Nonnegative Least-Mean-Square Algorithm

Jie Chen, Cédric Richard, *Senior Member, IEEE*, José Carlos M. Bermudez, *Senior Member, IEEE*, and Paul Honeine, *Member, IEEE*

***Abstract*—Dynamic system modeling plays a crucial role in the development of techniques for stationary and nonstationary signal processing. Due to the inherent physical characteristics of systems under investigation, nonnegativity is a desired constraint that can usually be imposed on the parameters to estimate. In this paper, we propose a general method for system identification under nonnegativity constraints. We derive the so-called *nonnegative least-mean-square algorithm* (NNLMS) based on stochastic gradient descent, and we analyze its convergence. Experiments are conducted to illustrate the performance of this approach and consistency with the analysis.**

***Index Terms*— Adaptive filters, adaptive signal processing, least mean square algorithms, nonnegative constraints, transient analysis.**

I. Introduction

In many real-life phenomena including biological and physiological ones, due to the inherent physical characteristics of systems under investigation, nonnegativity is a desired constraint that can be imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. For instance, in the study of a concentration field or a thermal radiation field, any observation is described with nonnegative values (ppm, joule). Nonnegativity as a physical constraint has received growing attention from the signal processing community during the last decade. For instance, consider the following nonnegative least-square inverse problem:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2 \quad \text{subject to } [\boldsymbol{x}]_i \geq 0, \quad \forall i \tag{1}$$

with $\boldsymbol{A}$ a real $M \times N$ matrix of rank $k \leq \min(M,N)$, $\boldsymbol{b}$ an M-length real vector, and $\boldsymbol{x}$ an N-length real vector. $\|\cdot\|$ denotes the Euclidean 2-norm and $[\cdot]_i$ the ith entry of the vector. This problem has been addressed in various contexts, with applications ranging from image deblurring in astrophysics [1] to deconvolution of emission spectra in chemometrics [2]. Another similar problem is the nonnegative matrix factorization (NMF), which is now a popular dimension reduction technique [3]–[5]. Given a matrix $\boldsymbol{X}$ with nonnegative entries, the squared error version of this problem can be stated as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2 \quad \text{subject to } [\boldsymbol{W}]_{ij} \geq 0, \quad [\boldsymbol{H}]_{ij} \geq 0, \quad \forall i,j \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This problem is closely related to the blind deconvolution one, and has found direct application in hyperspectral imaging [6]. Separation of nonnegative mixture of nonnegative sources has also been considered in [7], [8].

Over the last 15 years, a variety of methods have been developed to tackle nonnegative least-square problems (NNLS). Active set techniques for NNLS use the fact that if the set of variables which activate constraints is known, then the solution of the constrained least-square problem can be obtained by solving an unconstrained one that only includes inactive variables. The active set algorithm of Lawson and Hanson [9] is a batch resolution technique for NNLS problems. It has become a standard among the most frequently used methods. In [10], Bro and De Jong introduced a modification of the latter, called fast NNLS, which takes advantage of the special characteristics of iterative algorithms involving repeated use of nonnegativity constraints. Another class of tools is the class of projected gradient algorithms [11]–[14]. They are based on successive projections on the feasible region. In [15], Lin used this kind of algorithms for NMF problems. Low memory requirements and simplicity make algorithms in this class attractive for large scale problems. Nevertheless, they are characterized by slow convergence rate if not combined with appropriate step size selection. The class of multiplicative algorithms is very popular for dealing with NMF problems [4], [16]. Particularly efficient updates were derived in this way for a large number of problems involving nonnegativity constraints [17]. These algorithms however require batch processing, which is not suitable for online system identification problems.

In this paper, we consider the problem of system identification under nonnegativity constraints on the parameters to estimate. The Karush-Kuhn-Tucker (KKT) conditions are established for any convex cost function, and a fixed-point iteration strategy is then applied in order to derive a gradient descent algorithm. Considering the square-error criterion as a particular case, a stochastic gradient scheme is presented. A convergence analysis of this algorithm is proposed. The resulting model accurately predicts the algorithm behavior for both transient and steady-state conditions. Finally, experiments are conducted to evaluate the algorithm performance and its consistency with the analysis.

Manuscript received November 03, 2010; revised May 09, 2011; accepted June 27, 2011. Date of publication July 22, 2011; date of current version October 12, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Isao Yamada. This work was supported in part by CNPq Grant No. 305377/2009-4.

J. Chen and P. Honeine are with the Institut Charles Delaunay, CNRS, University of Technology of Troyes, 10010 Troyes cedex, France (e-mail: chenjieg@sina.com; paul.honeine@utt.fr).

C. Richard is with the Côte d'Azur Observatory, CNRS, University of Nice Sophia-Antipolis, Parc Valrose, 06108 Nice cedex 2, France (e-mail: cedric.richard@unice.fr).

J.-C. M. Bermudez is with the Department of Electrical Engineering, Federal University of Santa Catarina 88040-900, Florianópolis, SC, Brazil (e-mail: j.bermudez@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2162508

II. System Identification With Non-Negativity Constraints

Consider an unknown system, only characterized by a set of real-valued discrete-time responses to known stationary inputs. The problem is to derive a transversal filter model

$$y(n) = \boldsymbol{\alpha}^\top \boldsymbol{x}(n) + z_1(n) \tag{3}$$

with $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^\top$ the vector of the model parameters, and $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^\top$ the observed data vector. The input signal $x(n)$ and the desired output signal $y(n)$ are assumed zero-mean stationary. The sequence $z_1(n)$ accounts for measurement noise and modeling errors.

Due to the inherent physical characteristics of systems under investigation, in this paper, nonnegativity is a desired constraint that is imposed on the coefficient vector $\boldsymbol{\alpha}$. Therefore, the problem of identifying the optimum model can be formalized as follows:

$$\begin{aligned} \boldsymbol{\alpha}^o &= \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \\ &\quad \text{subject to } \alpha_i \geq 0, \quad \forall i \end{aligned} \tag{4}$$

with $J(\boldsymbol{\alpha})$ a continuously differentiable and strictly convex cost function in $\mathbb{R}^N$, and $\boldsymbol{\alpha}^o$ the optimal solution to the constrained optimization problem.

A. A Fixed-Point Iteration Scheme

In order to solve the problem (4), let us consider its Lagrangian function $Q(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ given by [18]

$$Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}^\top \boldsymbol{\alpha}$$

where $\boldsymbol{\lambda}$ is the vector of nonnegative Lagrange multipliers. The Karush-Kuhn-Tucker conditions must necessarily be satisfied at the optimum defined by $\boldsymbol{\alpha}^o$, $\boldsymbol{\lambda}^o$, namely

$$\begin{aligned} \nabla_\alpha Q(\boldsymbol{\alpha}^o, \boldsymbol{\lambda}^o) &= 0 \\ \alpha_i^o [\boldsymbol{\lambda}^o]_i &= 0, \quad \forall i \end{aligned}$$

where the symbol ∇_α stands for the gradient operator with respect to $\boldsymbol{\alpha}$. Using $\nabla_\alpha Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \nabla_\alpha J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}$, these equations can be combined into the following expression:

$$\alpha_i^o \left[-\nabla_\alpha J(\boldsymbol{\alpha}^o)\right]_i = 0 \tag{5}$$

where the extra minus sign is just used to make a gradient descent of $J(\boldsymbol{\alpha})$ apparent. To solve (5) iteratively, two important points have to be noticed. The first point is that $\boldsymbol{D}(-\nabla_\alpha J(\boldsymbol{\alpha}))$ is also a gradient descent of $J(\boldsymbol{\alpha})$ if $\boldsymbol{D}$ is a symmetric positive definite matrix. The second point is that equations of the form $\varphi(u) = 0$ can be solved with a fixed-point iteration algorithm, under some conditions on function φ, by considering the problem $u = u + \varphi(u)$. Implementing this strategy with (5) leads us to the component-wise gradient descent algorithm

$$\alpha_i(n+1) = \alpha_i(n) + \eta_i(n) f_i\left(\boldsymbol{\alpha}(n)\right) \alpha_i(n) \left[-\nabla_\alpha J\left(\boldsymbol{\alpha}(n)\right)\right]_i \tag{6}$$

with $\eta_i(n)$ a positive step size required to get a contraction scheme and to control the convergence rate. Function $f_i(\boldsymbol{\alpha}) > 0$ in (6) is the ith entry of a diagonal matrix $\boldsymbol{D}$. It is an arbitrary positive function of $\boldsymbol{\alpha}$. Some criteria $J(\boldsymbol{\alpha})$ are defined only for inputs $\boldsymbol{\alpha}$ with positive entries, e.g., Itakura-Saito distance, Kullback-Leibler divergence. If necessary, this condition can be managed by an appropriate choice of the step size parameter. Assume that $\alpha_i(n) \geq 0$. Nonnegativity of $\alpha_i(n+1)$ is guaranteed if

$$1 + \eta_i(n) f_i\left(\boldsymbol{\alpha}(n)\right) \left[-\nabla_\alpha J(\boldsymbol{\alpha}(n)\right]_i \geq 0. \tag{7}$$

