Reweighted nonnegative least-mean-square algorithm

Jie Chen(1), *Member, IEEE*, Cédric Richard(1), *Senior Member, IEEE*,

José Carlos M. Bermudez(2), *Senior Member, IEEE*

(1) Université de Nice Sophia-Antipolis, UMR CNRS 6525, Observatoire de la Côte d'Azur

Laboratoire Lagrange, Parc Valrose, 06108 Nice cedex 2 - France

tel.: +33.4.92.07.63.94 fax.: +33.4.92.07.63.21

dr.jie.chen@ieee.org cedric.richard@unice.fr

(2) Federal University of Santa Catarina

Department of Electrical Engineering, 88040-900, Florianópolis, SC - Brazil

tel.: +55.48.3721.7719 fax.: +55.48.3721.9280

j.bermudez@ieee.org

Abstract

Statistical inference subject to nonnegativity constraints is a frequently occurring problem in signal processing. 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. Simulations are conducted to validate the theoretical results. We also introduce a potential application of this algorithm to sparse system identification.

Index Terms

Online system identification, nonnegativity constraints, behavior analysis, sparse system identification.

This work has been partly supported by CNPq grant No. 307071/2013-8 and by CAPES - Proc. 3367-11-8.

I. Introduction

Adaptive filtering is a powerful framework for addressing system identification problems. Algorithms such as the Least-Mean-Square (LMS) and the Recursive Least-Square (RLS) algorithms aim at minimizing 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 [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]. PCA can also be conducted subject to nonnegativity constraints in order to enhance result interpretability [11]. Finally, there are important problems in signal processing that can be cast as optimization problems under nonnegativity constraints [12]. Other applications related to nonnegativity constraints can be found in [5], [13]–[16].

Non-negativity constrained problems can be solved in a batch mode via active set methods [17], [18], gradient projection methods [19], [20], and multiplicative methods [21], 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 [22] 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 [23], 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 [24]. It was observed that a drawback of the NNLMS algorithm is that the filter coefficients suffer from unbalanced convergence rates. In particular, convergence of small coefficients in the active set, that is, those tending to zero at steady-state, progressively slows down with time and almost stalls when approaching the steady-state. To alleviate this drawback, a variant of the NNLMS algorithm was proposed in [23] which applies a Gamma correction term to each component of the gradient in the update relation. This operation is however time consuming, which may preclude its use in real-time applications requiring filters with a large number of coefficients. In addition, this reweighting operation may be insufficient to address the convergence issue in the case of a large spread in coefficient values.

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 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 coefficient range. Transient and steady-state excess mean-square error are also analyzed. Simulations are conducted to illustrate the performance of the proposed algorithm and to validate the theoretical findings. The rest of this paper is organized as follows. Section II reviews the problem of system identification under non-negativity constraints and briefly introduces the NNLMS algorithm. Section III motivates and introduces a new variant of the NNLMS algorithm. In Section IV, the behavior in the mean and mean-square-error sense, and the tracking performance of this algorithm, are studied. Section V provides simulation results to illustrate the properties of the algorithm and the consistency with respect to theoretical results. Section VI concludes the paper.

In this paper normal font letters (x) denote scalars, boldface small letters ($\boldsymbol{x}$) denote vectors, boldface capital letters ($\boldsymbol{X}$) 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, trace$\{\cdot\}$ denotes trace of a matrix, and $E\{\cdot\}$ denotes statistical expectation. Either $\boldsymbol{D}_x$ or $\boldsymbol{D}\{x_1,\dots,x_N\}$ denote a diagonal matrix whose main diagonal is the vector $\boldsymbol{x} = [x_1,\dots,x_N]^\top$. The operator diag$\{\cdot\}$ forms a column vector with the main diagonal entries of its matrix argument.

II. 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 $\boldsymbol{\alpha}^\star = [\alpha_1^\star, \alpha_2^\star, \dots, \alpha_N^\star]^\top$ an unknown parameter vector, and $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^\top$ the vector of regressors with positive definite correlation matrix $\boldsymbol{R}_x > 0$. 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 with variance σ_z^2. We seek to identify this system by minimizing the following constrained mean-square error criterion:

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

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 $L(\boldsymbol{\alpha}, \lambda) = J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}^\top \boldsymbol{\alpha}$, with $\boldsymbol{\lambda}$ the vector of nonnegative Lagrange multipliers. The KKT conditions for (2) at the optimum $\boldsymbol{\alpha}^o$ can be combined into the following expression [22], [25]

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

where ∇_α stands for the gradient operator with respect to $\boldsymbol{\alpha}$. Implementing a fixed-point strategy with (4) leads to the component-wise gradient descent algorithm [22]

