system responses.

The rest of this paper is organized as follows. Section 2 reviews the problem of system identification under nonnegativity constraints and briefly introduces the NNLMS algorithm. Section 3 motivates and introduces the new variant of the NNLMS algorithm. In Section 4, the behavior in the mean and mean-square-error sense, and the tracking performance of this algorithm are studied. Section 5 provides simulation results to illustrate the properties of the algorithm and the accuracy of the theoretical analysis. Section 6 concludes the paper.

In this paper normal font letters (x) denote scalars, boldface small letters ($\boldsymbol{x}$) denote vectors, boldface capital letters ($\mathbf{X}$) denote matrices with $\boldsymbol{I}$ being the identity matrix. All vectors are column vectors. The superscript $(\cdot)^\top$ represents the transpose of a matrix or a vector, $\mathrm{tr}\{\cdot\}$ denotes trace of a matrix, and $E\{\cdot\}$ denotes statistical expectation. Either $\boldsymbol{D}_{\boldsymbol{x}}$ or $\boldsymbol{D}\{x_1, \ldots, x_N\}$ denote a diagonal matrix whose main diagonal is the vector $\boldsymbol{x} = [x_1, \ldots, x_N]^\top$. The operator $\mathrm{diag}\{\cdot\}$ forms a column vector with the main diagonal entries of its matrix argument.

2. Online system identification subject to nonnegativity constraints

Consider an unknown system with input-output relation characterized by the linear model:

$$y(n) = \boldsymbol{\alpha}^{\star\top}\boldsymbol{x}(n) + z(n) \tag{1}$$

with $\boldsymbol{\alpha}^\star = [\alpha_1^\star, \alpha_2^\star, \ldots, \alpha_N^\star]^\top$ an unknown parameter vector, and $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^\top$ the vector of regressors with positive definite correlation matrix $\boldsymbol{R}_{\boldsymbol{x}} > 0$. The input signal $x(n)$ and the desired output signal $y(n)$ are assumed zero-mean stationary. The modeling error $z(n)$ is assumed zero-mean stationary, independent and identically distributed (i.i.d.) with variance $\sigma_z{}^2$, and independent of any other signal. We seek to identify this system by minimizing the following constrained mean-square error criterion:

$$\begin{aligned}\boldsymbol{\alpha}^o &= \arg\min_{\alpha} J(\boldsymbol{\alpha})\\ &\quad \text{subject to } \alpha_i \ge 0, \quad \forall\, i\end{aligned} \tag{2}$$

where the nonnegativity of the estimated coefficients is imposed by inherent physical characteristics of the system, and $J(\boldsymbol{\alpha})$ is the mean-square error criterion

$$J(\boldsymbol{\alpha}) = E\{[y(n) - \boldsymbol{\alpha}^\top\boldsymbol{x}(n)]^2\} \tag{3}$$

and $\boldsymbol{\alpha}^o$ is the solution of the constrained optimization problem (2). The Lagrange function associated with this problem is given by $L(\boldsymbol{\alpha}, \lambda) = J(\boldsymbol{\alpha}) - \lambda^\top\boldsymbol{\alpha}$, with λ being the vector of nonnegative Lagrange multipliers. The KKT conditions for (2) at the optimum $\boldsymbol{\alpha}^o$ can be combined into the following expression [29,26]

$$\alpha_i^o[-\nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}^o)]_i = 0 \tag{4}$$

where $\nabla_{\boldsymbol{\alpha}}$ stands for the gradient operator with respect to $\boldsymbol{\alpha}$. Implementing a fixed-point strategy with (4) leads to the component-wise gradient descent updating rule [26]

$$\alpha_i(n+1) = \alpha_i(n) + \eta(n) f_i(\boldsymbol{\alpha}(n))\alpha_i(n)[-\nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}(n))]_i \tag{5}$$

where $\eta(n)$ is the positive step size that controls the convergence rate, $f_i(\boldsymbol{\alpha}(n))$ is a nonnegative scalar function of vector $\boldsymbol{\alpha}$ that weights its ith component α_i in the update term. Selecting different functions $f_i(\boldsymbol{\alpha}(n))$ leads to different adaptive algorithms. Particularly, making $f_i(\boldsymbol{\alpha}(n)) = 1$, using stochastic gradient approximations as done in deriving the LMS algorithm, and rewriting the update equation in vectorial form, we obtain the NNLMS algorithm [26]:

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n)\boldsymbol{D}_{\boldsymbol{\alpha}}(n)\boldsymbol{x}(n)e(n) \tag{6}$$

where $\boldsymbol{D}_{\boldsymbol{\alpha}}(n)$ is the diagonal matrix in which the elements of $\boldsymbol{\alpha}(n)$ compose the main diagonal, and $e(n)$ is the estimation error at time instant n:

$$e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n). \tag{7}$$

This iteration is similar in some sense to the expectation maximization (EM) algorithm [30]. The algorithm requires to be initialized with positive values. Suppose that $\boldsymbol{\alpha}(n)$ is nonnegative at time n. If the step size satisfies

$$0 < \eta(n) \le \min_i \frac{1}{-e(n)x_i(n)}, \tag{8}$$

for all $i \in \{j: e(n)x_j(n) < 0\}$, the nonnegativity constraint is satisfied at time $n+1$ with (6). If $e(n)x_i(n) \ge 0$, there is no restriction on the step size to guarantee the nonnegativity constraint. Convergence of the NNLMS algorithm was analyzed in [26]. Its steady-state excess mean-square error (EMSE) was studied in [28].

3. Motivating facts and the algorithm

3.1. Motivation

The weight update in (6) corresponds to the classical stochastic gradient LMS update with its ith component scaled by $\alpha_i(n)$. The mean value of the update vector $\boldsymbol{D}_{\boldsymbol{\alpha}}(n)\boldsymbol{x}(n)e(n)$ is thus no longer in the direction of the gradient of $J(\boldsymbol{\alpha})$, as is the case for LMS. On the other hand, it is exactly this scaling by $\alpha_i(n)$ that enables the corrections $\alpha_i(n)x_i(n)e(n)$ to reduce gradually to zero for coefficients $\alpha_i(n)$ tending to zero, which leads to low-variance estimates for these coefficients.[1] If a coefficient $\alpha_k(n)$ that approaches zero turns

[1] The update for these weights will be necessarily small in amplitude as $\alpha_i(n)$ tends to zero, thus leading to a small variance of the adaptive coefficient.

negative due to the stochastic update, its negative sign will induce a change $\alpha_k(n)x_k(n)e(n)$ in the kth weight component that is contrary to what would indicate the stochastic gradient, and thus towards zero.

The presence of the factor $\alpha_i(n)$ in the update $\alpha_i(n)x_i(n)e(n)$ of the ith coefficient leads to different convergence rates for coefficients of different values. This is particularly problematic for the coefficients in the active set as they approach zero. Because of the factor $\alpha_i(n)$, the convergence of these coefficients eventually stalls due to insignificant correction along their axes. Though the algorithm leads to a very low steady-state error for these coefficients, this happens only after a long convergence process. In addition, the dispersion of coefficient values introduces difficulties for step size selection and coefficient initialization, since each estimated coefficient acts as a different directional gain for the same step size. In order to address these problems, it is of interest to derive a variant of the NNLMS algorithm that satisfies the following requirements:

- The coefficients should converge to the fixed point satisfying (4), so that it still solves the nonnegativity constrained problem (2).
- The sensitivity of the algorithm (6) to the spread of the coefficient values at each iteration should be reduced, yielding more balanced convergence rates and steady-state weight errors than the original algorithm (6).
- The performance improvement should be achieved without introducing significant computational burden.

3.2. *The inversely proportional NNLMS algorithm*

The Exponential NNLMS [27] replaces the gradient scaling $\alpha_i(n)$ with $\alpha_i^\gamma(n) = \text{sgn}(\alpha_i(n))|\alpha_i(n)|^\gamma$, where $\gamma = \gamma_1/\gamma_2$ with γ_1 and γ_2 being two odd numbers such that $\gamma_2 > \gamma_1 > 0$. This variant mitigates the aforementioned drawbacks of NNLMS to some extent, but introduces additional computational burden. In addition, the stability of the algorithm is still affected by the weight dynamics since $\alpha_i^\gamma(n)$ is unbounded.