If $\left[\nabla_\alpha J(\boldsymbol{\alpha}(n))\right]_i \leq 0$, condition (7) is clearly satisfied and nonnegativity does not impose any restriction on the step size. Conversely, if $\left[\nabla_\alpha J(\boldsymbol{\alpha}(n))\right]_i > 0$, nonnegativity of $\alpha_i(n+1)$ holds if

$$0 \leq \eta_i(n) \leq \frac{1}{f_i\left(\boldsymbol{\alpha}(n)\right) \left[\nabla_\alpha J\left(\boldsymbol{\alpha}(n)\right)\right]_i}. \tag{8}$$

Using a single step size $\eta(n)$ in $[0, \eta_{\max}(n)]$ for all entries of $\boldsymbol{\alpha}$ so that

$$\eta_{\max}(n) = \min_i \frac{1}{f_i\left(\boldsymbol{\alpha}(n)\right) \left[\nabla_\alpha J\left(\boldsymbol{\alpha}(n)\right)\right]_i}, \quad i = 1, \ldots, N \tag{9}$$

the update equation can be written in vector form as

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n) \boldsymbol{d}(n) \tag{10}$$

where the weight adjustment direction $\boldsymbol{d}(n)$, whose ith entry is defined as follows

$$\left[\boldsymbol{d}(n)\right]_i = f_i\left(\boldsymbol{\alpha}(n)\right) \alpha_i(n) \left[-\nabla_\alpha J\left(\boldsymbol{\alpha}(n)\right)\right]_i \tag{11}$$

is a gradient descent direction because $f_i[\boldsymbol{\alpha}(n)]\alpha_i(n) \geq 0$. It should be noted that condition (9) on the step size $\eta(n)$ guarantees the nonnegativity of $\boldsymbol{\alpha}(n)$ for all n, but does not ensure the stability of the algorithm.

B. The Nonnegative Least-Mean-Square (NNLMS) Algorithm

Let us now consider the mean-square error criterion $J_{mse}(\boldsymbol{\alpha})$ to be minimized with respect to $\boldsymbol{\alpha}$, that is,

$$\begin{aligned} \boldsymbol{\alpha}^o &= \arg\min_{\boldsymbol{\alpha}} E\left\{\left[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)\right]^2\right\} \\ &\quad \text{subject to } \alpha_i^o \geq 0, \quad \forall i \end{aligned} \tag{12}$$

where we have included the nonnegativity constraint only on the optimum solution because $J_{mse}(\boldsymbol{\alpha})$ is defined for all $\boldsymbol{\alpha}$, that is, for all positive and negative entries α_i. The gradient of $J_{mse}(\boldsymbol{\alpha})$ can be easily computed as

$$\nabla_\alpha J(\boldsymbol{\alpha}) = 2(\boldsymbol{R}_x \boldsymbol{\alpha} - \boldsymbol{r}_{xy}) \tag{13}$$

with $\boldsymbol{R}_x$ the autocorrelation matrix of $\boldsymbol{x}(n)$ and $\boldsymbol{r}_{xy}$ the correlation vector between $\boldsymbol{x}(n)$ and $y(n)$. Using (10) and (11) with $f_i(\boldsymbol{\alpha}) = \frac{1}{2}$ for all i, the update rule for minimizing the mean-square error under nonnegativity constraints is given by

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n) \boldsymbol{D}_\alpha(n) \left(\boldsymbol{r}_{xy} - \boldsymbol{R}_x \boldsymbol{\alpha}(n)\right) \tag{14}$$

where $\boldsymbol{D}_\alpha(n)$ is the diagonal matrix with diagonal entries given by $\boldsymbol{\alpha}(n)$. Following a stochastic gradient approach, the second-

order moments $\boldsymbol{R}_x$ and $\boldsymbol{r}_{xy}$ are replaced in (14) by the instantaneous estimates $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$ and $y(n)\boldsymbol{x}(n)$, respectively. This leads us to the stochastic approximation of (14) given by[1]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n)e(n)\boldsymbol{D}_x(n)\boldsymbol{\alpha}(n), \quad \eta(n) > 0 \quad (15)$$

where $\boldsymbol{D}_x(n)$ stands for the diagonal matrix with diagonal entries given by $\boldsymbol{x}(n)$, and $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n)$.

It is interesting to notice how the term $\boldsymbol{\alpha}(n)$ in the update term on the right-hand side (RHS) of (15) affects the dynamics of the coefficient update when compared with the well-known LMS algorithm [19]. Note that the extra multiplying factor $\alpha_i(n)$ in the update term of the ith row of (15), which is not present in the LMS update, provides extra control of both the magnitude and the direction of the weight update, as compared to LMS. For a fixed step size η, the update term for the ith component of $\boldsymbol{\alpha}(n)$ is proportional to $-\alpha_i(n)e(n)x_i(n)$, the stochastic gradient component. Thus, compared to the LMS stochastic gradient component $-e(n)x_i(n)$, the constrained algorithm includes the multiplying factor $\alpha_i(n)$. A negative $\alpha_i(n)$ will then change the sign of the LMS adjustment, which on average tends to avoid convergence to negative coefficients of the unconstrained solution. Thus, coefficients that would normally converge, on average, to negative values using unconstrained LMS will tend to converge to zero using the constrained algorithm. In addition, $\alpha_i(n)$ close to zero will tend to slow its own convergence unless the magnitude of $e(n)x_i(n)$ is very large. Finally, $\alpha_i(n) = 0$ is clearly a stationary point of (15).

In the following, the adaptive weight behavior of the adaptive algorithm (15), called *nonnegative LMS*, is studied in the mean and mean-square senses for a time-invariant step size η.

III. Mean Behavior Analysis

We shall now propose a model to characterize the mean behavior of the nonnegative LMS algorithm. Fig. 1 shows a block diagram of the problem studied in this paper. The input signal $x(n)$ and the desired output signal $y(n)$ are assumed stationary and zero-mean. Let us denote by $\boldsymbol{\alpha}^*$ the solution of the unconstrained least-mean-square problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} E\left\{\left[y(n) - \boldsymbol{\alpha}^\top\boldsymbol{x}(n)\right]^2\right\} \quad (16)$$

whose solution $\boldsymbol{\alpha}^*$ satisfies the Wiener-Hopf equations

$$\boldsymbol{R}_x\boldsymbol{\alpha}^* = \boldsymbol{r}_{xy}. \quad (17)$$

The residual signal $z(n) = y(n) - \boldsymbol{x}^\top(n)\boldsymbol{\alpha}^*$ in Fig. 1 accounts for measurement noise and modeling errors. It is assumed in the following that $z(n)$ is stationary, zero-mean with variance σ_z^2 and statistically independent of any other signal. Thus, $E\{z(n)\boldsymbol{D}_x(n\} = 0$.

The adaptive algorithm (15) attempts to estimate the optimum $\boldsymbol{\alpha}^o$ for the constrained problem (12). The analytical determination of the optimal solution $\boldsymbol{\alpha}^o$ is not trivial in general. In the particular case of independent and identically distributed (i.i.d.) input samples, however, $\boldsymbol{R}_x = \sigma_x^2\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix. In this case, the Karush-Kuhn-Tucker conditions imply that $\boldsymbol{\alpha}^o$ is obtained by turning the negative entries of $\boldsymbol{\alpha}^*$ to zero

$$\boldsymbol{\alpha}^o = \{\boldsymbol{\alpha}^*\}_+ \quad (18)$$

where $\{u\}_+ = \max\{0, u\}$. The minimum mean-square error produced by solution $\boldsymbol{\alpha}^o$ is then

$$J_{ms_{min}} = \sigma_y^2 - 2\boldsymbol{r}_{xy}\{\boldsymbol{\alpha}^*\}_+ + \sigma_x^2\{\boldsymbol{\alpha}^*\}_+^\top\{\boldsymbol{\alpha}^*\}_+ \quad (19)$$

with σ_y^2 the variance of $y(n)$.