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

where $\eta(n)$ is the positive step size that controls the convergence rate and $f_i(\boldsymbol{\alpha}(n)) > 0$. This iteration is similar in some sense to the expectation maximization (EM) algorithm [26]. Using $f_i(\boldsymbol{\alpha}(n)) = 1$, stochastic gradient approximations as for the LMS algorithm, and rewriting the update equation in vectorial form, we obtain the NNLMS algorithm [22]:

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

where $\boldsymbol{D}_\alpha(n) = \text{diag}\{\boldsymbol{\alpha}(n)\}$ and $e(n)$ is the estimation error at time instant n:

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

The algorithm requires to be initialized with positive values. Suppose that $\boldsymbol{\alpha}(n)$ is nonnegative at time n. If the step size satisfies

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

for all $i \in \{j : e(n)\, x_j(n) < 0\}$, the nonnegativity constraint is satisfied at time $n+1$ with (6). If $e(n)\, x_i(n) \geq 0$, there is no restriction on the step size to guarantee the nonnegativity constraint. Further, if we allow instantaneous negative estimates because $\eta(n)$ does not obey the above condition, the algorithm can still converge to the constrained optimum $\boldsymbol{\alpha}^o$ provided that the step size is sufficiently small. Convergence of the NNLMS algorithm was analyzed in [22]. Its steady-state excess mean-square error (EMSE) was studied in [24].

III. Motivating facts and the algorithm

Compared with the LMS algorithm, it can be observed that the NNLMS in (6) can be thought of as a scaled stochastic gradient descent algorithm with the gap between each entry $\alpha_i(n)$ and the nonnegativity bound 0 used as a scaling factor, that is, each component of the gradient vector is scaled by $\alpha_i(n)$. The update vector $\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n)e(n)$ is thus no longer in the direction of the gradient. On the one hand, this 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. On the other hand, if a coefficient $\alpha_i(n)$ that approaches zero turns negative due to the stochastic update, the negative sign of this coefficient turns the update $\alpha_i(n)x_i(n)e(n)$ from the gradient descent strategy to a gradient ascent one, and coefficient $\alpha_i(n)$ moves again towards zero. Moreover, unlike LMS, the correction term $\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n)e(n)$ in the NNLMS algorithm is a nonlinear function of $\boldsymbol{\alpha}(n)$. This causes the two algorithms to have different convergence behaviors.

In the case of NNLMS, the presence of the factor $\alpha_i(n)$ in the update $\alpha_i(n)x_i(n)e(n)$ of the i-th coefficient leads to different convergence rates for coefficients of different values. This is particularly problematic for the coefficients in the active set, as they become progressively smaller through iterations and finally tend to zero. Hence, their convergence 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 update ranges introduces difficulties for step size selection and coefficient initialization, since the estimated coefficients act as different directional contributions to the 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 algorithm should lead to more balanced convergence rates and steady-state weight errors than the original algorithm (6), but without introducing significant computational burden;
- The sensitivity of the algorithm (6) to the spread of the coefficient updates should be reduced.

The Exponential NNLMS [23] replaces the gradient scaling $\alpha_i(n)$ with $\alpha_i^\gamma(n) = \mathrm{sgn}(\alpha_i(n))|\alpha_i(n)|^\gamma$, 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 (4):

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

with ϵ a small positive parameter so that $f_i(n) > 0$ as needed. Defining the diagonal matrix $\boldsymbol{D}_f(n)$ with i-th 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:

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

where

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

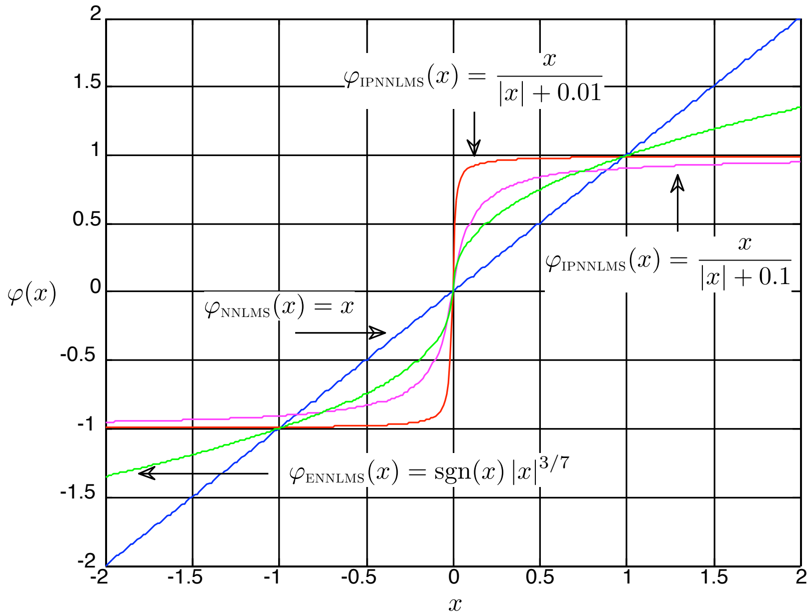


Fig. 1. Gradient scaling function of NNLMS, IP-NNLMS ($\epsilon = 0.1$ and 0.01), and Exponential NNLMS with $\gamma = \frac{3}{7}$ [23]. Note that $w(x)$ is unbounded for NNLMS and Exponential NNLMS.

is the diagonal matrix with i-th 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 [27]. 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 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.