To reduce the slow-down effect caused by the factor $\alpha_i(n)$ in the update term of (6) while keeping zero as a fixed-point to attract the entries $\alpha_i(n)$ in the active set, we propose the following expression for $f_i(\boldsymbol{\alpha}(n))$ in (5):

$$f_i(\boldsymbol{\alpha}(n)) = f_i(n) = \frac{1}{|\alpha_i(n)| + \epsilon} \tag{9}$$

with ϵ being a small positive parameter so that $f_i(n) > 0$ as needed. Defining the diagonal matrix $\boldsymbol{D}_f(n)$ with ith diagonal entries given by $f_i(n)$, and using this matrix to reweight the gradient correction term at each iteration, leads to the following algorithm:

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, \boldsymbol{D}_f(n)\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n)e(n) = \boldsymbol{\alpha}(n) + \eta\boldsymbol{D}_w(n)\boldsymbol{x}(n)e(n) \tag{10}$$

where

$$\boldsymbol{D}_w(n) = \boldsymbol{D}_f(n)\boldsymbol{D}_\alpha(n) \tag{11}$$

is the diagonal matrix with ith element

$$w_i(n) = f_i(n)\alpha_i(n) = \frac{\alpha_i(n)}{|\alpha_i(n)| + \epsilon}. \tag{12}$$

In expression (10), as each entry of the NNLMS correction term is reweighted by a scalar value that is inversely proportional to $|\alpha_i(n)| + \epsilon$, we name this algorithm Inversely-Proportionate NNLMS (IP-NNLMS) by analogy with the terminology Proportionate LMS [31]. The weight $w_i(n)$ corresponds to the application of a function $\varphi_{\text{IPNNLMS}}(x) = x/(|x| + \epsilon)$ at $x = \alpha_i(n)$. The function $\varphi_{\text{IPNNLMS}}(x)$ is plotted in Fig. 1 for $\epsilon = 0.1$ and $\epsilon = 0.01$. Observe that $\varphi_{\text{IPNNLMS}}(x)$ is a smooth approximation of the sign function. The same function has been considered in [32] in the context of sparsity enhancing by reweighted ℓ_1 minimization. The correction terms $w_i(n)$ in (12) are bounded, and adjustments in the positive orthant are close to 1 (except around the origin), which does not impose scaling effect on the gradient entries. Correction terms also converge to 0 for filter coefficients $\alpha_i(n)$ that gradually tend to 0, so as to ensure convergence for these coefficients in the active set. Furthermore, as the gradient correction terms are in $]-1, 1[$, the sensitivity of the algorithm to the dynamic range of the filter coefficients is reduced. The corresponding gradient correction terms for the NNLMS and Exponential NNLMS algorithms are determined by the application of functions $\varphi_{\text{NNLMS}}(x) = x$ and $\varphi_{\text{ENNLMS}}(x) = |x|x^\gamma$. These functions are also depicted in Fig. 1 (for $\gamma = 3/7$) for comparison.

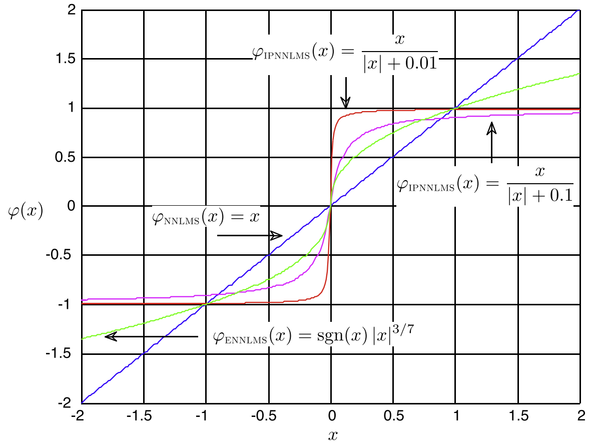


Fig. 1. Gradient scaling function of NNLMS, IP-NNLMS ($\epsilon = 0.1$ and 0.01), and Exponential NNLMS with $\gamma = \frac{3}{7}$ [27]. Note that $w(x)$ is unbounded for NNLMS and Exponential NNLMS.

3.3. *Computational complexity:*

A comparison of the computational complexities in the implementation of the NNLMS, Exponential NNLMS and IP-NNLMS algorithms needs to consider only the weight update term, since this term is what distinguishes the three algorithms. We consider their real-time implementation using N coefficients and m-bit integer operations. Also, because there exists a variety of multiplication algorithms, let us denote by $\mathcal{M}(m)$ the complexity of the chosen algorithm to multiply two m-bit integers. The NNLMS update (6) sets the weighting function $w_i(n)$ to $\alpha_i(n)$, and has complexity of $O(N\mathcal{M}(m))$. The Exponential NNLMS [27] sets $f_i(\boldsymbol{\alpha}) = \alpha_i^{\gamma-1}$ in (5), which leads to $w_i(n) = |\alpha_i(n)|\alpha_i^\gamma(n)$. Evaluating $\alpha_i^\gamma(n)$ has complexity of $O(\mathcal{M}(m)\log m)$ because it uses exponential and logarithm elementary functions. The Exponential NNLMS thus has complexity of $O(N\mathcal{M}(m)\log m)$. The IP-NNLMS update (10) sets $w_i(n) = \frac{\alpha_i(n)}{|\alpha_i(n)|+\epsilon}$, and thus has complexity of $O(N\mathcal{M}(m))$ since evaluating each $f_i(n)$ has complexity of $O(\mathcal{M}(m))$. Hence, IP-NNLMS has the same order of complexity as NNLMS, while the complexity of the Exponential NNLMS algorithm is larger than that of the other two by a factor of $\log m$. Then, IP-NNLMS addresses the two important NNLMS limitations described above without a significant increase in computational complexity.

Fig. 2 presents the computation time required for the calculation of the weight updates of NNLMS, IP-NNLMS and Exponential NNLMS for 10^6 iterations on a laptop with Matlab as a function of the number of filter coefficients N. This experiment shows that complexity of these algorithms increases linearly with N, with a factor that is significantly larger for Exponential NNLMS.

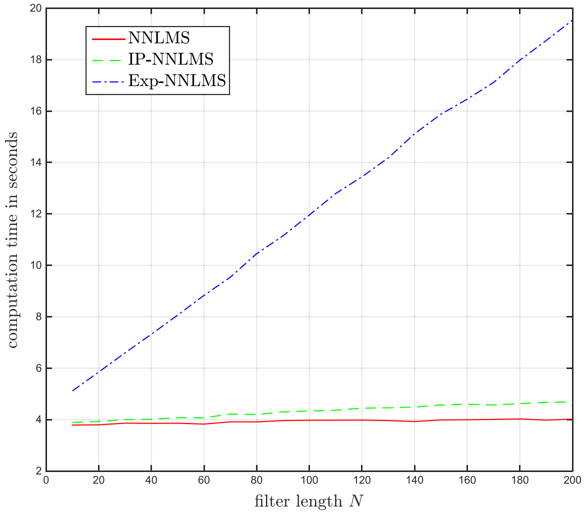


Fig. 2. Computation time in seconds of NNLMS, IP-NNLMS and Exponential NNLMS for 10^6 iterations as a function of the number of filter coefficients N.

4. Stochastic behavior study

Direct analysis of the IP-NNLMS update relation for deriving stochastic behavior models is made difficult by the nonlinearity of the correction term. Even for the Proportionate LMS algorithm, the analysis was conducted by considering a fixed reweighting matrix [31]. In this section, we shall derive analytical models for the IP-NNLMS algorithm. The analysis requires hypotheses and approximations for feasibility. Simulations to be presented in Section 5 will validate the proposed models.

4.1. Weight error equation

Define the weight error vector $\boldsymbol{v}(n)$ as the difference between the estimated weight $\boldsymbol{\alpha}(n)$ and the unconstrained optimum $\boldsymbol{\alpha}^\star$, that is,

$$\boldsymbol{v}(n)=\boldsymbol{\alpha}(n)-\boldsymbol{\alpha}^\star. \tag{13}$$

Subtracting $\boldsymbol{\alpha}^\star$ from both sides of the weight update relation (10) yields

$$\boldsymbol{v}(n+1)=\boldsymbol{v}(n)+\eta\boldsymbol{D}_w(n)\boldsymbol{x}(n)e(n) \tag{14}$$

The estimation error $e(n)$ can also be expressed in terms of $\boldsymbol{v}(n)$ as follows:

$$e(n)=y(n)-\boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n)=z(n)-\boldsymbol{v}^\top(n)\boldsymbol{x}(n). \tag{15}$$

Using (14) and (15), the weight error update relation can then be expressed as

$$\boldsymbol{v}(n+1)=\boldsymbol{v}(n)-\eta\boldsymbol{D}_w(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)+\eta\boldsymbol{D}_w(n)\boldsymbol{x}(n)z(n) \tag{16}$$

4.2. Statistical hypotheses and approximations

The analysis is performed for an input signal $x(n)$ stationary, zero-mean and Gaussian with autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{x}}=E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\}$, and under the following statistical assumptions and approximations:

(A1) The modeling error $z(n)$ is zero-mean, independent and identically distributed (i.i.d.), Gaussian, and statistically independent of any other signal.

(A2) $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$ and $\boldsymbol{v}(n)$ are statistically independent. This assumption is usually termed modified independence assumption (MIA), and is required for mathematical tractability. MIA is a milder assumption than the often employed independence assumption (IA), which assumes statistical independence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$ [2].

(A3) In evaluating higher order (greater than 2) moments of $\boldsymbol{v}(n)$, we approximate $\boldsymbol{v}(n)\boldsymbol{v}^\top(n)$ by its mean value $\boldsymbol{K}(n)=E\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\}$. This approximation preserves the mean (in odd order moments) and fluctuation behaviors of $\boldsymbol{v}(n)$ while keeping the mathematical problem tractable. It has been previously employed with success in analyses of adaptive algorithms with weight updates that are nonlinear with respect to the weight vector [33].

The simulation results will show that Assumptions A1 and A2 and Approximation A3 lead to analytical models that are accurate enough in predicting the behavior of the algorithms for design purposes.