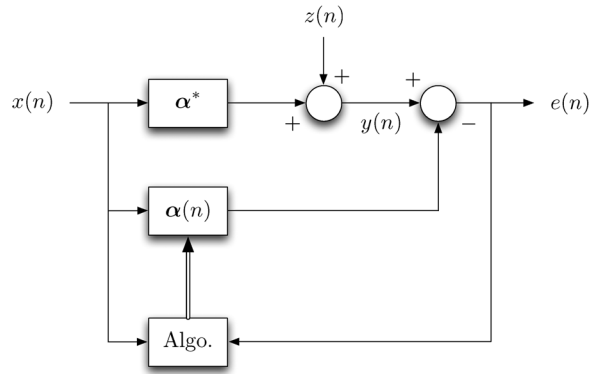

Fig. 1. Adaptive system under study.

A. Mean Weight Behavior Model

Defining the weight-error vector as follows:

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^* = [v_1(n), v_2(n), \ldots, v_N(n)]^\top$$

the update (15) can be written as

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta e(n)\boldsymbol{D}_x(n)\left(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*\right). \quad (20)$$

Using $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n) = z(n) - \boldsymbol{v}^\top(n)\boldsymbol{x}(n)$ leads us to the following expression:

$$\begin{aligned}\boldsymbol{v}(n+1) = \boldsymbol{v}(n) &+ \eta z(n)\boldsymbol{D}_x(n)\boldsymbol{v}(n) + \eta z(n)\boldsymbol{D}_x(n)\boldsymbol{\alpha}^* \\ &-\eta\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n) - \eta\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*x^\top(n)\boldsymbol{v}(n). \quad (21)\end{aligned}$$

Taking the expectation of (21), neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$,[2] and using that $E\{z(n)\boldsymbol{D}_x(n)\} = 0$ yields

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} \approx \left(\boldsymbol{I} - \eta E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\right\}\right)E\{\boldsymbol{v}(n)\} \\ -\eta E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right\}. \quad (22)\end{aligned}$$

The first expectation on the RHS of (22) is given by

$$E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\right\} = E\left\{\boldsymbol{D}_{\alpha^*}\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\} = \boldsymbol{D}_{\alpha^*}\boldsymbol{R}_x. \quad (23)$$

[1]Note that $\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n) = \boldsymbol{D}_x(n)\boldsymbol{\alpha}(n)$.

[2]This assumption is less restrictive than the well-known independence assumption [19, p. 247], as it does not require $x(n)$ be Gaussian.

In order to evaluate the second expectation on the RHS of (22), let us compute the ith component of the vector $\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)$. We have

$$\left[\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right]_i = \sum_{j=1}^{N} x(n-i+1)v_i(n)v_j(n)x(n-j+1). \quad (24)$$

Taking the expectation, defining $\boldsymbol{K}(n) = E\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\}$, and neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$, we obtain

$$\left[E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right\}\right]_i \approx \sum_{j=1}^{N} r_x(j-i)\left[\boldsymbol{K}(n)\right]_{ij} = \left[\boldsymbol{R}_x\boldsymbol{K}(n)\right]_{ii}. \quad (25)$$

This implies that

$$E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right\} \approx \operatorname{diag}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}$$

where $\operatorname{diag}\{\boldsymbol{A}\}$ denotes the vector whose ith entry is defined by $[\boldsymbol{A}]_{ii}$. Using these results with (22) yields the following mean weight-error vector update equation:

$$E\left\{\boldsymbol{v}(n+1)\right\} = (\boldsymbol{I} - \eta\boldsymbol{D}_{\alpha^*}\boldsymbol{R}_x)E\left\{\boldsymbol{v}(n)\right\} - \eta\operatorname{diag}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}. \quad (26)$$

This equation requires second-order moments defined by the matrix $\boldsymbol{K}(n)$ in order to update the first-order one provided by $E\{\boldsymbol{v}(n)\}$. A recursive model will be derived for $\boldsymbol{K}(n)$ in Section IV, see (39). That model can be used along with (26) to predict the mean weight behavior of the algorithm. Nevertheless, we have found that a sufficiently accurate and more intuitive mean behavior model can be obtained using the following separation approximation

$$\boldsymbol{K}(n) \approx E\left\{\boldsymbol{v}(n)\right\}E\left\{\boldsymbol{v}^\top(n)\right\}. \quad (27)$$

Using (27) in (26) we obtain the following result

$$E\left\{\boldsymbol{v}(n+1)\right\} = (\boldsymbol{I} - \eta\boldsymbol{D}_{\alpha^*}\boldsymbol{R}_x)E\left\{\boldsymbol{v}(n)\right\} - \eta\operatorname{diag}\left\{\boldsymbol{R}_x E\left\{\boldsymbol{v}(n)\right\}E\left\{\boldsymbol{v}^\top(n)\right\}\right\}. \quad (28)$$

Approximation (27) assumes that

$$\operatorname{Cov}\left\{v_i(n), v_j(n)\right\} \ll E\left\{v_i(n)\right\}E\left\{v_j(n)\right\}. \quad (29)$$

In general, (29) is valid when the adaptive weights are far from convergence, as the mean weight-error component tends to be much larger than the weight-error fluctuation about the mean. For correlated $x(n)$, the level of the weight-error fluctuations at convergence tends to be much less than the values of the nonzero optimal weights. For white input signals $E\{v_i(n)\}$ tends to zero for those indexes corresponding to the positive coefficients of $\boldsymbol{\alpha}^o$. In this case, approximation (29) will in fact tend to eliminate the weight estimation error at convergence. Extensive simulation results have shown that the simplified model in (28) yields a prediction of the mean weight behavior which is sufficient for design purposes. Furthermore, this simplification allows the more detailed analytical study of the mean weight behavior shown in the next section.

B. Special Case of a White Input Signal

In general, the behavior of (28) can become very complex to be studied analytically [20]. In order to obtain analytical results that allow some understanding of the mean weight behavior, we study here the particular case of $x(n)$ i.i.d. and drawn from a zero-mean distribution. Unit variance σ_x^2 is also assumed without loss of generality. Using $\boldsymbol{R}_x = \boldsymbol{I}$ in (28) yields the component-wise expression

$$E\left\{v_i(n+1)\right\} = (1-\eta\alpha_i^*)\,E\left\{v_i(n)\right\} - \eta E\left\{v_i(n)\right\}^2. \quad (30)$$

Function $E\{v_i(n+1)\}$ in (30) is a parabola in $E\{v_i(n)\}$ with roots at $E\{v_i(n)\} = 0$ and $E\{v_i(n)\} = \frac{(1-\eta\alpha_i^*)}{\eta}$. It reaches its maximum $\frac{(1-\eta\alpha_i^*)^2}{4\eta}$ at $\frac{(1-\eta\alpha_i^*)}{2\eta}$. Fixed points are found by solving $E\{v_i(n+1)\} = E\{v_i(n)\}$, which yields $E\{v_i(n)\} = 0$ or $E\{v_i(n)\} = -\alpha_i^*$. This result is consistent with solution (18) where

$$v_i^o = \begin{cases} 0 & \text{if } \alpha_i^* \geq 0 \\ -\alpha_i^* & \text{otherwise} \end{cases} \quad (31)$$

with v_i^o the ith entry of $\boldsymbol{v}^o = \boldsymbol{\alpha}^o - \boldsymbol{\alpha}^*$.

Let us derive conditions ensuring convergence of (30) to 0 and $-\alpha_i^*$. Writing $u(n) = \frac{\eta E\{v_i(n)\}}{(1-\eta\alpha_i^*)}$, where the index i has been dropped to simplify the notation, we obtain the following difference equation known as the *logistic map* [20]–[22]

$$u(n+1) = \rho u(n)\left(1-u(n)\right) \quad (32)$$

with $\rho = 1 - \eta\alpha_i^*$, which is assumed nonzero. Fixed points defined in (31) now correspond to $u = 0$ and $u = \frac{\rho-1}{\rho}$, respectively. Convergence of the logistic map to these values depends on parameter ρ and on the initial condition $u(0)$ as follows. See [20]–[22] for details and Fig. 2 for illustration.

Case 1) $0 < \rho < 1$
An illustration of this case is shown in Fig. 2 (left). The fixed point $u = 0$ attracts all the trajectories originating in the interval $\left]\frac{(\rho-1)}{\rho}; \frac{1}{\rho}\right[$. The logistic map $u(n)$ is identically equal to $\frac{(\rho-1)}{\rho}$ for $n \geq 1$ if $u(0) = \frac{(\rho-1)}{\rho}$ or $u(0) = \frac{1}{\rho}$. Outside this interval, it diverges to $-\infty$.