IV. 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 [27]. In this section, we shall provide analytical models for the IP-NNLMS algorithm involving several reasonable approximations. Simulations will validate the proposed models.

A. Mean weight behavior analysis

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,

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

Subtraction $\boldsymbol{\alpha}^\star$ 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

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

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

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

To conduct the analysis, the modified independence assumption (MIA) will be used throughout this section. It assumes that the weight error vector $\boldsymbol{v}(n)$ is statistically independent of $\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)$. This assumption is less restrictive than the conventional independence assumption as discussed in detail in [28].

Taking the expected value on both sides of (16), we note that the last term on its RHS is equal to zero as the noise $z(n)$ is assumed zero-mean and independent of any other signal. It was found in [23] that the first order behavior of NNLMS type algorithms is not highly sensitive to approximations applied to the mean weight recursion relation. Based on the zeroth-order approximation of the gradient scaling factor about $E\{\boldsymbol{\alpha}(n)\}$, we write

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

As a consequence, we have

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

Using the MIA, the mean weight error behavior can finally be described by

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

Monte Carlo simulations reported in Section V, and depicted in Fig. 2, 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 [22] for a related analysis, that of the NNLMS algorithm, based on a fixed-point equation analysis.

B. 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. Using $e(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$, neglecting the statistical dependence of $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$ and $\boldsymbol{v}(n)$, and using the properties for $z(n)$, yields the mean-square estimation error (MSE):

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

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 noise power σ_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 input signals, we derive a model for the behavior of $\zeta(n) = \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}$. 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 approximate the nonlinear reweighting term (12) by its first-order Taylor approximation about $E\{\alpha_i(n)\}$

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

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 the same steps as (64)–(72) in [23] and the assumptions therein, we have

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

with

$$\boldsymbol{P}_1(n) = E\Big\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{x}^\top(n)\Big\} = \boldsymbol{D}_r(n)\,\boldsymbol{R}_x \tag{26}$$

$$\boldsymbol{P}_2(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{r}^\top(n)\,\boldsymbol{D}_x(n)\} \approx \boldsymbol{D}_r(n)\,\boldsymbol{R}_x\,\boldsymbol{D}_r(n) \tag{27}$$

$$\begin{aligned}\boldsymbol{P}_3(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{D}_r(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_v(n)\}\,\boldsymbol{D}_s(n)\end{aligned} \tag{28}$$

$$\begin{aligned}\boldsymbol{P}_4(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{D}_s(n)\big(\boldsymbol{R}_x \circ \boldsymbol{K}(n)\big)\,\boldsymbol{D}_s(n)\end{aligned} \tag{29}$$

$$\begin{aligned}\boldsymbol{P}_5(n) &= \big\{\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\big\}\\ &= \mathrm{diag}\Big\{\boldsymbol{R}_x\,E\{\boldsymbol{v}(n)\}\Big\}\boldsymbol{D}_s(n)\end{aligned} \tag{30}$$

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

$$\begin{aligned}\boldsymbol{P}_7(n) &= E\{\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\}\\ &\approx \boldsymbol{D}_r(n)\,\boldsymbol{Z}(n)\,E\{\boldsymbol{D}_v(n)\}\,\boldsymbol{D}_s(n)\end{aligned} \tag{32}$$

$$\begin{aligned}\boldsymbol{P}_8(n) &= E\{\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\}\\ &\approx \boldsymbol{D}_s(n)\,\Big(\boldsymbol{Z}(n)\circ\boldsymbol{K}(n)\Big)\,\boldsymbol{D}_s(n)\end{aligned} \tag{33}$$

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

$$\boldsymbol{Z}(n) = 2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x. \tag{34}$$

Using (26)–(33) with (25) leads to a recursive model for the transient behavior of $\boldsymbol{K}(n)$. Monte Carlo simulations reported in Section V, and depicted in Fig. 3, illustrate the consistency of this model.