4.3. Mean weight behavior analysis

Taking the expected value on both sides of (16) and using A1 we note that the last term on its RHS is equal to zero. The evaluation of $E\{\boldsymbol{D}_w(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\}$ requires approximations due to the strong nonlinearity on $\boldsymbol{v}(n)$. An approximation that leads to mathematical tractability without significantly compromising accuracy is obtained by using a zeroth-order approximation of the scaling factors $w_i(n)$ about $E\{\alpha_i(n)\}$. Thus, we write

$$w_i(n)=\frac{\alpha_i(n)}{|\alpha_i(n)|+\epsilon}\approx\frac{E\{\alpha_i(n)\}}{|E\{\alpha_i(n)\}|+\epsilon},\qquad\text{for } i=1,\ldots,N. \tag{17}$$

Using this approximation in the expectation of (16) yields

$$E\{\boldsymbol{v}(n+1)\}=E\{\boldsymbol{v}(n)\}-\eta\boldsymbol{D}\left\{\frac{E\{\alpha_1(n)\}}{|E\{\alpha_1(n)\}|+\epsilon},\ldots,\frac{E\{\alpha_N(n)\}}{|E\{\alpha_N(n)\}|+\epsilon}\right\}E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\}. \tag{18}$$

Using (13) and A2, the mean weight error behavior can finally be described by

$$E\{\boldsymbol{v}(n+1)\}=E\{\boldsymbol{v}(n)\}-\eta\boldsymbol{D}\left\{\frac{E\{v_1(n)+\alpha_1^\star\}}{|E\{v_1(n)+\alpha_1^\star\}|+\epsilon},\ldots,\frac{E\{v_N(n)+\alpha_N^\star\}}{|E\{v_N(n)+\alpha_N^\star\}|+\epsilon}\right\}\boldsymbol{R}_{\boldsymbol{x}}E\{\boldsymbol{v}(n)\}. \tag{19}$$

Monte Carlo simulations reported in Section 5, and depicted in Fig. 3, illustrate the consistency of this model.

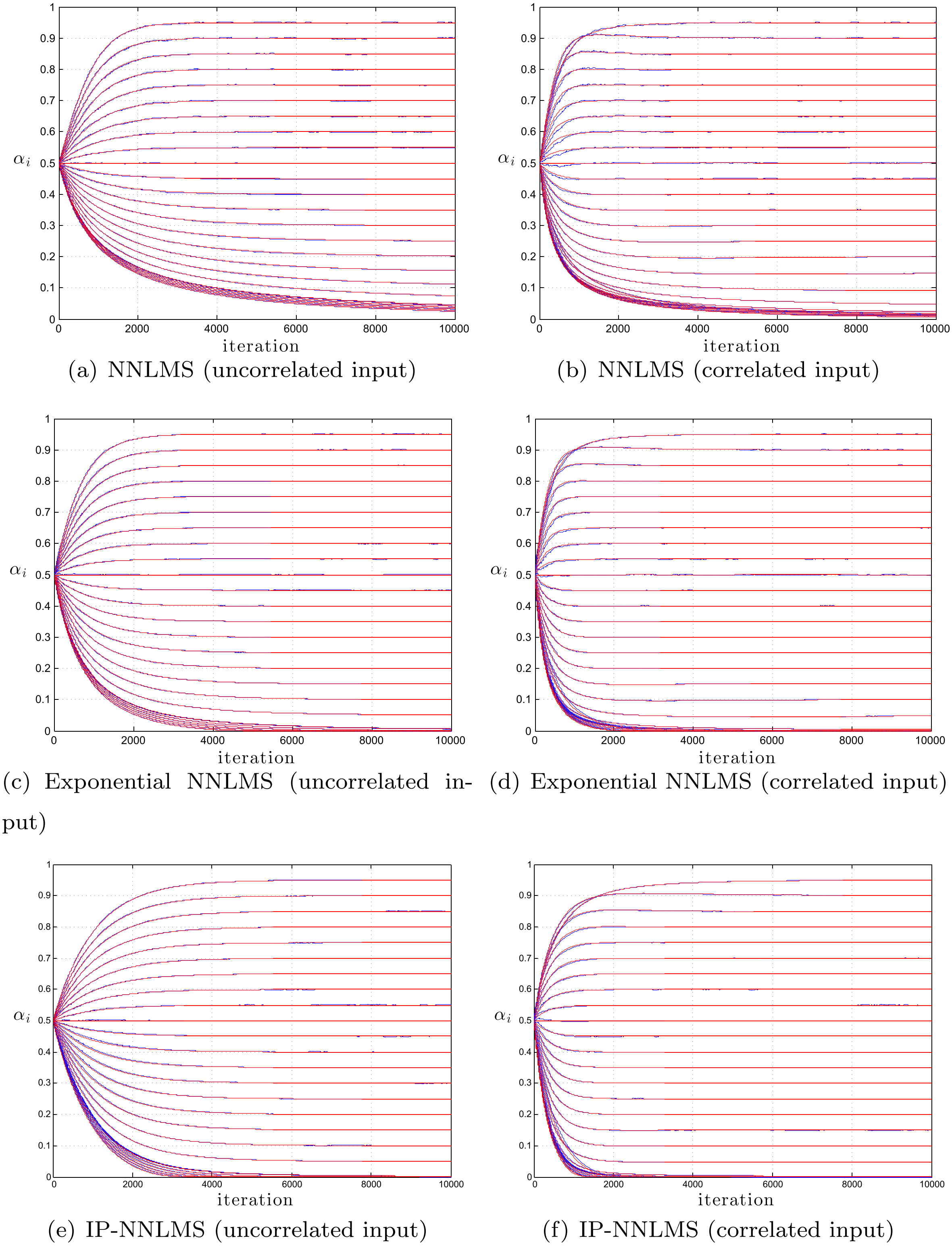
Model (19) is nonlinear with respect to $\boldsymbol{v}(n)$, which makes derivation of a condition for stability difficult. Interested readers are referred to [26] for a related analysis done for the NNLMS algorithm and based on a fixed-point equation analysis.

4.4. Transient excess mean-square error model

The objective of this section is to derive a model for the transient mean-square error (MSE) behavior of the algorithm. Squaring (15) and using A1 and A2 yields following expression for the MSE:

$$\begin{aligned}E\{e^2(n)\}&=E\{(z(n)-\boldsymbol{v}^\top(n)\boldsymbol{x}(n))(z(n)-\boldsymbol{v}^\top(n)\boldsymbol{x}(n))\}\\&=\sigma_z^2+\mathrm{tr}\{\boldsymbol{R}_{\boldsymbol{x}}\,\boldsymbol{K}(\mathrm{n})\}.\end{aligned} \tag{20}$$

where $\boldsymbol{K}(n)=E\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\}$ is the autocorrelation matrix of the weight error vector. The second term in the RHS of (20) incorporates the excess mean-square error (EMSE), which is due to the vector $\boldsymbol{\alpha}(n)-\boldsymbol{\alpha}^o$, and part of the minimum MSE as, the latter is composed by the noise power ${\sigma_z}^2$ and by the power contributed by the vector difference $\boldsymbol{\alpha}^\star-\boldsymbol{\alpha}^o$, which affects $\boldsymbol{v}(n)$. As the analytical determination of $\boldsymbol{\alpha}^o$ is not possible except for the case of white



(a) NNLMS (uncorrelated input)

(b) NNLMS (correlated input)

(c) Exponential NNLMS (uncorrelated input)

(d) Exponential NNLMS (correlated input)

(e) IP-NNLMS (uncorrelated input)

(f) IP-NNLMS (correlated input)

Fig. 3. Mean weight behavior of NNLMS, IP-NNLMS and Exponential NNLMS algorithms. *Left column*: uncorrelated input. *Right column*: correlated input with $\tau = 0.5$. Red curves were obtained with theoretical models, and blue curves were obtained by averaging over 100 Monte Carlo simulations. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

input signals, we derive a model for the behavior of $\zeta(n) = \text{tr}\{\boldsymbol{R}_x \boldsymbol{K}(n)\}$. Thus, in the sequel we determine a recursive update equation for $\boldsymbol{K}(n)$.

Although the zeroth-order approximation (17) may be sufficient for deriving accurate mean weight behavior models, it is insufficient to accurately characterize the second-order behavior of the algorithm. To proceed with the analysis, we then approximate the nonlinear scaling term (12) by its first-order Taylor series about $E\{\alpha_i(n)\}$

$$w_i(n) \approx \frac{E\{\alpha_i(n)\}}{|E\{\alpha_i(n)\}| + \epsilon} + \nabla w(E\{\alpha_i(n)\})(\alpha_i(n) - E\{\alpha_i(n)\}) = r_i(n) + s_i(n)v_i(n) \tag{21}$$

where

$$r_i(n) = \frac{E\{v_i(n)\} + \alpha_i^\star}{|E\{v_i(n)\} + \alpha_i^\star| + \epsilon} - \nabla w(E\{v_i(n)\} + \alpha_i^\star)E\{v_i(n)\} \tag{22}$$

$$s_i(n) = \nabla w(E\{v_i(n)\} + \alpha_i^\star). \tag{23}$$

Defining the diagonal matrix $\boldsymbol{D_s}(n) = \boldsymbol{D}\{s_1(n), \ldots, s_N(n)\}$, (21) can be written in vector form as

$$\boldsymbol{w}(n) \approx \boldsymbol{r}(n) + \boldsymbol{D_s}(n)\boldsymbol{v}(n). \tag{24}$$