Case 2) $\rho = 1$
The fixed point $u = 0$ attracts all the trajectories originating in the interval [0;1]. The logistic map $u(n)$ is identically equal to 0 for $n \geq 1$ if $u(0) = 0$ or 1. It diverges to $-\infty$ if $u(0) \notin [0;1]$.

Case 3) $1 < \rho \leq 3$
An illustration of this case is shown in Fig. 2 (right). The fixed point $u = \frac{\rho-1}{\rho}$ attracts all the trajectories originating in]0;1[. With the initial conditions $u(0) = 0$ or $u(0) = 1$, we have $u(n) = 0$ for all $n > 0$. It can be shown that the logistic map diverges to $-\infty$ if $u(0) \notin [0;1]$.

Case 4) $\rho > 3$
Fixed points become unstable. New fixed points appear between which the system alternates in stable cycles of period 2^k, with k tending to infinity as ρ

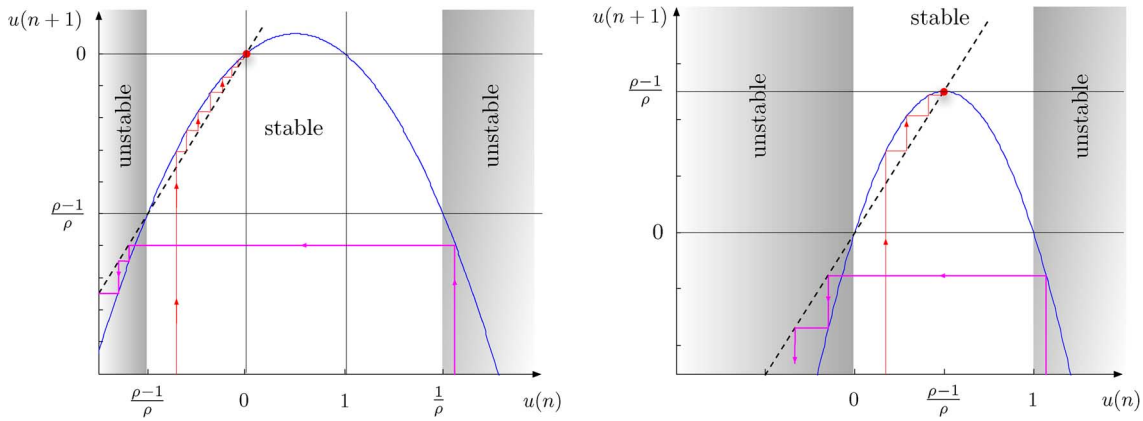

Fig. 2. Convergence of the logistic map, in the Case 1 (left) and in the Case 3 (right). The dashed line is the line of equation $u(n+1) = u(n)$.

increases. This case may even lead to a chaotic behavior, and falls out of the scope of our study.

To derive conditions for convergence of the difference (30) to 0 or $-\alpha_i^*$, we must consider separately components of $E\{v_i(n)\}$ associated with positive or negative unconstrained optimum α_i^*, respectively. On the one hand, based on the analysis of the logistic map (32), convergence of (30) to 0 corresponds to the conditions on ρ and $u(0)$ satisfying Case 1 and Case 2 above. This yields

$$0 < \eta < \frac{1}{\alpha_i^*} \quad -\alpha_i^* < v_i(0) < \frac{1}{\eta} \tag{33}$$

in the case where $\alpha_i^* > 0$. If $\alpha_i^* = 0$, these two conditions become $\eta > 0$ and $0 < v_i(0) < \frac{1}{\eta}$. On the other hand, ρ and $u(0)$ must obey the conditions presented in Case 3 for convergence of (30) to $-\alpha_i^*$. This leads to

$$0 < \eta \leq -\frac{2}{\alpha_i^*} \quad 0 < v_i(0) < \frac{1}{\eta} - \alpha_i^* \tag{34}$$

in the case where $\alpha_i^* < 0$. Finally, combining these inequalities leads to the following theoretical conditions for convergence of $E\{\boldsymbol{v}(n)\}$:

$$0 < \eta \leq \min_i \frac{1}{|\alpha_i^*|} \quad \text{and} \quad 0 < v_i(0) < \frac{1}{\eta} \quad \text{for all } i \tag{35}$$

or, using also (33) and (34), for convergence of $E\{\boldsymbol{\alpha}(n)\}$

$$0 < \eta \leq \min_i \frac{1}{|\alpha_i^*|} \quad \text{and} \quad 0 < \alpha_i(0) < \frac{1}{\eta} \quad \text{for all } i. \tag{36}$$

Conditions (35) and (36) on $v_i(0)$ and $\alpha_i(0)$ show that there is more freedom in choosing $\alpha_i(0)$ for small values of η. They guarantee convergence of the difference (30).

C. Simulation Examples for the First-Order Moment Analysis

This section presents simulation examples to verify the validity of the first-order moment analysis of the nonnegative LMS algorithm. We illustrate the accuracy of the model (30) through a first example where inputs $x(n)$ and noise $z(n)$ are i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_z^2 = 10^{-2}$, respectively. The impulse response $\boldsymbol{\alpha}^*$ is given by

$$\boldsymbol{\alpha}^* = [0.8\ 0.6\ 0.5\ 0.4\ 0.3\ 0.2\ 0.1\ -0.1\ -0.3\ -0.6]^\top \tag{37}$$

The initial impulse response $\boldsymbol{\alpha}(0)$ is drawn from the uniform distribution $\mathcal{U}([0;1])$, and kept unchanged for all the simulations. The algorithm's stability limit was determined experimentally to be $\eta_{\max} \approx 5 \times 10^{-3}$. As usually happens with adaptive algorithms, this limit is more restrictive than the mean weight convergence limit given by (36), as stability is determined by the weight fluctuations [19]. The mean value $E\{\alpha_i(n)\}$ of each coefficient is shown in Fig. 3 for $\eta = 10^{-3} = \frac{\eta_{\max}}{5}$ and $\eta = 5 \times 10^{-5} = \frac{\eta_{\max}}{10}$. The simulation curves (solid line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (dashed line) were obtained from model (30). One can notice that all the curves are perfectly superimposed and, as predicted by the result (18), each coefficient $\alpha_i(n)$ tends to $\{\alpha_i^*\}_+$ as n goes to infinity.

It is interesting to note that convergence towards the solution $\{\boldsymbol{\alpha}^*\}_+$ agrees with the theoretically predicted behavior of (32). For each positive entry α_i^* of $\boldsymbol{\alpha}^*$, the corresponding value of $\rho = 1 - \eta\alpha_i^*$ is in]0;1[. This corresponds to Case 1 in Section III-B, where the fixed point $u = 0$ attracts all the trajectories and $v_i(n)$ converges to zero. It can also be verified that each ρ associated with a negative entry α_i^* is in]1;3]. This corresponds to Case 3 where $u = \frac{(\rho-1)}{\rho}$ attracts all the trajectories and $\lim_{n\to\infty} v_i(n) = -\alpha_i^*$.

The second simulation example illustrates the accuracy of the model (30) for inputs $x(n)$ correlated in time. We consider a first-order AR model given by

$$x(n) = ax(n-1) + w(n)$$

with $a = \frac{1}{2}$. The noise $w(n)$ is i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_w^2 = 1 - \frac{1}{4}$, so that $\sigma_x^2 = 1$ as in the first example. The other parameters of the initial experimental setup remain unchanged, except for the step size values. In order to verify the model's accuracy also for large step sizes we performed the simulations for $\eta = 2.5 \times 10^{-3} = \frac{\eta_{\max}}{2}$ and $\eta = 5 \times 10^{-5} = \frac{\eta_{\max}}{10}$. The mean value $E\{\alpha_i(n)\}$ of each coefficient is shown in Fig. 4. As before, the simulation curves (solid line) and the theoretical curves (dashed line) are superimposed. It can be noticed that $\boldsymbol{\alpha}(n)$ no longer converges to $\{\boldsymbol{\alpha}^*\}_+$ since the input samples $x(n)$ are now correlated. We can easily verify that $E\{e^2(n)\} = 4.97$ dB using $\{\boldsymbol{\alpha}^*\}_+$, and