C. Steady-state performance and tracking properties

In order 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 [2], [29]

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

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

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

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,

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

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

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

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:

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

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

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

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

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

With these matrices, we have

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

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

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

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

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

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 and the limit, we have

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

which yields

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

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

$$\begin{aligned} &\lim_{n\to\infty} E\big\{\boldsymbol{v}'^\top(n)\overline{\boldsymbol{I}}\,\boldsymbol{x}(n)\,e(n)\big\} \\ &\quad= -\lim_{n\to\infty} E\big\{\boldsymbol{v}'^\top(n)\,\overline{\boldsymbol{I}}\,\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}'(n) + \boldsymbol{v}'^\top(n)\,\overline{\boldsymbol{I}}\,\boldsymbol{x}(n)\boldsymbol{x}^\top(n)E\{\boldsymbol{v}(\infty)\}\big\} \\ &\quad= -\zeta'(\infty) + \lim_{n\to\infty} E\{\boldsymbol{v}'^\top(n)\,(\boldsymbol{I} - \overline{\boldsymbol{I}})\,\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}'(n)\} \\ &\quad= -\zeta'(\infty) - \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Gamma}\,(\boldsymbol{I} - \overline{\boldsymbol{I}})\} \end{aligned} \tag{54}$$

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 $\boldsymbol{x}^\top(n)\,\overline{\boldsymbol{I}}\,\boldsymbol{D}_w(n)\,\boldsymbol{x}(n)$ for $n \to \infty$. This leads to

$$\lim_{n\to\infty} E\{\boldsymbol{x}^\top(n)\overline{\boldsymbol{I}}\boldsymbol{D}_w(n)\boldsymbol{x}(n)e^2(n)\} \approx \text{trace}\{E\{\boldsymbol{D}_w(\infty)\}\boldsymbol{R}_x\}\,(\sigma_z^2 + \zeta'(\infty) + \zeta^\infty). \tag{55}$$

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

$$\lim_{n\to\infty} E\big\{\|\boldsymbol{\theta}(n)\|^2_{\overline{\boldsymbol{D}}_w^{-1}(n)}\big\} = \text{trace}\{E\{\overline{\boldsymbol{D}}_w^{-1}(\infty)\}\,\boldsymbol{R}_\theta\} \tag{56}$$

$$\lim_{n\to\infty} E\big\{\|\boldsymbol{q}(n+1)\|^2_{\overline{\boldsymbol{D}}_w^{-1}(n)}\big\} = \text{trace}\{E\{\overline{\boldsymbol{D}}_w^{-1}(\infty)\}\,\boldsymbol{R}_q\} \tag{57}$$

The sixth term on the RHS of (50) can be written as

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

and, finally, the last term is given by

$$\begin{aligned}\lim_{n\to\infty} E\{e(n)\boldsymbol{x}^\top(n)\,\overline{\boldsymbol{I}}\boldsymbol{\theta}(n)\} &= \lim_{n\to\infty} -E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\overline{\boldsymbol{I}}\boldsymbol{\theta}(n)\}\\ &= -\text{trace}\{\overline{\boldsymbol{I}}\boldsymbol{\Gamma}\boldsymbol{R}_x\}\end{aligned} \tag{59}$$

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

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

where

$$\beta = \text{trace}\{\boldsymbol{R}_x\boldsymbol{\Gamma}(\overline{\boldsymbol{I}}-\boldsymbol{I})\} \tag{61}$$

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

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

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

To obtain the steady-state performance in a stationary environment, it is sufficient to set ρ and $\boldsymbol{R}_q$ to zero in (63).

V. Simulation results

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

A. 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. 2. 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 [22], [23]. 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 coefficients in the active set did 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. 3 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 [22], [23]. 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.

B. 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. 4. As expected, it can be observed that the steady-state estimation error increases with the variance σ_q^2.