Post-multiplying (14) by its transpose, using (24), taking the expected value, and using A1–A3, yields

$$\begin{aligned}\boldsymbol{K}(n+1) = \boldsymbol{K}(n) &- \eta(\boldsymbol{P}_1(n)\boldsymbol{K}(n) + \boldsymbol{K}(n)\boldsymbol{P}_1^\top(n)) \\ &- \eta(\boldsymbol{P}_5(n)\boldsymbol{K}(n) + \boldsymbol{K}(n)\boldsymbol{P}_5^\top(n)) \\ &+ \eta^2(\boldsymbol{P}_6(n) + \boldsymbol{P}_7(n) + \boldsymbol{P}_7^\top(n) + \boldsymbol{P}_8(n)) \\ &+ \eta^2\sigma_z^2(\boldsymbol{P}_2(n) + \boldsymbol{P}_3(n) + \boldsymbol{P}_3^\top(n) + \boldsymbol{P}_4(n))\end{aligned} \tag{25}$$

with

$$\boldsymbol{P}_1(n) = E\{\boldsymbol{D_x}(n)\boldsymbol{r}(n)\boldsymbol{x}^\top(n)\} = \boldsymbol{D_r}(n)\boldsymbol{R_x} \tag{26}$$

$$\boldsymbol{P}_2(n) = E\{\boldsymbol{D_x}(n)\,\boldsymbol{r}(n)\boldsymbol{r}^\top(n)\boldsymbol{D_x}(n)\} = \boldsymbol{D_r}(n)\boldsymbol{R_x}\,\boldsymbol{D_r}(n) \tag{27}$$

$$\boldsymbol{P}_3(n) = E\{\boldsymbol{D_x}(n)\boldsymbol{r}(n)\boldsymbol{v}^\top(n)\boldsymbol{D_s}(n)\boldsymbol{D_x}(n)\} \approx \boldsymbol{D_r}(n)\boldsymbol{R_x}E\{\boldsymbol{D_v}(n)\}\,\boldsymbol{D_s}(n) \tag{28}$$

$$\begin{aligned}\boldsymbol{P}_4(n) = E\{\boldsymbol{D_x}(n)\,\boldsymbol{D_s}(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D_s}(n)\boldsymbol{D_x}(n)\} \approx \boldsymbol{D_s}(n)(\boldsymbol{R_x}\bigcirc\boldsymbol{K}(n)) \\ \boldsymbol{D_s}(n)\end{aligned} \tag{29}$$

$$\boldsymbol{P}_5(n) = \{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D_x}(n)\boldsymbol{D_s}(n)\} = \mathrm{diag}\{\boldsymbol{R_x}\ \mathrm{E}\{\boldsymbol{v}(\mathrm{n})\}\}\boldsymbol{D_s}(\mathrm{n}) \tag{30}$$

$$\begin{aligned}\boldsymbol{P}_6(n) = E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D_x}(n)\boldsymbol{r}(n)\boldsymbol{r}^\top(n)\boldsymbol{D_x}(n)\,\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\} \approx \boldsymbol{D_r}(n)\boldsymbol{Z}(n) \\ \boldsymbol{D_r}(n)\end{aligned} \tag{31}$$

$$\begin{aligned}\boldsymbol{P}_7(n) = E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D_x}(n)\,\boldsymbol{r}(n)\boldsymbol{v}^\top(n)\boldsymbol{D_s}(n)\boldsymbol{D_x}(n)\,\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\} \approx \boldsymbol{D_r}(n) \\ \boldsymbol{Z}(n)E\{\boldsymbol{D_v}(n)\}\,\boldsymbol{D_s}(n)\end{aligned} \tag{32}$$

$$\begin{aligned}\boldsymbol{P}_8(n) &= E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D_x}(n)\boldsymbol{D_s}(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D_s}(n)\boldsymbol{D_x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\} \\ &\approx \boldsymbol{D_s}(n)(\boldsymbol{Z}(n)\bigcirc\boldsymbol{K}(n))\boldsymbol{D_s}(n)\end{aligned} \tag{33}$$

where $\boldsymbol{D_r}(n) = \boldsymbol{D}\{r_1(n), \ldots, r_N(n)\}$, the symbol $\circ$in (30) denotes the Hadamard product, and $\boldsymbol{Z}(n)$ in (31)–(33) is given by

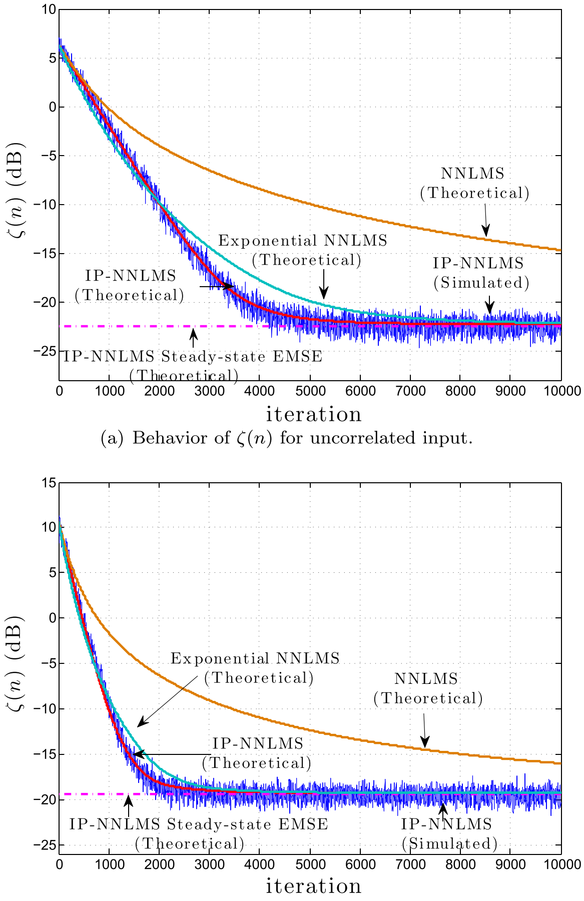
$$\boldsymbol{Z}(n) = 2\,\boldsymbol{R_x}\,\boldsymbol{K}(n)\boldsymbol{R_x} + \mathrm{tr}\{\boldsymbol{R_x}\,\boldsymbol{K}(\mathrm{n})\}\,\boldsymbol{R_x}. \tag{34}$$

The derivation of the expectations in (26)–(33) is lengthy but straightforward using A1–A3, and the details are omitted to conserve space.

Using (26)–(33) with (25) leads to a recursive model for the transient behavior of $\boldsymbol{K}(n)$. Monte Carlo simulations reported in Section 5, and depicted in Fig. 4, illustrate the consistency of this model.

4.5. Steady-state performance and tracking properties

To characterize the steady-state performance and the tracking properties of the algorithm, we consider in this section the following time-variant unknown parameter vector [34,2]



(a) Behavior of $\zeta(n)$ for uncorrelated input.

(b) Behavior of $\zeta(n)$ for correlated input.

Fig. 4. Learning curves of NNLMS, IP-NNLMS and Exponential NNLMS algorithms. *Left column*: uncorrelated input. *Right column*: correlated input with $\tau = 0.5$.

$$\boldsymbol{\alpha}^\star(n) = \boldsymbol{\alpha}^\star + \boldsymbol{\theta}(n) \tag{35}$$

with

$$\boldsymbol{\theta}(n) = \rho\,\boldsymbol{\theta}(n-1) + \boldsymbol{q}(n) \tag{36}$$

where $-1 < \rho < 1$ and $\boldsymbol{q}(n)$ is a zero-mean i.i.d. sequence with covariance matrix $\boldsymbol{R_q} = \sigma_q^2\boldsymbol{I}$. At steady-state, the covariance matrix of $\boldsymbol{\theta}(n)$ is then given by

$$\boldsymbol{R_\theta} = \frac{1}{1-\rho^2}\,\boldsymbol{R_q}. \tag{37}$$

With the unknown parameter vector (35), the weight error vector $\boldsymbol{v}(n)$ is given by

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^\star(n). \tag{38}$$

Assuming convergence of $\boldsymbol{\alpha}(n)$ to $\boldsymbol{\alpha}(\infty)$, the weight error vector (38) can then be expressed as

$$\begin{aligned}\boldsymbol{v}(n) &= [\boldsymbol{\alpha}(n) - E\{\boldsymbol{\alpha}(\infty)\}] + [E\{\boldsymbol{\alpha}(\infty)\} - \boldsymbol{\alpha}^\star(n)] \\ &= [\boldsymbol{\alpha}(n) - E\{\boldsymbol{\alpha}(\infty)\} - \boldsymbol{\theta}(n)] + [E\{\boldsymbol{\alpha}(\infty)\} - \boldsymbol{\alpha}^\star].\end{aligned} \tag{39}$$

We define

$$\boldsymbol{v}'(n) = \boldsymbol{\alpha}(n) - E\{\boldsymbol{\alpha}(\infty)\} - \boldsymbol{\theta}(n) \tag{40}$$

and note that the second term on the RHS of (39) is equal to $E\{\boldsymbol{v}(\infty)\}$ since $E\{\boldsymbol{\theta}(n)\} = 0$. Then, the estimation error at instant n can be written as

$$e(n) = z(n) - \boldsymbol{v}'^{\top}(n)\boldsymbol{x}(n) - E\{\boldsymbol{v}^{\top}(\infty)\}\,\boldsymbol{x}(n). \tag{41}$$