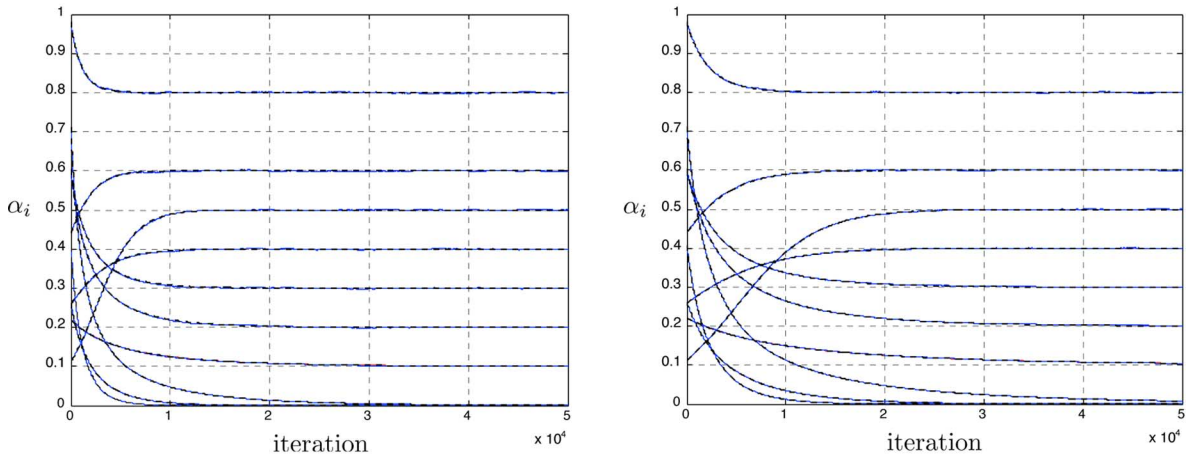

Fig. 3. Convergence of the coefficients $\alpha_i(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered: $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure. The theoretical curves (dashed line) obtained from (30) and simulation curves (solid line) are perfectly superimposed.

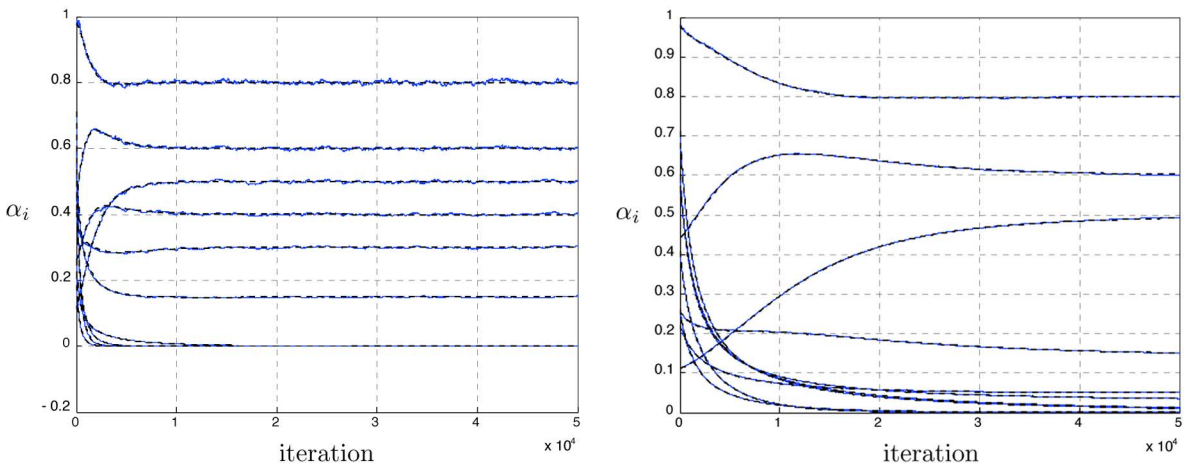

Fig. 4. Same experiment as in Fig. 3 except that input sequence $x(n)$ is generated by a first-order AR process. Two different step sizes are considered: $\eta = 2.5 \times 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure.

$E\{e^2(n)\} = 3.82$ dB at convergence of the nonnegative LMS algorithm.

IV. Second-Order Moment Analysis

We now present a model for the behavior of the second-order moments of the adaptive weights. To allow further analysis progress, we assume in this section that the input $x(n)$ is Gaussian.

A. Second Moment Behavior Model

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

$$\begin{aligned} E\left\{e^2(n)\right\} &= E\left\{\left(z(n) - \boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right)\left(z(n) - \boldsymbol{v}^\top(n)\boldsymbol{x}(n)\right)\right\} \\ &= \sigma_z^2 + E\left\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\right\} \\ &\approx \sigma_z^2 + \operatorname{trace}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}. \end{aligned} \tag{38}$$

Equation (26) clearly shows that the mean behavior of each coefficient is a function of a single diagonal entry of the matrix $\boldsymbol{R}_x\boldsymbol{K}(n)$. In this case, approximation (28) could be used without compromising the accuracy of the resulting mean behavior model. This accuracy has been verified through Monte Carlo simulations in Section III-C. The MSE in (38), however, is a function of the trace of $\boldsymbol{R}_x\boldsymbol{K}(n)$. Thus, the effect of the second order moments of the weight-error vector entries on the MSE behavior becomes more significant than in (26), and in general cannot be neglected. Thus, we determine a recursion for $\boldsymbol{K}(n)$ starting again from the weight error update (21).

Premultiplying (21) by its transpose, taking the expected value, and using the statistical properties of $z(n)$,[3] yields

$$\begin{aligned} \boldsymbol{K}(n+1) = \boldsymbol{K}(n) &- \eta\boldsymbol{P}_1(n)\boldsymbol{K}(n) - \eta\boldsymbol{K}(n)\boldsymbol{P}_1^\top(n) \\ &+ \eta^2\sigma_z^2\boldsymbol{P}_2(n) + \eta^2\sigma_z^2\left[\boldsymbol{P}_3(n) + \boldsymbol{P}_3^\top(n)\right] \\ &+ \eta^2\sigma_z^2\boldsymbol{P}_4(n) - \eta\left[\boldsymbol{P}_5(n) + \boldsymbol{P}_5^\top(n)\right] \\ &+ \eta^2\boldsymbol{P}_6(n) + \eta^2\boldsymbol{P}_7(n) \\ &+ \eta^2\boldsymbol{P}_8(n) + \eta^2\boldsymbol{P}_9(n) \end{aligned} \tag{39}$$

[3]The two important properties of $z(n)$ used in evaluating (39) are its independence of any other signal and its zero-mean.

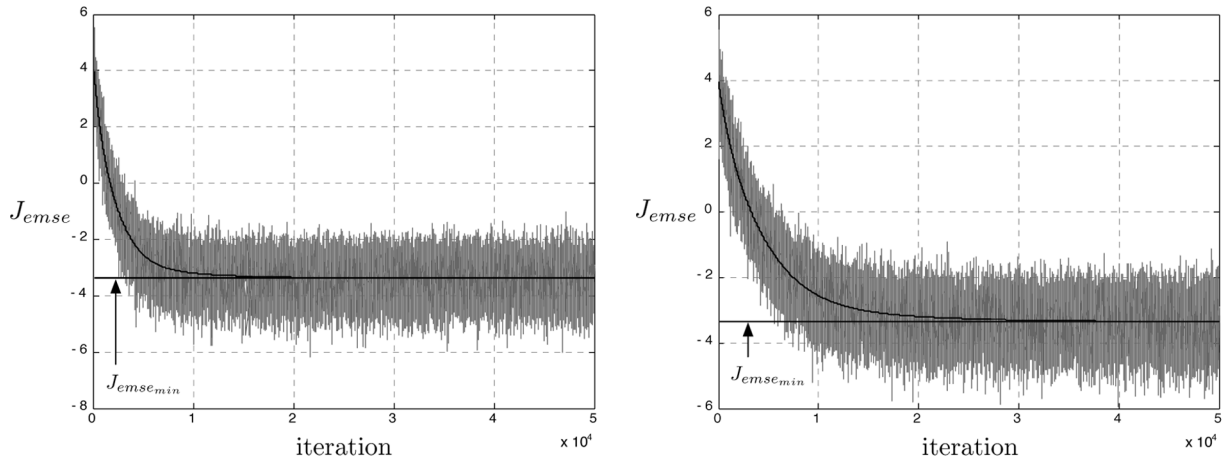

Fig. 5. Convergence of $J_{emse}(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered: $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure. The theoretical curves (black line) obtained from (38) and (39) and simulation curves (gray line) are perfectly superimposed.