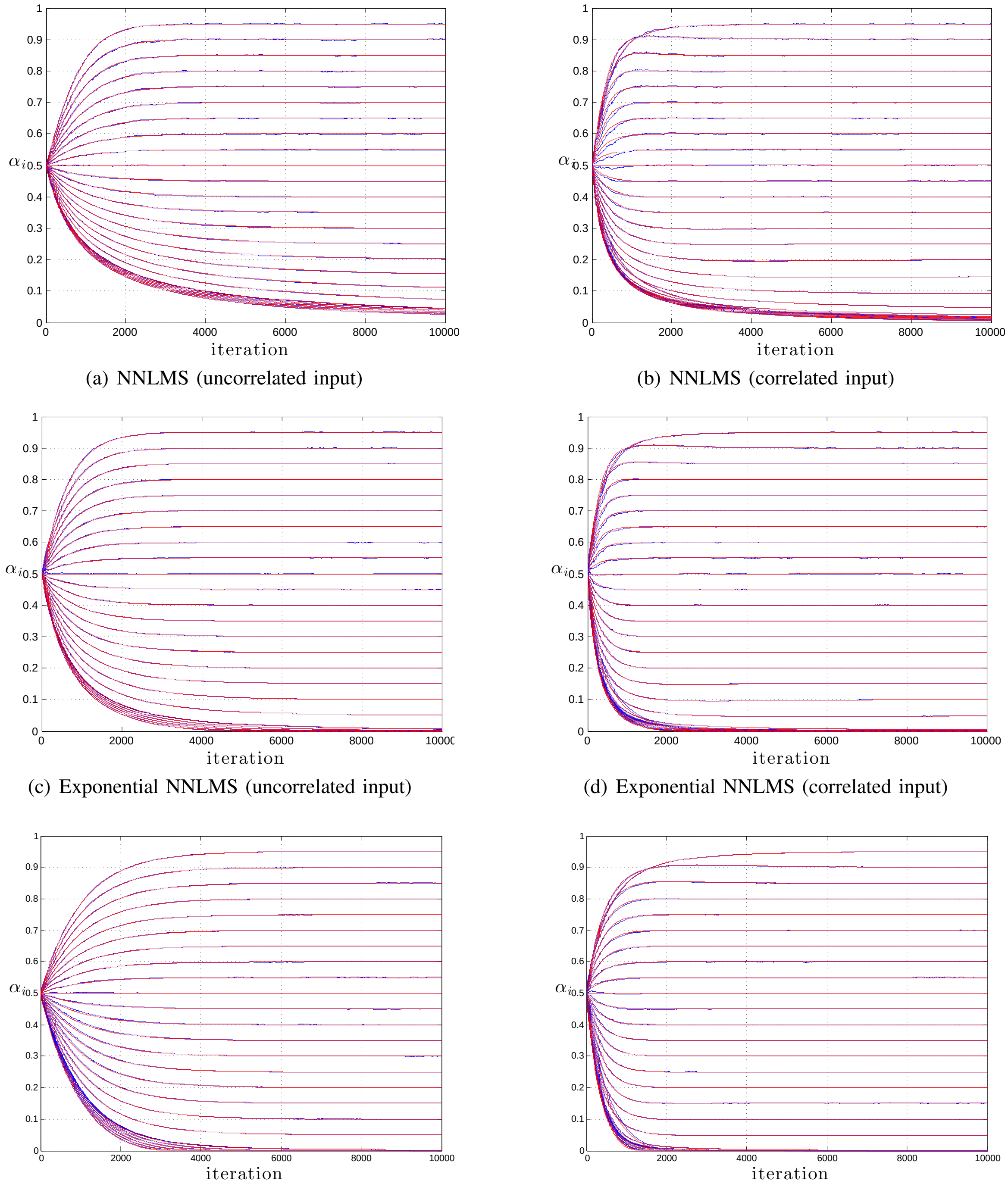
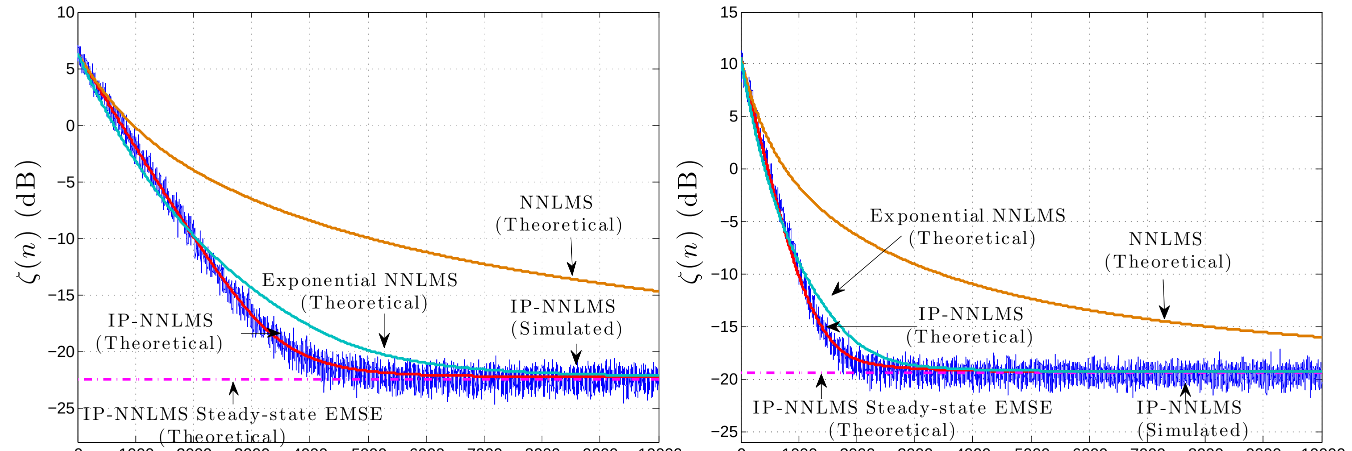


Fig. 2. Mean weight behavior of NNLMS, IP-NNLMS and Exponential NNLMS algorithms. Left column: uncorrelated input. Right column: correlated input with $\tau = 0.5$. Red curves were obtained with theoretical models, and blue curves were obtained by averaging over 100 Monte Carlo simulations.