Hence, $\zeta(n) = E\{e^2(n)\} - \sigma_z^2$ can be expressed as

$$\zeta(n) = \underbrace{E\{[\boldsymbol{x}^{\top}(n)\boldsymbol{v}'(n)]^2\}}_{\zeta'(n)} + \underbrace{\mathrm{tr}\{\boldsymbol{R}_{\boldsymbol{x}}\mathrm{E}\{\boldsymbol{v}(\infty)\}\mathrm{E}\{\boldsymbol{v}^{\top}(\infty)\}\}}_{\zeta^{\infty}} + 2E\{\boldsymbol{v}'^{\top}(n)\}\,\boldsymbol{R}_{\boldsymbol{x}}E\{\boldsymbol{v}(\infty)\}. \tag{42}$$

To determine $\lim_{n\to\infty}\zeta(n)$ we note that the second term ζ^{∞} on the RHS of (42) is deterministic. Also, the third term vanishes since $\lim_{n\to\infty}E\{\boldsymbol{v}'(\infty)\} = 0$. Then, we need to evaluate $\lim_{n\to\infty}\zeta'(n)$. Subtracting (40) at $n+1$ from the same expression at n and using (10) and (36) we have

$$\boldsymbol{v}'(n+1) = \boldsymbol{v}'(n) + \eta\,\boldsymbol{D}_w(n)\boldsymbol{x}(n)e(n) + (1-\rho)\boldsymbol{\theta}(n) - \boldsymbol{q}(n+1). \tag{43}$$

For the analysis that follows, we group the weights $w_i(n)$ into two distinct sets. The set $\mathcal{S}_+$ contains the indices of the weights that converge in the mean to positive values, namely,

$$\mathcal{S}_+ = \{i\colon E\{w_i(\infty)\} > 0\}. \tag{44}$$

The set $\mathcal{S}_0$ contains the indices of the weights that converge in the mean to zero, namely,

$$\mathcal{S}_0 = \{i\colon E\{w_i(\infty)\} = 0\}. \tag{45}$$

Considering that the nonnegativity constraint is always satisfied at steady-state, if $E\{w_i(\infty)\} = 0$ for all $i \in \mathcal{S}_0$, then $\alpha_i(\infty) = 0$ for all $i \in \mathcal{S}_0$ and for any realization. This implies from (40) that:

$$v_i'(\infty) = -\,\theta_i(\infty), \quad \text{for all } i \in \mathcal{S}_0. \tag{46}$$

Now let $\bar{\boldsymbol{D}}_w^{-1}(n)$ be the diagonal matrix with entries

$$[\bar{\boldsymbol{D}}_w^{-1}(n)]_{ii} = \begin{cases} \dfrac{1}{w_i(n)}, & i \in \mathcal{S}_+ \\ 0, & i \in \mathcal{S}_0 \end{cases} \tag{47}$$

and $\bar{\boldsymbol{I}}$ the diagonal matrix defined as

$$[\bar{\boldsymbol{I}}]_{ii} = \begin{cases} 1, & i \in \mathcal{S}_+ \\ 0, & i \in \mathcal{S}_0. \end{cases} \tag{48}$$

With these matrices, we have

$$\bar{\boldsymbol{D}}_w^{-1}(n)\boldsymbol{D}_w(n) = \bar{\boldsymbol{I}}. \tag{49}$$

Now, evaluating the weighted square-norm $\|\cdot\|^2_{\bar{\boldsymbol{D}}_{w(n)}^{-1}}$ of both sides of (43), taking the expected value of both sides, and taking the limit as $n \to \infty$ we have

$$\begin{aligned} &\lim_{n\to\infty} E\left\{\|\boldsymbol{v}'(n+1)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} \\ &\quad = \lim_{n\to\infty}\Bigg(E\left\{\|\boldsymbol{v}'(n)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} \\ &\qquad + 2\eta E\{\boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{x}(n)e(n)\} + \eta^2 E\left\{\boldsymbol{x}^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{D}_w(n)\boldsymbol{x}(n)e^2(n)\right\} \\ &\qquad + (1-\rho)^2 E\left\{\|\boldsymbol{\theta}(n)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} + E\left\{\|\boldsymbol{q}(n+1)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} \\ &\qquad + 2(1-\rho)E\left\{\boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{D}}_w^{-1}(n)\boldsymbol{\theta}(n)\right\} + 2\eta(1-\rho) \\ &\qquad E\{e(n)\boldsymbol{x}^{\top}(n)\bar{\boldsymbol{I}}\,\boldsymbol{\theta}(n)\}\Bigg). \end{aligned} \tag{50}$$

Assuming convergence, the following relation is valid at steady-state:

$$\lim_{n\to\infty} E\left\{\|\boldsymbol{v}'(n+1)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} = \lim_{n\to\infty} E\left\{\|\boldsymbol{v}'(n)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\}. \tag{51}$$

Before evaluating all the terms on the RHS of (50), we calculate the cross-covariance matrix $E\{\boldsymbol{v}'(n)\boldsymbol{\theta}^{\top}(n)\}$ at steady-state, namely,

$$\boldsymbol{\Gamma} = \lim_{n\to\infty} E\{\boldsymbol{v}'(n)\boldsymbol{\theta}^{\top}(n)\}.$$

Post-multiplying both sides of (43) by $\boldsymbol{\theta}^{\top}(n+1)$, taking expected value under A1 and A2 and taking the limit as $n \to \infty$, we have

$$\begin{aligned} \boldsymbol{\Gamma} &= \lim_{n\to\infty} E\{[\boldsymbol{v}'(n) + \eta\boldsymbol{D}_w(n)\boldsymbol{x}(n)e(n)] \\ &\quad \boldsymbol{\theta}^{\top}(n+1) + [(1-\rho)\boldsymbol{\theta}(n) - \boldsymbol{q}(n+1)]\boldsymbol{\theta}^{\top}(n+1)\} \\ &= [\rho\boldsymbol{\Gamma} - \eta\rho E\{\boldsymbol{D}_w(\infty)\}\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{\Gamma}] + [\rho(1-\rho)\boldsymbol{R}_{\boldsymbol{\theta}} - \boldsymbol{R}_{\boldsymbol{q}}], \end{aligned} \tag{52}$$

which yields

$$\boldsymbol{\Gamma} = \left(\frac{1}{1+\rho}\right)[\rho(\boldsymbol{I} - \eta\,E\{\boldsymbol{D}_w(\infty)\}\boldsymbol{R}_{\boldsymbol{x}}) - \boldsymbol{I}]^{-1}\boldsymbol{R}_{\boldsymbol{q}}. \tag{53}$$

Using (41) and A2, the second term on the RHS of (50) for $n \to \infty$ can be expressed as

$$\begin{aligned} &\lim_{n\to\infty} E\{\boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{x}(n)e(n)\} \\ &\quad = -\lim_{n\to\infty} E\{\boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)\boldsymbol{v}'(n) + \boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)E\{\boldsymbol{v}(\infty)\}\} \\ &\quad = -\,\zeta'(\infty) + \lim_{n\to\infty} E\{\boldsymbol{v}'^{\top}(n)(\boldsymbol{I} - \bar{\boldsymbol{I}})\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)\boldsymbol{v}'(n)\} = -\,\zeta'(\infty) \\ &\qquad - \mathrm{tr}\{\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{\Gamma}(\boldsymbol{I} - \bar{\boldsymbol{I}})\} \end{aligned} \tag{54}$$

where we used $v_i'(\infty) = -\,\theta_i(\infty)$ for $i \in \mathcal{S}_0$ (see (46)), and $E\{\boldsymbol{v}'(\infty)\} = 0$. Consider now the third term on the RHS of (50). As this term is of second order in η and we are interested on its value at steady-state, we approximate its evaluation by disregarding the correlation between $e^2(n)$ and $\boldsymbol{x}^{\top}(n)\,\bar{\boldsymbol{I}}\,\boldsymbol{D}_w(n)\boldsymbol{x}(n)$ for $n \to \infty$. This approximation is reasonable because $\boldsymbol{x}^{\top}(n)\,\bar{\boldsymbol{I}}\,\boldsymbol{D}_w(n)\boldsymbol{x}(n)$ corresponds to the energy of the input vector components exciting the nonzero coefficients (here assumed to be much larger than ϵ so that the corresponding w_i approach 1 per (12)), and this part of the input vector tends to be uncorrelated with the estimation error is steady-state. Under this approximation and using A1 and A2,

$$\begin{aligned} \lim_{n\to\infty} E\{\boldsymbol{x}^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{D}_w(n)\boldsymbol{x}(n)e^2(n)\} &\approx \mathrm{tr}\{\mathrm{E}\{\boldsymbol{D}_w(\infty)\}\boldsymbol{R}_{\boldsymbol{x}}\} \\ &\qquad (\sigma_z^2 + \zeta'(\infty) + \zeta^{\infty}). \end{aligned} \tag{55}$$

The fourth and the fifth terms in the RHS of (50) can be directly expressed as

$$\lim_{n\to\infty} E\left\{\|\boldsymbol{\theta}(n)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} = \mathrm{tr}\{\mathrm{E}\{\bar{\boldsymbol{D}}_w^{-1}(\infty)\}\,\boldsymbol{R}_{\boldsymbol{\theta}}\} \tag{56}$$