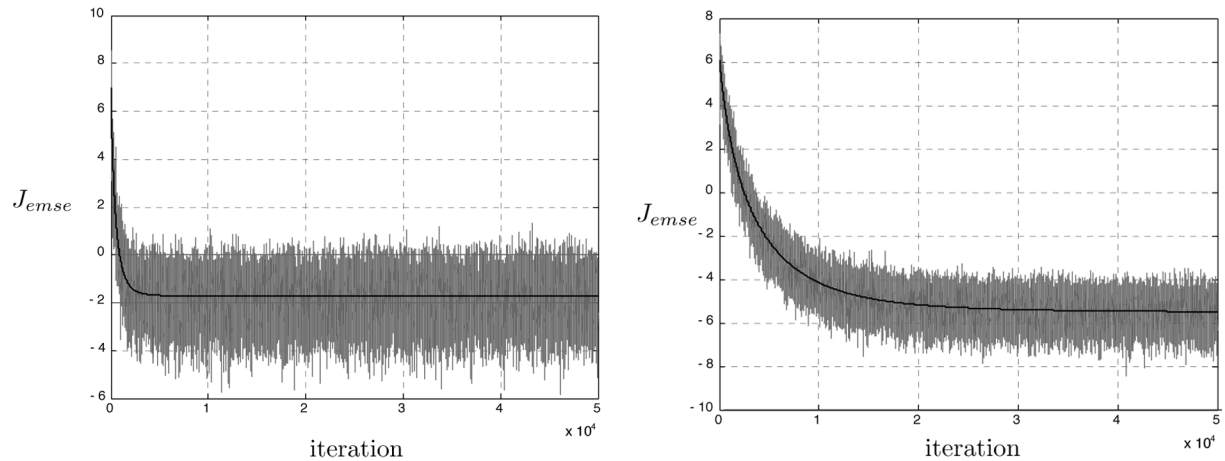

Fig. 6. Same experiment as in Fig. 5 except that input sequence $x(n)$ is generated by a first-order AR process. Two different step sizes are considered: $\eta = 2.5 \times 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure.

where matrices $\boldsymbol{P}_1$ to $\boldsymbol{P}_9$ are defined by

$$\boldsymbol{P}_1 = E\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\} \tag{40}$$

$$\boldsymbol{P}_2 = E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\} \tag{41}$$

$$\boldsymbol{P}_3 = E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\} \tag{42}$$

$$\boldsymbol{P}_4 = E\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\} \tag{43}$$

$$\boldsymbol{P}_5 = E\{\boldsymbol{v}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\} \tag{44}$$

$$\boldsymbol{P}_6 = E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\} \tag{45}$$

$$\boldsymbol{P}_7 = E\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\} \tag{46}$$

$$\boldsymbol{P}_8 = E\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{\alpha}_*^\top\boldsymbol{D}_x(n)\} \tag{47}$$

$$\boldsymbol{P}_9 = E\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\}\ . \tag{48}$$

The expected values in (40)–(48) are calculated in the following. In order to keep the calculations mathematically tractable, the following statistical assumptions are employed:

A1: The input vector $\boldsymbol{x}(n)$ is zero-mean Gaussian.

A2: The weight-error vector $\boldsymbol{v}(n)$ is statistically independent of $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$. The reasoning for this approximation has been discussed in detail in [23].

A3: The statistical dependence of $\boldsymbol{v}(n)\boldsymbol{v}^\top(n)$ and $\boldsymbol{v}(n)$ is neglected. This assumption follows the same reasoning valid for **A2**, see [23].

A4: $\boldsymbol{v}(n)$ and $(\boldsymbol{x}^\top(n)\boldsymbol{v}(n))^2$ are statistically independent given **A2**. This is because $(\boldsymbol{x}^\top(n)\boldsymbol{v}(n))^2$ is a linear combination of the entries of $\boldsymbol{v}(n)\boldsymbol{v}^\top(n)$. Thus, this approximation follows basically the same reasoning discussed in [23] to justify **A2**.

$\boldsymbol{P}_1$: This expected value has been already calculated in (23). Remember that

$$\boldsymbol{P}_1 = E\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\} = \boldsymbol{D}_{\alpha^*}\boldsymbol{R}_x. \tag{49}$$

$\boldsymbol{P}_2$: Basic linear algebra gives

$$\begin{aligned}\boldsymbol{P}_2 &= E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\} \\ &= E\{\boldsymbol{D}_{\alpha^*}\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{D}_{\alpha^*}\} \\ &= \boldsymbol{D}_{\alpha^*}\boldsymbol{R}_x\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{50}$$

$\boldsymbol{P}_3$: Neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$ yields

$$\begin{aligned}\boldsymbol{P}_3 &= E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\}\\ &\approx E\left\{\boldsymbol{D}_v(n)\right\}\boldsymbol{R}_x\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{51}$$

$\boldsymbol{P}_4$: The (i,j)th entry of the matrix within the expectation in $\boldsymbol{P}_4$ is given by

$$\begin{aligned}&\left[\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\right]_{ij}\\ &\qquad = x(n-i+1)v_i(n)v_j(n)x(n-j+1).\end{aligned} \tag{52}$$

Using **A2**, $E\{x(n-i+1)v_i(n)v_j(n)x(n-j+1)\}\approx E\{x(n-i+1)x(n-j+1)\}E\{v_i(n)v_j(n)\}$ and

$$\boldsymbol{P}_4 \approx \boldsymbol{R}_x \circ \boldsymbol{K}(n) \tag{53}$$

where $\circ$ denotes the so-called Hadamard product.

$\boldsymbol{P}_5$: Defining $\boldsymbol{D}_v(n)$ as the diagonal matrix with diagonal entries given by $\boldsymbol{v}(n)$, we first note that

$$\begin{aligned}&E\left\{\boldsymbol{v}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\right\}\\ &\qquad = E\left\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{D}_v(n)\right\}.\end{aligned} \tag{54}$$

Now, using **A2** and **A3**, the expectation can be approximated as

$$\begin{aligned}&E\left\{\boldsymbol{v}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\right\}\\ &\qquad \approx E\left\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}E\left\{\boldsymbol{D}_v(n)\right\}.\end{aligned} \tag{55}$$

Finally, using again **A2** we obtain

$$\boldsymbol{P}_5 \approx \boldsymbol{K}(n)\boldsymbol{R}_x E\left\{\boldsymbol{D}_v(n)\right\}. \tag{56}$$

$\boldsymbol{P}_6$: Basic linear algebra gives

$$\begin{aligned}\boldsymbol{P}_6 &= E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\}\\ &= \boldsymbol{D}_{\alpha^*}E\left\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{57}$$

Under **A1** and applying the same methodology used to derive [24, Eq. (29)],

$$\begin{aligned}\boldsymbol{P}_6 &\approx \boldsymbol{D}_{\alpha^*}\left(2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x + E\left\{\boldsymbol{v}^\top(n)\boldsymbol{R}_x\boldsymbol{v}(n)\right\}\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}\\ &= \boldsymbol{D}_{\alpha^*}\left(2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x + E\left\{\mathrm{trace}\left\{\boldsymbol{v}^\top(n)\boldsymbol{R}_x\boldsymbol{v}(n)\right\}\right\}\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}\\ &= \boldsymbol{D}_{\alpha^*}\left(2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\boldsymbol{K}(n)\}\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{58}$$

$\boldsymbol{P}_7$: Using basic algebra, **A2** and **A3** as done to obtain (55), we have

$$\begin{aligned}\boldsymbol{P}_7 &= E\left\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\right\}\\ &\approx \boldsymbol{D}_{\alpha^*}E\left\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}\\ &\quad\times E\left\{\boldsymbol{D}_v(n)\right\}.\end{aligned} \tag{59}$$

Finally, under **A1** and applying the same methodology as in [24, Equation (29)], yields

$$\boldsymbol{P}_7 \approx \boldsymbol{D}_{\alpha^*}\left(2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x + \mathrm{trace}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}\boldsymbol{R}_x\right)E\left\{\boldsymbol{D}_v(n)\right\}. \tag{60}$$

$\boldsymbol{P}_8$: Using basic algebra we obtain

$$\begin{aligned}\boldsymbol{P}_8 &= E\left\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\right\}\\ &= E\left\{\boldsymbol{D}_v(n)\left(\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\right)^2\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{61}$$

Using **A4**, $\boldsymbol{P}_8$ becomes

$$\boldsymbol{P}_8 \approx E\left\{\boldsymbol{D}_v(n)\right\}E\left\{\left(\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\right)^2\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}\boldsymbol{D}_{\alpha^*}. \tag{62}$$

