Non-negative least-mean-square algorithm

Jie Chen, Cédric Richard, José C. M. Bermudez, Paul Honeine

Non-negative least-mean-square algorithm

Jie Chen[1,2], Cédric Richard[1], *Senior Member, IEEE*,
Jose Carlos M. Bermudez[3], *Senior Member, IEEE*, Paul Honeine[2], *Member, IEEE*

[1] Université de Nice Sophia-Antipolis, UMR CNRS 6525, Observatoire de la Côte d'Azur
Laboratoire Hippolyte Fizeau, Parc Valrose, 06102 Nice cedex 2 - France
tel.: +33.4.92.07.63.94 fax.: +33.4.92.07.63.21
chenjieg@sina.com cedric.richard@unice.fr

[2] Université de Technologie de Troyes, UMR CNRS 6279
Laboratoire LM2S, BP 2060, 10010 Troyes cedex - France
tel.: +33.3.25.71.56.25 fax.: +33.3.25.71.56.99
paul.honeine@utt.fr

[3] 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

Dynamic system modeling plays a crucial role in the development of techniques for stationary and non-stationary signal processing. Due to the inherent physical characteristics of systems under investigation, non-negativity 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 non-negativity constraints. We derive the so-called *non-negative least-mean-square algorithm* 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.

This work has been partly supported by CNPq grant No. 305377/2009-4.

I. Introduction

In many real-life phenomena including biological and physiological ones, due to the inherent physical characteristics of systems under investigation, non-negativity 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 non-negative values (ppm, joule). Non-negativity as a physical constraint has received growing attention from the signal processing community during the last decade. For instance, consider the following non-negative least-square inverse problem

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

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 i-th 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 non-negative matrix factorization (NMF), which is now a popular dimension reduction technique [3], [4], [5]. Given a matrix $\boldsymbol{X}$ with non-negative entries, the squared error version of this problem can be stated as follows

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

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 non-negative mixture of non-negative sources has also been considered in [7], [8].

Over the last fifteen years, a variety of methods have been developed to tackle non-negative 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 non-negativity constraints. Another class of tools is the class of projected gradient algorithms [11], [12], [13], [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 non-negativity 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 non-negativity 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.

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 stationary and zero-mean. 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, non-negativity 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

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

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

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

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 Equation (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 Equation (5) leads us to the component-wise gradient descent algorithm

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

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 i-th 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. Let us assume that $\alpha_i(n) \geq 0$. Non-negativity of $\alpha_i(n+1)$ is guaranteed if

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

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

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

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

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

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 i-th entry is defined as follows

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

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 non-negativity of $\boldsymbol{\alpha}(n)$ for all n, but does not ensure the stability of the algorithm.

B. The non-negative least-mean-square algorithm

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

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

where we have included the non-negativity 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 non-negativity constraints is given by

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

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 to the stochastic approximation of (14) given by[1]

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

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

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

It is interesting to notice how the term $\boldsymbol{\alpha}(n)$ in the update term on the right-hand side (r.h.s.) of (15) affects the dynamics of the coefficient update when compared with the well know LMS algorithm [19]. Note that the extra multiplying factor $\alpha_i(n)$ in the update term of the i-th 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 i-th 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 Equation (15).

In the following, the adaptive weight behavior of the adaptive algorithm (15), called *non-negative 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 non-negative LMS algorithm. Figure 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

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

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

$$\boldsymbol{R}_x\, \boldsymbol{\alpha}^* = \boldsymbol{r}_{xy}. \tag{17}$$

The residual signal $z(n) = y(n) - \boldsymbol{x}^\top(n)\, \boldsymbol{\alpha}^*$ in Figure 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

$$\boldsymbol{\alpha}^o = \{\boldsymbol{\alpha}^*\}_+ \tag{18}$$

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

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

with σ_y^2 the variance of $y(n)$.