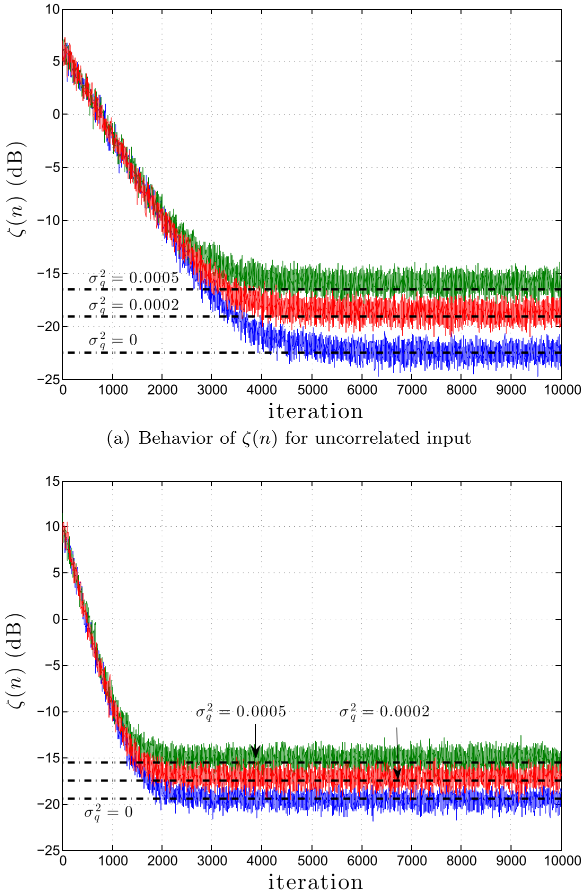
$$\lim_{n\to\infty} E\left\{\|\boldsymbol{q}(n+1)\|^2_{\bar{\boldsymbol{D}}_w^{-1}(n)}\right\} = \mathrm{tr}\{\mathrm{E}\{\bar{\boldsymbol{D}}_w^{-1}(\infty)\}\boldsymbol{R}_{\boldsymbol{q}}\} \tag{57}$$

where we have used the independence of both $\boldsymbol{\theta}(n)$ and $\boldsymbol{q}(n)$ with respect to $\bar{\boldsymbol{D}}_w^{-1}(n)$ as $n \to \infty$, as the latter becomes approximately time invariant for the nonzero coefficients. Using the same assumptions and approximations as above, the sixth term on the RHS of (50) can be written as

$$\lim_{n\to\infty} E\{\boldsymbol{v}'^{\top}(n)\bar{\boldsymbol{D}}_w^{-1}(n)\boldsymbol{\theta}(n)\} = \mathrm{tr}\{\mathrm{E}\{\bar{\boldsymbol{D}}_w^{-1}(\infty)\}\,\boldsymbol{\Gamma}\}, \tag{58}$$

and, finally, the last term is given by

$$\begin{aligned} \lim_{n\to\infty} E\{e(n)\boldsymbol{x}^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{\theta}(n)\} &= \lim_{n\to\infty} -\,E\{\boldsymbol{v}^{\top}(n)\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)\bar{\boldsymbol{I}}\boldsymbol{\theta}(n)\} \\ &= -\,\mathrm{tr}\{\bar{\boldsymbol{I}}\boldsymbol{\Gamma}\boldsymbol{R}_{\boldsymbol{x}}\} \end{aligned} \tag{59}$$



(a) Behavior of $\zeta(n)$ for uncorrelated input

(b) Behavior of $\zeta(n)$ for correlated input

Fig. 5. Learning curves of NNLMS, IP-NNLMS and Exponential NNLMS algorithm for a time-varying system. *Left column*: uncorrelated input. *Right column*: correlated input with $\tau = 0.5$. The dash-dot lines represent the steady-state $\zeta(n)$ values calculated with (63). Solid learning curves were obtained by averaging over 100 Monte Carlo simulations.

Replacing (51)–(59) into (50) and solving the equation with respect to $\zeta'(\infty)$, we have

$$\zeta'(\infty) = \frac{\eta\,\mathrm{tr}\{\mathrm{E}\{\boldsymbol{D}_w(\infty)\}\boldsymbol{R}_x\}(\sigma_z^2 + \zeta^\infty) + \beta + \eta^{-1}\gamma}{2 - \eta\,\mathrm{tr}\{\mathrm{E}\{\boldsymbol{D}_w(\infty)\}\,\boldsymbol{R}_x\}} \tag{60}$$

where

$$\beta = \mathrm{tr}\{\boldsymbol{R}_x\boldsymbol{\Gamma}(\bar{\boldsymbol{I}} - \boldsymbol{I})\} \tag{61}$$

$$\gamma = \mathrm{tr}\{2\,\mathrm{E}\{\bar{\boldsymbol{D}}_w^{-1}(\infty)\}[\boldsymbol{R}_q + (1-\rho)\,\boldsymbol{\Gamma}]\} - 2\eta(1-\rho)\mathrm{tr}\{\bar{\boldsymbol{I}}\boldsymbol{\Gamma}\boldsymbol{R}_x\}. \tag{62}$$

with $\boldsymbol{\Gamma}$ given by (52). Finally, using (42), we obtain the steady-state result:

$$\zeta(\infty) = \frac{\eta\,\mathrm{tr}\{\mathrm{E}\{\boldsymbol{D}_w(\infty)\}\,\boldsymbol{R}_x\}(\sigma_z^2 + \zeta^\infty) + \beta + \eta^{-1}\gamma}{2 - \eta\,\mathrm{tr}\{\mathrm{E}\{\boldsymbol{D}_w(\infty)\}\,\boldsymbol{R}_x\}} + \zeta^\infty. \tag{63}$$

To obtain the steady-state performance in a stationary environment, it is sufficient to set ρ and $\boldsymbol{R}_q$ (and thus $\boldsymbol{\Gamma}$) to zero in (61) and (62).

5. Simulation results

This section presents simulation results to validate the derived theoretical models. The simulation curves were obtained by averaging over 100 Monte Carlo runs.

5.1. Model validation in a stationary environment

Consider the application of NNLMS-type algorithms for the online identification of the 30-coefficient sparse impulse response

$$\alpha_i^\star = \begin{cases} 1 - 0.05\,i & i = 1, \ldots, 20 \\ 0 & i = 21, \ldots, 25 \\ -0.01\,(i-25) & i = 26, \ldots, 30. \end{cases} \tag{64}$$

where the last five negative coefficients were included in order to activate the nonnegativity constraints. The input signal $x(n)$ was generated with the first-order AR model

$$x(n) = \tau\,x(n-1) + \xi(n), \tag{65}$$

where $\xi(n)$ was an i.i.d. zero-mean Gaussian sequence with variance $\sigma_\xi^2 = \sqrt{1-\tau^2}$ so that $\sigma_x^2 = 1$, and independent of any other signal. We considered the two settings $\tau = 0$ and $\tau = 0.5$. The former corresponds to an uncorrelated input, while the latter results in a correlated input. The additive noise $z(n)$ was zero-mean i.i.d. Gaussian with variance $\sigma_z^2 = 0.01$. The filter coefficients were initialized with $\alpha_i(0) = 0.5$ for all $i = 1, \ldots, N$, with N=30.

Besides verifying the IP-NNLMS model accuracy, we also compare the performance of IP-NNLMS with the original NNLMS and the Exponential NNLMS algorithms. Their step sizes were set to $\eta_{\text{NNLMS}} = 0.002$, $\eta_{\text{ENNLMS}} = 0.0018$, and $\eta_{\text{IPNNLMS}} = 0.001$, respectively, so that they approximatively reach the same steady-state performance. The exponential parameter γ of the Exponential NNLMS algorithm was set to $\gamma = \frac{3}{7}$. The parameter ϵ in the IP-NNLMS algorithm was set to 0.01.

The mean weight behaviors of these algorithms are shown in Fig. 3. The theoretical curves for the IP-NNLMS algorithm were obtained with model (19). Those of the other two algorithms were obtained with the models derived in [26,27]. All the theoretical curves match well those obtained with Monte Carlo simulations. As already mentioned, the original NNLMS algorithm is characterized by low convergence rates for small coefficients. Several NNLMS coefficients in the active set had not converged to zero even after a long time. Compared to the NNLMS algorithm, the Exponential NNLMS and the IP-NNLMS algorithms have more balanced convergence rates for all coefficients. The Exponential NNLMS, however, has a higher computational complexity than the IP-NNLMS. Fig. 4 provides the behavior of $\zeta(n)$ for the three algorithms. For the sake of clarity, only the theoretical learning curves are represented for the NNLMS and Exponential NNLMS algorithms. They were obtained from [26,27]. The transient learning curves of the IP-NNLMS algorithm were obtained using (25). The steady-state performance was estimated by (63). All these results show that the proposed algorithm has a performance that is at least comparable to those of the other algorithms, and that the theoretical model accurately predict its performance.

5.2. Tracking performance in a non-stationary environment

Consider the time-varying system with coefficients defined by the time-variant relation (35). The mean values of the coefficients were set as in (64). The parameter ρ of the random perturbation in (35) was set to $\rho = 0.5$. The random vector $\boldsymbol{q}(n)$ had a covariance matrix $\boldsymbol{R}_q = \sigma_q^2\,\boldsymbol{I}$, with $\sigma_q^2 = 0.2 \times 10^{-4}$ and $\sigma_q^2 = 5 \times 10^{-4}$ successively. The step size was set to $\eta = 10^{-5}$. All the other parameters were not changed compared to the previous experiment. The results for uncorrelated and correlated inputs are shown in Fig. 5. As expected, it can be observed that the steady-state estimation error increases with the variance $\sigma_q{}^2$.