The expected value $E\{(\boldsymbol{x}^\top(n)\boldsymbol{v}(n))^2\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\}$ for zero-mean Gaussian signal $\boldsymbol{x}(n)$ has already been evaluated in [24, eqs. (7)–(9)], using results from [25]. Following the same procedure as in [24] yields

$$\begin{aligned}&E\left\{\left(\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\right)^2\boldsymbol{x}(n)\boldsymbol{x}^\top(n)|\boldsymbol{v}(n)\right\}\\ &\approx \boldsymbol{v}^\top(n)\boldsymbol{R}_x\boldsymbol{v}(n)\boldsymbol{R}_x + 2\boldsymbol{R}_x\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{R}_x\\ &= \mathrm{trace}\left\{\boldsymbol{R}_x\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\right\}\boldsymbol{R}_x + 2\boldsymbol{R}_x\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{R}_x.\end{aligned} \tag{63}$$

Now, taking the expected value with respect to $\boldsymbol{v}(n)$

$$\begin{aligned}&E\left\{\left(\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\right)^2\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right\}\\ &\qquad \approx \mathrm{trace}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}\boldsymbol{R}_x + 2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x.\end{aligned} \tag{64}$$

Then we obtain the final result

$$\boldsymbol{P}_8 \approx E\left\{\boldsymbol{D}_v(n)\right\}\left(\mathrm{trace}\left\{\boldsymbol{R}_x\boldsymbol{K}(n)\right\}\boldsymbol{R}_x + 2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}. \tag{65}$$

$\boldsymbol{P}_9$: Computing the (i,j)th entry of matrix $\boldsymbol{P}_9$ within the expectation, and using **A2**, yields

$$\begin{aligned}\left[\boldsymbol{P}_9\right]_{ij} &= \sum_\ell\sum_k E\left\{x(n-i+1)\left[\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\right]_{ik}\right.\\ &\qquad\times\left[\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\right]_{k\ell}\left[\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\right]_{\ell j}\\ &\qquad\left.\times\, x(n-j+1)\right\}\\ &= \sum_\ell\sum_k E\left\{x(n-i+1)x(n-k+1)x(n-\ell+1)\right.\\ &\qquad\left.\times\, x(n-j+1)\right\}\\ &\quad\times E\left\{v_i(n)v_j(n)v_k(n)v_\ell(n)\right\}\end{aligned} \tag{66}$$

For $\boldsymbol{x}(n)$ a zero-mean Gaussian signal (**A1**), we know that [26]

$$\begin{aligned}&E\left\{x(n-i+1)x(n-k+1)x(n-\ell+1)x(n-j+1)\right\}\\ &\quad = r_x(k-i)r_x(j-\ell) + r_x(\ell-i)r_x(j-k)\\ &\qquad + r_x(j-i)r_x(\ell-k).\end{aligned} \tag{67}$$

The expectation $E\{v_i(n)v_j(n)v_k(n)v_\ell(n)\}$ cannot be evaluated directly, as the statistics of $\boldsymbol{v}(n)$ are unknown. Approximate expressions can be obtained using numerous different approaches. We have chosen to use the following approximation which preserves relevant information about the second moment behavior of the adaptive weights while keeping the mathematical problem tractable. We write

$$\begin{aligned}E\{v_i(n)v_j(n)v_k(n)v_\ell(n)\} &= E\{v_i(n)v_j(n)\}\,E\{v_k(n)v_\ell(n)\}\\ &\quad+\mathrm{Cov}\{v_i(n)v_j(n), v_k(n)v_\ell(n)\}.\end{aligned} \tag{68}$$

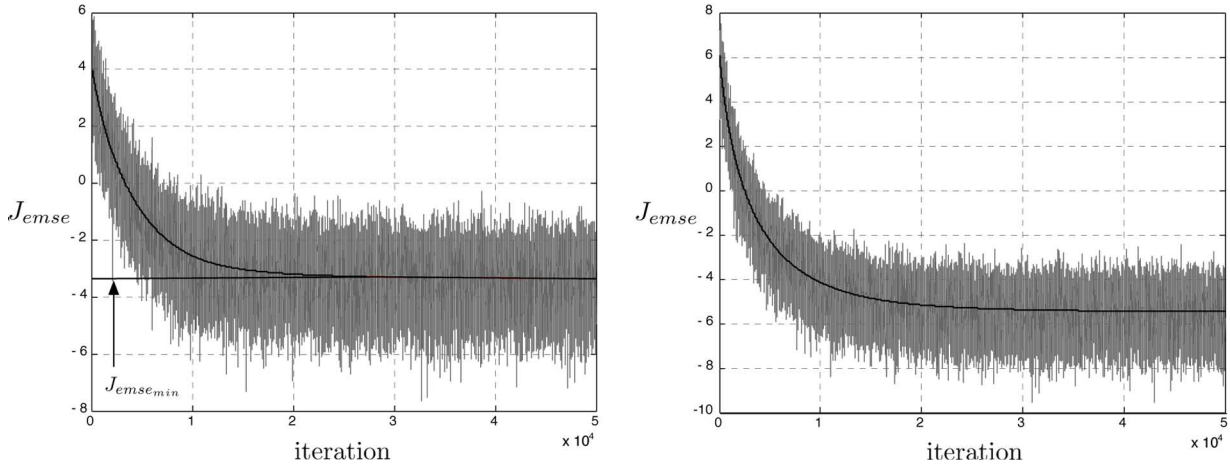

Fig. 7. Convergence of $J_{emse}(n)$ with step size $\eta = 5 \times 10^{-4}$, in the case where input $x(n)$ is i.i.d. on the left figure, and generated by a first-order AR process on the right figure. Compared to Figs. 5 (right) and 6 (right), the variance of the noise $z(n)$ has been increased from 10^{-2} to 1.

Now, writing

$$v_i(n+1)v_j(n+1) = (v_i(n) + \eta\Delta v_i(n))\,(v_j(n) + \eta\Delta v_j(n)) \tag{69}$$

we see that the fluctuations in $v_i(n+1)v_j(n+1)$ are proportional to η. Using the same reasoning for $v_k(n)v_\ell(n)$ we note that the covariance in (68) is proportional to η^2. The higher order moments of the entries of $\boldsymbol{v}(n)$ in (68) will then be proportional to η^p with $p \geq 2$. Thus, for sufficiently small values of η, neglecting these terms yields the approximation

$$E\{v_i(n)v_j(n)v_k(n)v_\ell(n)\} \approx E\{v_i(n)v_j(n)\}\,E\{v_k(n)v_\ell(n)\}\,. \tag{70}$$

This approximation is supported by the simulation results presented in Section IV-B. Substituting the two equations above into the expression of $[\boldsymbol{P}_9]_{ij}$ leads to

$$\begin{aligned}[\boldsymbol{P}_9]_{ij} \approx & r_x(j-i)\sum_\ell\sum_k r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_\ell\sum_k r_x(k-i)r_x(j-\ell)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_\ell\sum_k r_x(\ell-i)r_x(j-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij}\,. \end{aligned} \tag{71}$$

The first right-hand term of (71) can be expressed as follows

$$\begin{aligned} & r_x(j-i)\sum_\ell\sum_k r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & = [\boldsymbol{R}_x]_{ij}\sum_k\left(\sum_\ell [\boldsymbol{R}_x]_{k\ell}\,[\boldsymbol{K}(n)]_{k\ell}\right)[\boldsymbol{K}(n)]_{ij} \\ & = [\mathrm{trace}\,\{\boldsymbol{R}_x\boldsymbol{K}(n)\}\,\boldsymbol{R}_x]_{ij}\,[\boldsymbol{K}(n)\}]_{ij}\,. \end{aligned} \tag{72}$$

The second and third right-hand terms write

$$\begin{aligned} & \sum_\ell\sum_k r_x(k-i)r_x(j-\ell)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & = \left(\sum_\ell\sum_k [\boldsymbol{R}_x]_{ik}\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{R}_x]_{\ell j}\right)[\boldsymbol{K}(n)]_{ij} \\ & = [\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x]_{ij}\,[\boldsymbol{K}(n)\}]_{ij}\,. \end{aligned} \tag{73}$$

This leads to the following close-form expression:

$$[\boldsymbol{P}_9] = (\mathrm{trace}\,\{\boldsymbol{R}_x\boldsymbol{K}(n)\}\,\boldsymbol{R}_x + 2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x) \circ \boldsymbol{K}(n). \tag{74}$$

Using the expected values $\boldsymbol{P}_1$ to $\boldsymbol{P}_9$ in (39), we finally obtain a recursive analytical model for the behavior of $\boldsymbol{K}(n)$. This result can be used to study the convergence properties of $E\{e^2(n)\}$, and can be applied to design.[4] The next section illustrates the model accuracy in predicting the nonnegative LMS algorithm behavior.