A. Mean weight behavior model

Defining the weight-error vector $\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^* = [v_1(n), v_2(n), \ldots, v_N(n)]^\top$, the update equation (15) can be written as

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\, e(n)\, \boldsymbol{D}_x(n)\, \big(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*\big). \tag{20}$$

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

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

Taking the expectation of expression (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

$$E\{\boldsymbol{v}(n+1)\} \approx \big(\boldsymbol{I} - \eta\, E\{\boldsymbol{D}_x(n)\, \boldsymbol{\alpha}^*\, \boldsymbol{x}^\top(n)\}\big)\, E\{\boldsymbol{v}(n)\} - \eta\, E\{\boldsymbol{D}_x(n)\, \boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\}. \tag{22}$$

The first expectation on the r.h.s. of (22) is given by

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

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

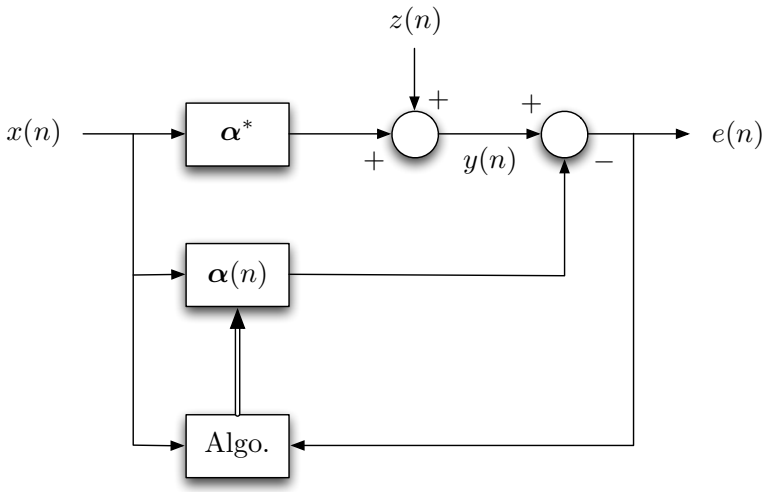

Fig. 1. Adaptive system under study

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

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

Taking the expectation of this expression, 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

$$\begin{aligned}[E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\}]_i &\approx \sum_{j=1}^{N} r_x(j-i)\,[\boldsymbol{K}(n)]_{ij} \\ &= [\boldsymbol{R}_x\,\boldsymbol{K}(n)]_{ii}\end{aligned} \tag{25}$$

This implies that $E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\} \approx \mathrm{diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}$, where $\mathrm{diag}\{\boldsymbol{A}\}$ denotes the vector whose i-th entry is defined by $[\boldsymbol{A}]_{ii}$. Using these results with Equation (22) yields the following mean weight-error vector update equation

$$E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} - \eta\,\mathrm{diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}. \tag{26}$$

This equation requires second-order moments defined by $\boldsymbol{K}(n)$ in order to update the first-order one $E\{\boldsymbol{v}(n)\}$. A recursive model will be derived for $\boldsymbol{K}(n)$ in Section IV, see Equation (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

$$\boldsymbol{K}(n) \approx E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}. \tag{27}$$

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

$$E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} - \eta\,\mathrm{diag}\{\boldsymbol{R}_x\,E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}\}. \tag{28}$$

Approximation (27) assumes that

$$\mathrm{Cov}\{v_i(n), v_j(n)\} \ll E\{v_i(n)\}E\{v_j(n)\} \tag{29}$$

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

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

Function $E\{v_i(n+1)\}$ in Equation (30) is a parabola in the variable $E\{v_i(n)\}$ with roots at $E\{v_i(n)\} = 0$ and $E\{v_i(n)\} = \frac{1-\eta\,\alpha_i^*}{\eta}$. It reaches its maximum value $\frac{(1-\eta\,\alpha_i^*)^2}{4\eta}$ at $E\{v_i(n)\} = \frac{1-\eta\,\alpha_i^*}{2\eta}$. 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

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

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

Let us now derive conditions ensuring convergence of (30) to 0 and $-\alpha_i^*$. Writing $u(n) = \frac{\eta}{1-\eta\,\alpha_i^*}E\{v_i(n)\}$, where the index i has been dropped to simplify the notation, we obtain the following difference equation known as the *logistic map* [20], [21], [22]

$$u(n+1) = \rho\,u(n)\,(1 - u(n)) \tag{32}$$

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 the parameter ρ and on the initial condition $u(0)$ as follows. See [20], [21], [22] for details and Figure 2 for illustration.