5.3. Application to sparse online system identification

Consider the online system identification problem consisting of minimizing the MSE criterion with ℓ_1-norm regularization

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\} + \nu \parallel \boldsymbol{\alpha} \parallel_1 \tag{66}$$

with the parameter ν providing a tradeoff between data fidelity and solution sparsity. This problem can be rewritten as a standard NNLMS problem by introducing two $N \times 1$ nonnegative vectors $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ such that

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \quad \text{with} \quad \boldsymbol{\alpha}^+ \geq 0 \quad \text{and} \quad \boldsymbol{\alpha}^- \geq 0. \tag{67}$$

Let us define the vectors $\tilde{\boldsymbol{\alpha}} = \mathrm{col}\{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-\}$ and $\tilde{\boldsymbol{x}}(n) = \mathrm{col}\{\boldsymbol{x}(n), -\boldsymbol{x}(n)\}$ where the operator $\mathrm{col}\{\cdot\}$ stacks its vector arguments on top of each other. The problem (66) can then be reformulated as

$$\begin{aligned} \tilde{\boldsymbol{\alpha}}^o = \arg\min_{\tilde{\boldsymbol{\alpha}}} \frac{1}{2} E\{[y(n) - \tilde{\boldsymbol{x}}^\top(n)\tilde{\boldsymbol{\alpha}}]^2\} + \nu\, \mathbf{1}^\top \tilde{\boldsymbol{\alpha}} \\ \text{subject to } \tilde{\boldsymbol{\alpha}} \geq 0 \end{aligned} \tag{68}$$

where $\mathbf{1}$ is an all-one vector of length $2N$. Problem (68) is a quadratic problem with nonnegativity constraint with respect to $\tilde{\boldsymbol{\alpha}}$. Note that although there are an infinite number of the decompositions satisfying (67), the regularization term $\mathbf{1}^\top \tilde{\boldsymbol{\alpha}}$ forces (68) to admit a unique solution. Using the proposed IP-NNLMS algorithm,

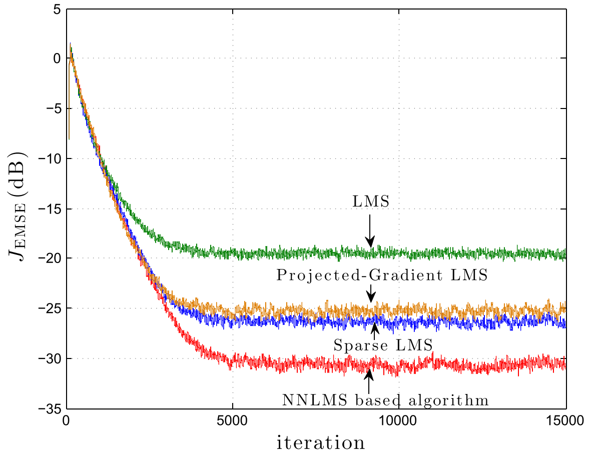
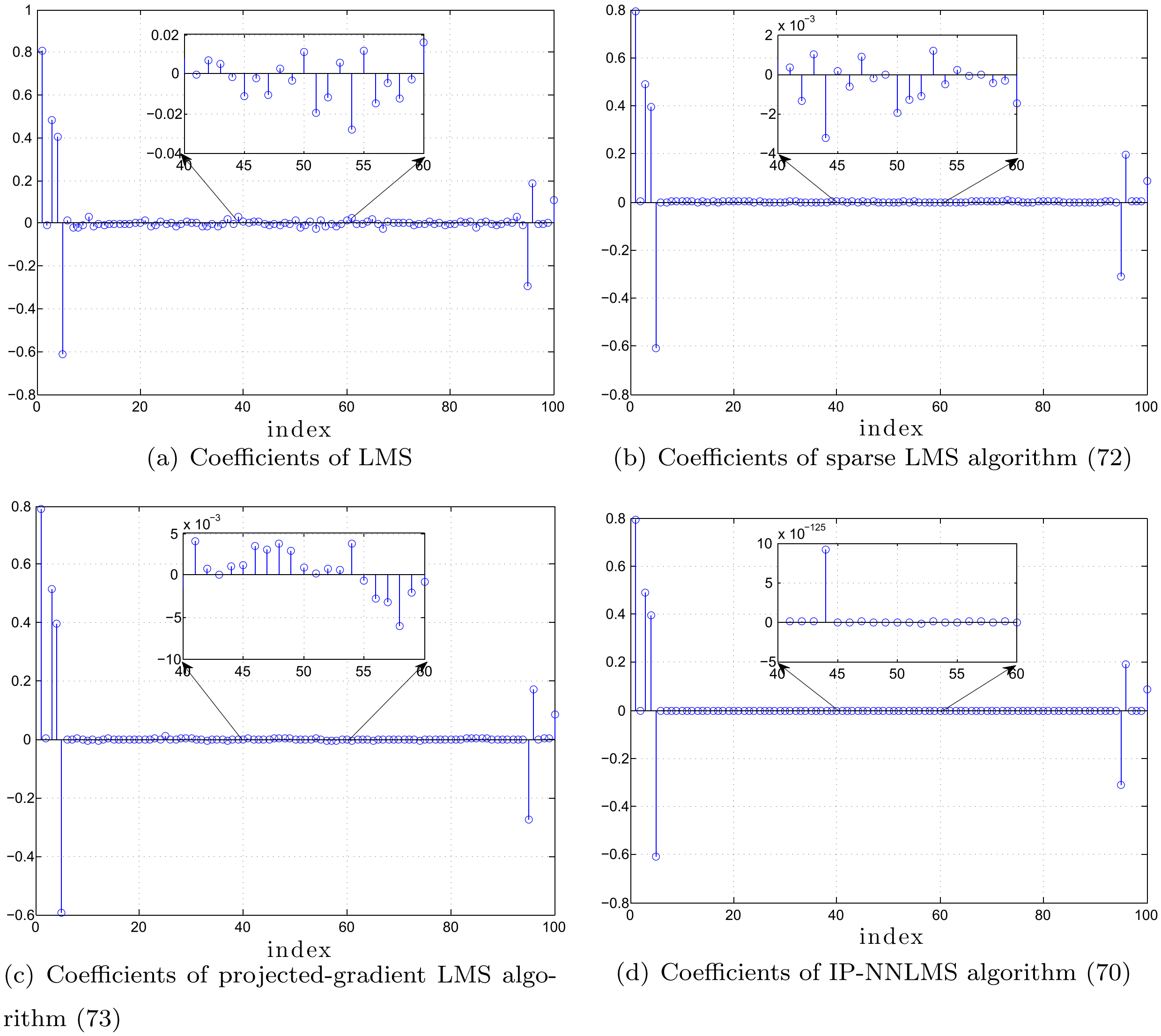


Fig. 6. EMSE learning curves for the four compared algorithms. The advantage of the NNLMS-based algorithm is evident from these results.



(a) Coefficients of LMS

(b) Coefficients of sparse LMS algorithm (72)

(c) Coefficients of projected-gradient LMS algorithm (73)

(d) Coefficients of IP-NNLMS algorithm (70)

Fig. 7. Filter coefficients corresponding to a single realization. Though all these algorithms converge to the same result in the mean sense, NNLMS based algorithms converge more accurately for each realization, which is a favorable property for practical applications.

problem (68) can be solved in an online manner as follows [35]:

$$\tilde{\boldsymbol{\alpha}}(n+1) = \tilde{\boldsymbol{\alpha}}(n) + \eta \boldsymbol{D}_f(n)\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}}(n)[\tilde{\boldsymbol{x}}(n)e(n) - \nu\, \mathbf{1}] \tag{69}$$

with $\boldsymbol{D}_f(n)$ being the diagonal matrix with ith diagonal entries $f_i(n)$ defined in the form of (9), namely, $f_i(n) = \frac{1}{|\tilde{\alpha}_i(n)\,| + \epsilon}$. The performance of the algorithm can be further improved by considering a reweighted ℓ_1–norm approach, which leads to the algorithm

$$\tilde{\boldsymbol{\alpha}}(n+1) = \tilde{\boldsymbol{\alpha}}(n) + \eta \boldsymbol{D}_f(n)\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}}(n)[\tilde{\boldsymbol{x}}(n)e(n) - \nu\boldsymbol{\gamma}(n)] \tag{70}$$

where the ith entry of $\boldsymbol{\gamma}(n)$ is given by $\gamma_i(n) = \frac{1}{\tilde{\alpha}_i(n) + \mu}$, with μ being a small positive number.