B. Simulations for the Second-Order Moment Analysis

This section presents simulation examples to check the accuracy of model (39). Figs. 5 and 6 show the behavior of the excess MSE $J_{emse}(n) = \mathrm{trace}\{\boldsymbol{R}_x\boldsymbol{K}(n)\}$ corresponding to the experiments that has been conducted in Section III-C. The simulation curves (gray line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (black line) were obtained from model (39). Note the model's accuracy even for step sizes as large as $\frac{\eta_{\max}}{2}$ (left side of Fig. 6). Also note that the theoretical value of the minimum excess mean-square error $J_{emse_{min}}$ is represented in Fig. 5.[5] It can be observed that $J_{emse}(n)$ tends to $J_{emse_{min}}$ as n goes to infinity. Fig. 7 highlights the performance of the model for uncorrelated and correlated input signals $x(n)$ through the same experimental setup as

[4]This model can also be used in (26) for the mean weight behavior if needed. However, our experience has been that the simplified model given by (28) suffices for predicting the mean weight behavior for most practical needs. It also makes the analytical study presented in Section III-B tractable.

[5]It can be shown, from (17)–(19), that $J_{emse_{min}} = \|\boldsymbol{\alpha}^* - (\boldsymbol{\alpha}^*)_+\|^2$ in the case where $\boldsymbol{R}_x = \boldsymbol{I}$.

described before, except that the noise variance σ_z^2 is now set to 1. All these experiments illustrate the accuracy of the model, which can provide important guidelines for the use of the nonnegative LMS algorithm in practical applications.

V. Conclusion

In many real-life phenomena, due to the inherent physical characteristics of systems under investigation, nonnegativity is a desired constraint that can be imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. In this paper, we proposed a general method for system identification under nonnegativity constraints, and we derived the so-called nonnegative LMS based on stochastic gradient descent. This algorithm switches automatically between a gradient descent mechanism and a gradient ascent one depending whether the nonnegativity constraint is violated or not. Finally, we analyzed the algorithm convergence in the mean sense and in the mean-square sense. In future research efforts, we intend to explore these models in practical applications since they provide important guidelines to algorithm designers. We also plan to derive variants of this approach, e.g., in the spirit of the normalized-LMS and the sign-LMS algorithms.

References

[1] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero, "Nonnegative least-squares image deblurring: Improved gradient projection approaches," *Inverse Problems*, vol. 26, no. 1, 2010.

[2] M. H. Van Benthem and M. R. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *J. Chemometr.*, vol. 18, pp. 441–450, 2004.

[3] 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, 1999.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 556–562, 2001.

[5] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations*. New York: Wiley, 2009.

[6] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computat. Statist. Data Anal.*, vol. 52, no. 1, pp. 155–173, 2007.

[7] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 534–543, 2003.

[8] S. Moussaoui, D. Brie, A. Mohammad-Djafari, and C. Carteret, "Separation of non-negative mixture of non-negative sources using a Bayesian approach and MCMC sampling," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4133–4145, 2006.

[9] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA: Society for Indust. and Appl. Math., 1995.

[10] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *J. Chemometr.*, vol. 11, no. 5, pp. 393–401, 1997.

[11] J. B. Rosen, "The gradient projection method for nonlinear programming. Part 1: Linear constraints," *J. Soc. Indust. Appl. Math.*, vol. 8, no. 1, pp. 181–217, 1960.

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

[13] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.

[14] 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 Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[15] C. J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computat.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[16] C. J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, 2007.

[17] 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 Process.*, vol. 81, no. 5, pp. 945–974, 2001.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[19] A. Sayed, *Adaptive Filters*. New York: Wiley-Intersci., 2008.

[20] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 10, pp. 459–467, 1976.

[21] K. Alligood, T. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*. New York: Springer-Verlag, 1997.

[22] D. Perrin, La suite logistique et le chaos Univ. de Paris-Sud, France, 2008, Tech. Rep., Dép. Math. d'Orsay.

[23] J. Minkoff, "Comment: On the unnecessary assumption of statistical independence between reference signal and filter weights in feedforward adaptive systems," *IEEE Trans. Signal Process.*, vol. 49, no. 5, p. 1109, May 2001.

[24] P. I. Hubscher and J. C. M. Bermudez, "An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 664–671, Mar. 2003.

[25] N. J. Bershad, P. Celka, and J.-M. Vesin, "Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output measurements," *IEEE Trans. Signal Process.*, vol. 47, no. 3, pp. 675–689, Mar. 1999.

[26] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

Jie Chen was born in Xi'an, China, in 1984. He received the Dipl.-Ing. and the M.S. degrees in 2009 from the University of Technology of Troyes (UTT), France, and from Xi'an Jiaotong University, China, respectively, all in information and telecommunication engineering.

He is currently pursuing the Ph.D. degree at the UTT. He is conducting his research work at the Côte d'Azur Observatory, University of Nice Sophia-Antipolis, France. His current research interests include adaptive signal processing, kernel methods, and supervised and unsupervised learning.

Cédric Richard (S'98–M'01–SM'07) was born on January 24, 1970, in Sarrebourg, France. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from the University of Technology of Compiègne (UTC), France, all in electrical and computer engineering.

He joined the Côte d'Azur Observatory, University of Nice Sophia-Antipolis, France, in 2009. He is currently a Professor of electrical engineering. From 1999 to 2003, he was an Associate Professor at the University of Technology of Troyes (UTT), France. From 2003 to 2009, he was a Professor at the UTT, and the supervisor of a group consisting of 60 researchers and Ph.D. students. In winter 2009 and autumn 2010, he was a Visiting Researcher with the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianòpolis, Brazil. Prof. He is the author of more than 120 papers. His current research interests include statistical signal processing and machine learning.

Dr. Richard is a junior member of the Institut Universitaire de France since October 2010. He was the General Chair of the XXIth Francophone Conference GRETSI on Signal and Image Processing held in Troyes, France, in 2007, and of the IEEE Statistical Signal Processing Workshop (IEEE SSP'11) held in Nice, France, in 2011. Since 2005, he is a member of the Board of the Federative Cnrs Research Group ISIS on Information, Signal, Images, and Vision. He is a member of GRETSI Association Board and of the EURASIP society. He served as an Associate Editor of the IEEE Transactions on Signal Processing from 2006 to 2010, and of the EURASIP *Journal on Signal Processing* since 2009. In 2009, he was nominated liaison local officer for EURASIP, and member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society. He and P. Honeine received the Best Paper Award for "Solving the pre-image problem in kernel machines: a direct method" at the 2009 IEEE Workshop on Machine Learning for Signal Processing (IEEE MLSP'09).

José Carlos M. Bermudez (S'78–M'85–SM'02) received the B.E.E. degree from Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, the M.Sc. degree in electrical engineering from COPPE/UFRJ, and the Ph.D. degree in electrical engineering from Concordia University, Montreal, Canada, in 1978, 1981, 1985, respectively.

He joined the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 1985. He is currently a Professor of electrical engineering. In the winter of 1992, he was a Visiting Researcher with the Department of Electrical Engineering, Concordia University. In 1994, he was a Visiting Researcher with the Department of Electrical Engineering and Computer Science, University of California, Irvine (UCI). His research interests have involved analog signal processing using continuous-time and sampled-data systems. His recent research interests are in digital signal processing, including linear and nonlinear adaptive filtering, active noise and vibration control, echo cancellation, image processing, and speech processing.

Prof. Bermudez served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING in the area of adaptive filtering from 1994 to 1996 and from 1999 to 2001, as the Signal Processing Associate Editor for the *Journal of The Brazilian Telecommunications Society* (2005–2006) and as Associate Editor for the EURASIP *Journal on Advances in Signal Processing* (2006–2010). He was a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society from 1998 to 2004.

Paul Honeine (M'07) was born in Beirut, Lebanon, on October 2, 1977. He received the Dipl.-Ing. degree in mechanical engineering in 2002 and the M.Sc. degree in industrial control in 2003, both from the Faculty of Engineering, the Lebanese University, Lebanon. In 2007, he received the Ph.D. degree in systems optimisation and security from the University of Technology of Troyes, France.

He was a Postdoctoral Research Associate with the Systems Modeling and Dependability Laboratory, University of Technology of Troyes, from 2007 to 2008, and since then has been an Assistant Professor. His research interests include nonstationary signal analysis and classification, nonlinear signal processing, sparse representations, machine learning, and wireless sensor networks.

Dr. Honeine and C. Richard received the Best Paper Award for "Solving the pre-image problem in kernel machines: a direct method" at the 2009 IEEE Workshop on Machine Learning for Signal Processing (IEEE MLSP'09).