C. Application to sparse online system identification

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

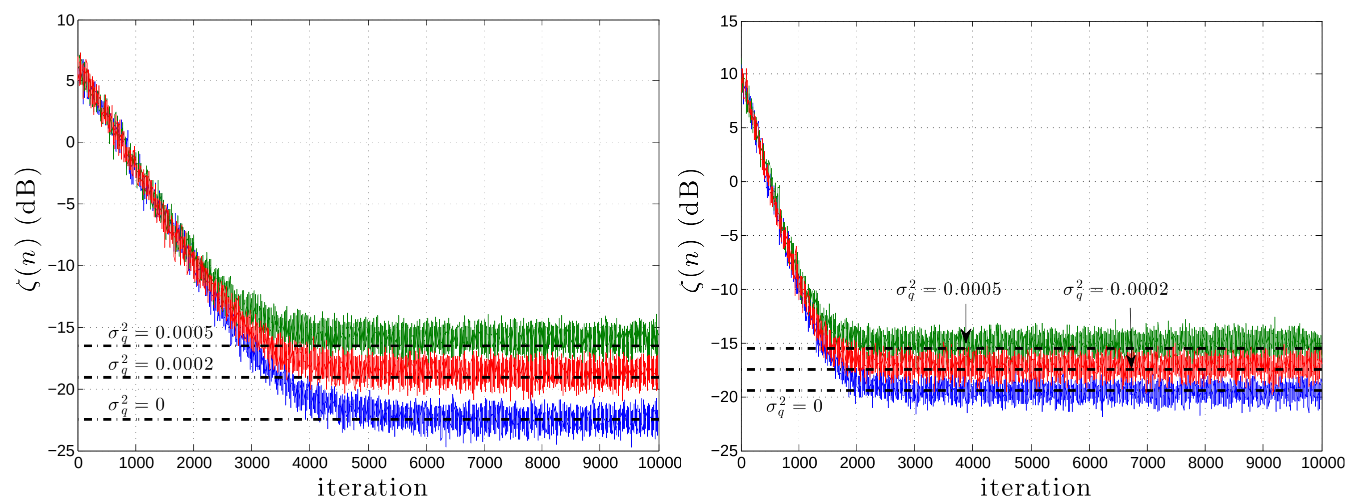
$$\boldsymbol{\alpha}^o = \arg\min_{\alpha} \frac{1}{2} E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\} + \nu \, \|\boldsymbol{\alpha}\|_1 \tag{66}$$



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

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

Fig. 3. Learning curves of NNLMS, IP-NNLMS and Exponential NNLMS algorithms. Left column: uncorrelated input. Right column: correlated input with $\tau = 0.5$.



(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 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.

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 $\tilde{\boldsymbol{\alpha}} = \text{col}\{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-\}$ and $\tilde{\boldsymbol{x}}(n) = \text{col}\{\boldsymbol{x}(n), -\boldsymbol{x}(n)\}$ where the operator $\text{col}\{\cdot\}$ 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, problem (68) can be solved in an online manner as follows [30]

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

with $\boldsymbol{D}_f(n)$ the diagonal matrix with i-th 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 i-th entry of $\boldsymbol{\gamma}(n)$ is given by $\gamma_i(n) = \frac{1}{\tilde{\alpha}_i(n)+\mu}$, with μ a small positive number.

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

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

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 [31] with reweighted ℓ_1-norm regularizer defined as

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

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(c) 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. 6(a)-6(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.

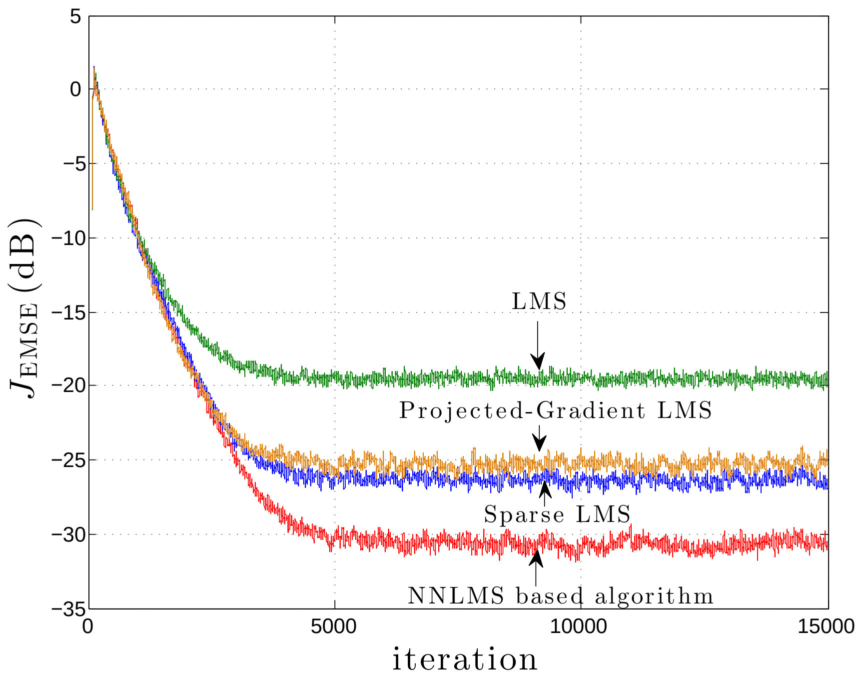


Fig. 5. EMSE learning curves.