The above algorithm was tested by considering the sparse system of order N=100 defined as follows:

$$\begin{cases} \alpha_1^\star = 0.8, \quad \alpha_3^\star = 0.5, \quad \alpha_4^\star = 0.4, \quad \alpha_5^\star = -\,0.6, \\ \qquad \alpha_{95}^\star = -\,0.3, \quad \alpha_{96}^\star = 0.2, \quad \alpha_{100}^\star = 0.1 \\ \alpha_i^\star = 0, \quad \text{otherwise.} \end{cases} \tag{71}$$

The input signal was generated with the autoregressive model (65) with $\tau = 0.5$. The observation noise $z(n)$ was zero-mean i.i.d. Gaussian with variance $\sigma_z^2 = 0.1$. The algorithm (70), the LMS algorithm, the Sparse LMS algorithm [36] with reweighted ℓ_1–norm regularizer defined as

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\left[\boldsymbol{x}(n)e(n) - \nu\boldsymbol{D}\left\{\frac{\operatorname{sgn}(\alpha_1(n))}{|\alpha_1(n)| + \mu}, \ldots, \frac{\operatorname{sgn}(\alpha_N(n))}{|\alpha_N(n)| + \mu}\right\}\right] \tag{72}$$

and the projected-gradient algorithm given by

$$\tilde{\boldsymbol{\alpha}}(n+1) = \max\{\tilde{\boldsymbol{\alpha}}(n) + \eta[\tilde{\boldsymbol{x}}(n)e(n) - \nu\boldsymbol{\gamma}(n)];\, \mathbf{0}\} \tag{73}$$

where $\max\{\cdot\,;\,\cdot\}$ denotes the component-wise maximum operator applied to its vector arguments, were tested for comparison purpose. The parameters ν and μ in (70) and (72) were set to $\nu = 0.001$ and $\mu = 0.01$, respectively. The parameter ϵ in (70) was set to $\epsilon = 0.01$. The step sizes were set to $\eta = 0.002$ for all the algorithms.

Fig. 6 shows the EMSE learning curves of all the algorithms. The gain from promoting sparsity is clearly shown by the performance of the Sparse LMS and the proposed algorithm. The proposed algorithm shows lower estimation error compared with the others since it encourages small values to converge toward 0. This advantage can be seen in Figs. 7(a)–(d), which depict instantaneous weight values at steady-state for a single realization. The values of filter coefficients $\alpha_i(n)$ with $i = 40, \ldots, 60$ are shown in zoomed-in subfigures for a clearer presentation. The IP-NNLMS algorithm shows a clear advantage in accurately estimating these null-valued coefficients.

6. Conclusion

In this paper, we proposed a new algorithm for solving online learning problems subject to nonnegativity constraint. This algorithm has been designed to address two important limitations of the existing NNLMS algorithm. One limitation is the unbalanced convergence rate for coefficients of different values, which penalizes specially the coefficients in the active set (those that converge to zero). The second limitation is the larger variability of the NNLMS weight updates corresponding to larger coefficients, what may complicate the choice of a suitable step-size. Both issues are addressed through the proposition of a new scaling function for the weight updates. The proposed function is bounded and tends to unity for large coefficient values. Moreover, the proposed nonlinear function has a large derivative for coefficients close to zero, what significantly accelerates the convergence of the coefficients in the active set. We analyzed the algorithm in the mean and mean-square-error senses, and considered the case of time-varying systems to determine its tracking properties. Simulations were conducted to illustrate the performance of the proposed algorithm and to validate the theoretical models. Finally, an application to sparse identification was considered in which the nonnegativity constraint is not a direct requirement of the defined problem.

Acknowledgments

The work of J. Chen was supported in part by NSFC 61573288 and G2015KY0303. The work of J. Bermudez was supported in part by CNPq Grant no. 307071/2013-8 and by CAPES – Proc. 3367-11-8.

References

[1] S. Haykin, Adaptive Filter Theory, 4th ed., Pearson Education, India, 2005.

[2] A.H. Sayed, Adaptive Filters, John Wiley & Sons, New Jersey, 2008.

[3] T.A. Johansen, Constrained and regularized system identification, Model. Identif. Control 79 (2) (1998) 109–116.

[4] A. Yeredor, On the role of constraints in system identification, in: Proceedings of the International Workshop on Total Least Squares and Errors-in-Variables Modeling, Leuven, Belgium, 2006, pp. 46–48.

[5] D. Chen, R. Plemmons, Nonnegativity constraints in numerical analysis, in: Proceedings of the Symposium on the Birth of Numerical Analysis, Leuven Belgium, 2007, pp. 109–140.

[6] R.C. Puetter, T.R. Gosnell, A. Yahil, Digital image reconstruction: deblurring and denoising, Annu. Rev. Astron. Astrophys. 43 (2005) 139–194.

[7] J.M. Bardsley, J.K. Merikoski, R. Vio, The stabilizing properties of nonnegativity constraints in least squares image reconstruction, Int. J. Pure Appl. Math. 43 (1) (2008) 95–109.

[8] K.B. Nakshatralaa, M.K. Mudunurua, A.J. Valocchib, A numerical framework for diffusion-controlled bimolecular-reactive systems to enforce maximum principles and the non-negative constraint, J. Comput. Phys. 253 (15) (2013) 278–307.

[9] N. Keshava, J.F. Mustard, Spectral unmixing, IEEE Signal Process. Mag. 19 (1) (2002) 44–57.

[10] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[11] N. Guan, D. Tao, Z. Luo, B. Yuan, Online nonnegative matrix factorization with robust stochastic approximation, IEEE Trans. Neural Netw. Learn. Syst. 23 (7) (2012) 1087–1099.

[12] R. Zass, A. Shashua, Nonnegative sparse PCA, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, vol. 19, MIT Press, Cambridge, MA, 2007, pp. 1561–1568.

[13] J. Chorowski, J.M. Zurada, Learning understandable neural networks with nonnegative weight constraints, IEEE Trans. Neural Netw. Learn. Syst. 26 (1) (2015) 62–69.

[14] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Select. Topics Signal Process. 1 (4) (2007) 586–597.

[15] L.K. Saul, F. Sha, D.D. Lee, Statistical signal processing with nonnegative constraints, in: Proceedings of the European Conference on Speech Communication and Technology, Geneva, Switzerland, 2003, pp. 1001–1004.

[16] Y. Lin, D.D. Lee, Bayesian regularization and nonnegative deconvolution for room impulse response estimation, IEEE Trans. Signal Process. 54 (3) (2006) 839–847.

[17] R. Zdunek, M.J. Nawrocki, Improved modeling of highly loaded UMTS network with nonnegativity constraints, in: Proceedings of the Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 2006, pp. 1–5.

[18] M.A. Khajehnejad, A.G. Dimakis, W. Xu, B. Hassibi, Sparse recovery of nonnegative signals with minimal expansion, IEEE Trans. Signal Process. 59 (1) (2011) 196–208.

[19] Z. Yang, Y. Xiang, S. Xie, S. Ding, Y. Rong, Nonnegative blind source separation by sparse component analysis based on determinant measure, IEEE Trans. Neural Netw. Learn. Syst. 23 (10) (2012) 1601–1610.

[20] Z. Yang, Y. Xiang, Y. Rong, K. Xie, A convex geometry-based blind source separation method for separating nonnegative sources, IEEE Trans. Neural Netw. Learn. Syst. 26 (8) (2015) 1635–1644.

[21] C.L. Lawson, R.J. Hanson, Solving Least Square Problems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995.

[22] R. Bro, S.De. Jong, A fast non-negativity-constrained least squares algorithm, J. Chemomet. 11 (5) (1997) 393–401.

[23] P.H. Calamai, J.J. Moré, Projected gradient methods for linearly constrained problems, Math. Program. 39 (1) (1987) 93–116.

[24] S. Theodoridis, K. Slavakis, I. Yamada, Adaptive learning in a world of projections: a unifying framework for linear and nonlinear classification and regression tasks, IEEE Signal Process. Mag. 28 (1) (2011) 97–123.

[25] C.J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization, IEEE Trans. Neural Netw. 18 (6) (2007) 1589–1596.

[26] J. Chen, C. Richard, J.C.M. Bermudez, P. Honeine, Nonnegative least-mean square algorithm, IEEE Trans. Signal Process. 59 (11) (2011) 5225–5235.

[27] J. Chen, C. Richard, J.-C.M. Bermudez, P. Honeine, Variants of nonnegative least-mean-square algorithm and convergence analysis, IEEE Trans. Signal Process. 62 (15) (2014) 3990–4005.

[28] J. Chen, J.-C.M. Bermudez, C. Richard, Steady-state performance of non-negative least-mean-square algorithm and its variants, IEEE Signal Process. Lett. 21 (8) (2014) 928–932.

[29] H. Lantéri, M. Roche, O. Cuevas, C. Aime, A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints, Signal Process. 81 (5) (2001) 945–974.

[30] L. Kaufman, Maximum likelihood, least-squares, and penalized least squares for pet, IEEE Trans. Med. Imaging 12 (2) (1993) 200–214.

[31] D.L. Duttweiler, Proportionate normalized least-mean-squares adaptation in echo cancelers, IEEE Trans. Speech Audio Process. 8 (5) (2000) 508–518.

[32] E.J. Cands, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted 1 minimization, J. Fourier Anal. Appl. 14 (5–6) (2008) 877–905, http://dx.doi.org/10.1007/s00041-008-9045-x.

[33] M.H. Costa, J.C.M. Bermudez, N.J. Bershad, Stochastic analysis of the Filtered-X LMS algorithm in systems with nonlinear secondary paths, IEEE Trans. Signal Process. 50 (6) (2002) 1327–1342.

[34] N.R. Yousef, A.H. Sayed, Ability of adaptive filters to track carrier offsets and random channel nonstationarities, IEEE Trans. Signal Process. 50 (7) (2002) 1533–1544.

[35] J. Chen, C. Richard, J.-C. M. Bermudez, P. Honeine, Identification en ligne avec régularisation L1. Algorithme et analyse de convergence en environnement non-stationnaire, in: Colloque Gretsi, Brest, France, 2013, pp. 1–4.

[36] Y. Chen, Y. Gu, A.O. Hero, Sparse LMS for system identification, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, China, 2009, pp. 1325–3128.