VI. Conclusion

In this paper, we presented an algorithm for online system identification subject to nonnegativity constraint. This algorithm is a variant of the NNLMS algorithm that is able to balance the convergence rate of the filter coefficients more efficiently by reweighting the gradient entries. We analyzed this algorithm in the mean and mean-square-error sense, and considered the case of time-varying systems in order to appreciate its tracking ability. Simulations were conducted to illustrate the performance of the proposed algorithm and to validate the theoretical findings. Finally, an application to sparse identification was considered.

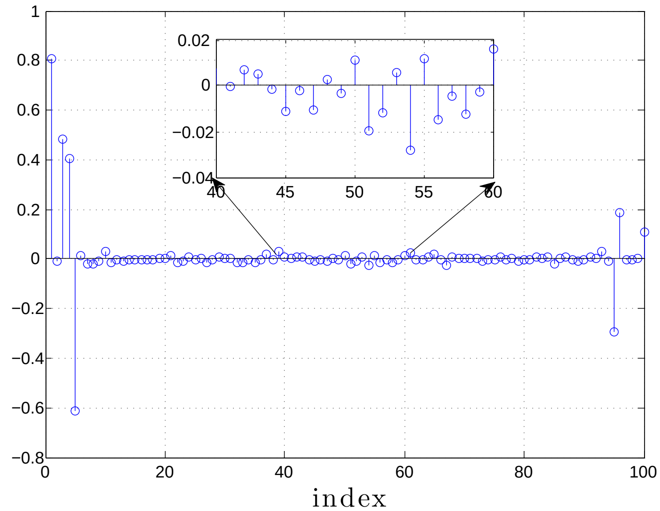
References

[1] S. Haykin, *Adaptive filter theory*, Pearson Education India, 4th edition, 2005.

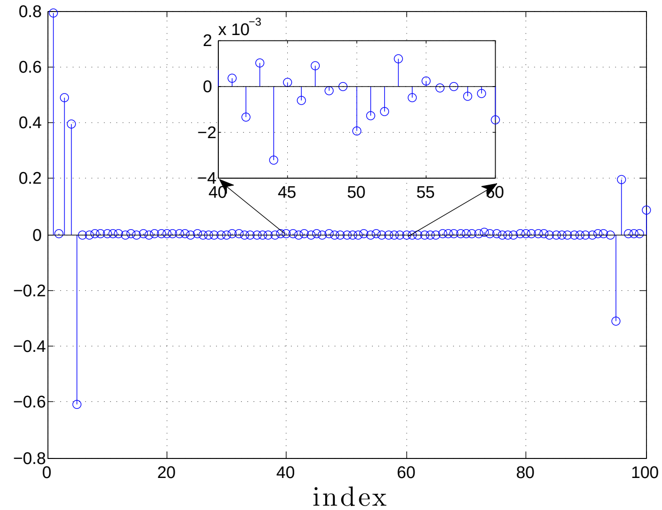
[2] A. H. Sayed, *Adaptive filters*, John Wiley & Sons, NJ, 2008.

[3] T. A. Johansen, "Constrained and regularized system identification," *Modeling Identification and Control*, vol. 79, no. 2, pp. 109–116, 1998.

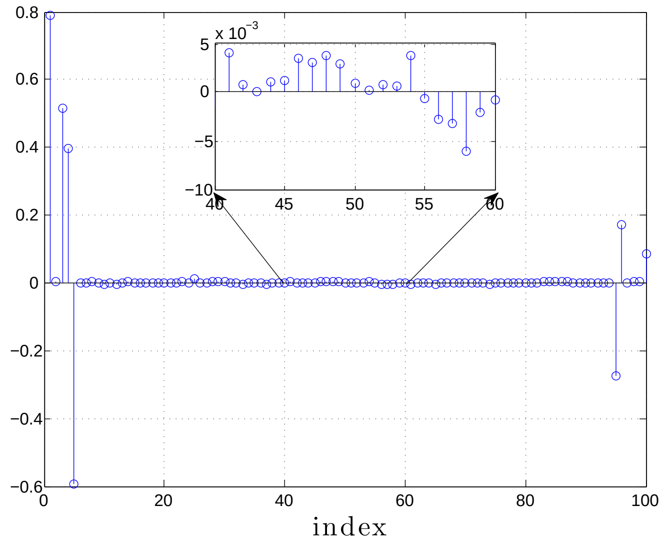
[4] A. Yeredor, "On the role of constraints in system identification," in *Proc. of International Workshop on Total Least Squares and Errors-in-Variables Modeling*, Leuven, Belgium, August 2006, pp. 46–48.



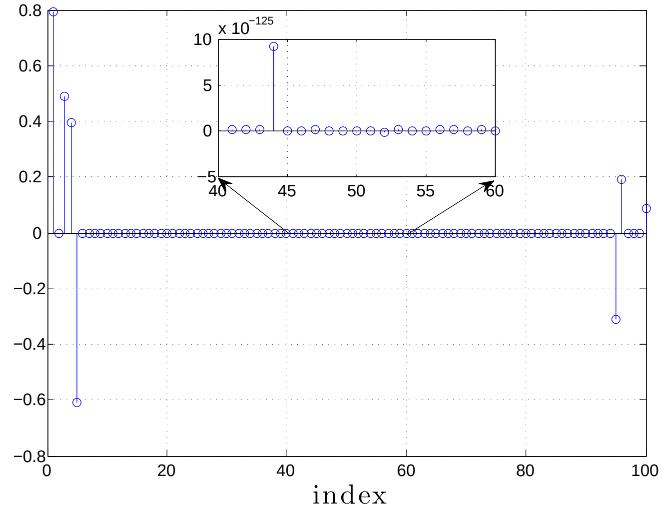
(a) Coefficients of LMS



(b) Coefficients of sparse LMS algorithm (72)



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



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

Fig. 6. Filter coefficients corresponding to a single realization.

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

[6] R. C. Puetter, T. R. Gosnell, and A. Yahil, "Digital image reconstruction: Deblurring and denoising," *Annual Review of Astronomy and Astrophysics*, vol. 43, pp. 139–194, 2005.

[7] J. M. Bardsley, J. K. Merikoski, and R. Vio, "The stabilizing properties of nonnegativity constraints in least squares image reconstruction," *International Journal of Pure and Applied Mathematics*, vol. 43, no. 1, pp. 95–109, 2008.

[8] K. B. Nakshatralaa, M. K. Mudunurua, and A. J. Valocchib, "A numerical framework for diffusion-controlled bimolecular-reactive systems to enforce maximum principles and the non-negative constraint," *Journal of Computational Physics*, vol. 253, no. 15, pp. 278–307, November 2013.

[9] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, January 2002.

[10] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, October 1999.

[11] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *Advances in Neural Information Processing Sysems*, B. Schölkopf, J. Platt,

and T. Hoffman, Eds., vol. 19, pp. 1561–1568. MIT Press, Cambridge, MA, 2007.

[12] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, December 2007.

[13] L. K. Saul, F. Sha, and D. D. Lee, "Statistical signal processing with nonnegative constraints," in *Proc. of European Conference on Speech Communication and Technology*, Geneva, Switerland, September 2003, pp. 1001–1004.

[14] Y. Lin and D. D. Lee, "Bayesian regularization and nonnegative deconvolution for room impulse response estimation," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 839–847, March 2006.

[15] R. Zdunek and M. J. Nawrocki, "Improved modeling of highly loaded UMTS network with nonnegativity constraints," in *Proc. of Annual IEEE International Sysmbosium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Helsinki, Finland, September 2006, pp. 1–5.

[16] M. A. Khajehnejad, A. G. Dimakis, W. Xu, and B. Hassibi, "Sparse recovery of nonnegative signals with minimal expansion," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 196–208, January 2011.

[17] C. L. Lawson and R. J. Hanson, *Solving Least Square Problems*, Society for Industrial and applied mathematics, Philadelphia, PA, 1995.

[18] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, September/October 1997.

[19] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mahematical Programming*, vol. 39, no. 1, pp. 93–116, September 1987.

[20] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections: A unifying framework for linear and nonlinear classification and regression tasks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, January 2011.

[21] C. J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, November 2007.

[22] J. Chen, C. Richard, J. C. M. Bermudez, and P. Honeine, "Nonnegative least-mean square algorithm," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5225–5235, November 2011.

[23] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, "Variants of nonnegative least-mean-square algorithm and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3990–4005, August 2014.

[24] J. Chen, J.-C. M. Bermudez, and C. Richard, "Steady-state performance of non-negative least-mean-square algorithm and its variants," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 928–932, August 2014.

[25] H. Lantéri, M. Roche, O. Cuevas, and C. Aime, "A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints," *Signal Processing*, vol. 81, no. 5, pp. 945–974, May 2001.

[26] L. Kaufman, "Maximum likelihood, least-squares, and penalized least squares for pet," *IEEE Transaction on Medical Imaging*, vol. 12, no. 2, pp. 200–214, June 1993.

[27] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, September 2000.

[28] J. Minkoff, "On the unnecessary assumption of statistical independence between reference signal and filter weights in feedforward adaptive systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 5, pp. 1109, May 2001.

[29] N. R. Yousef and A. H. Sayed, "Ability of adaptive filters to track carrier offsets and random channel nonstationarities," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1533–1544, July 2002.

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

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