Case 1: $0 < \rho < 1$

An illustration of this case is shown in Figure 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 the interval, $u(n)$ 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 Figure 2 (right). The fixed point $u = \frac{\rho-1}{\rho}$ attracts all the trajectories

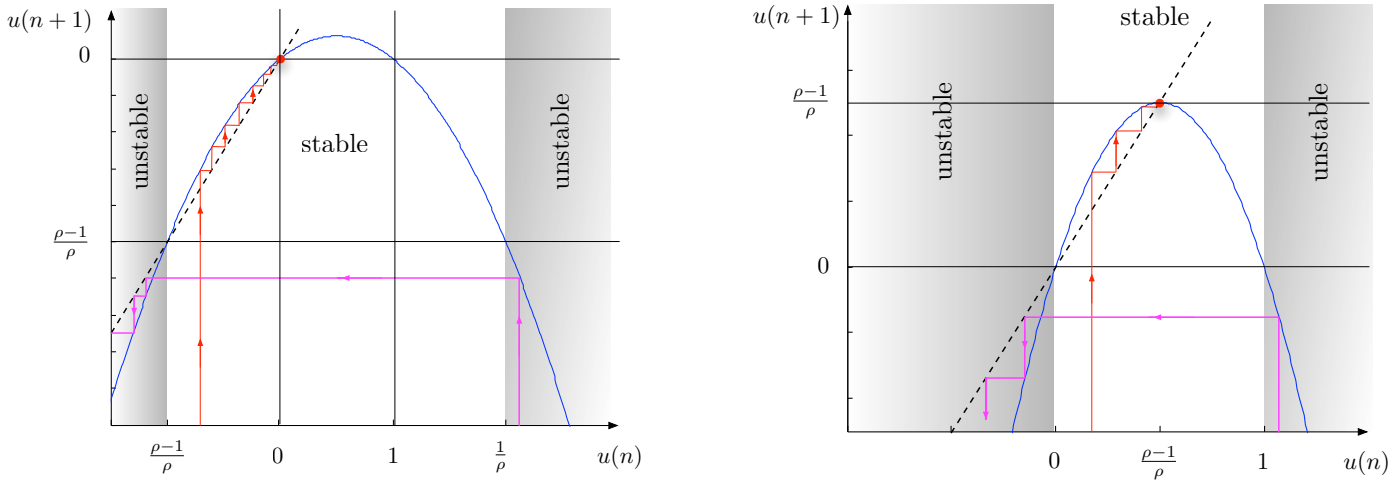

Fig. 2. Convergence of the logistic map, in the Case 1 (left) and in the Case 3 (right). The dashed line is the line of equation $u(n+1) = u(n)$.

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 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 equation (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

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

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 Equation (30) to $-\alpha_i^*$. This leads to

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

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

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

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

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

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 the convergence of the difference equation (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 non-negative 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

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

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 Figure 3 for $\eta = 10^{-3} = \eta_{\max}/5$ and $\eta = 5 \times 10^{-5} = \eta_{\max}/10$. The simulation curves (blue line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (red 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 $u = (\rho - 1)/\rho$ 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 $x(n) = a\,x(n-1) + w(n)$, 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 $\eta = 2.5 \times 10^{-3} = \eta_{\max}/2$ and $\eta = 5 \times 10^{-5} = \eta_{\max}/10$. The mean value $E\{\alpha_i(n)\}$ of each coefficient is shown in Figure 4. As before, the simulation curves (blue line) and the theoretical curves (red line) are superimposed.

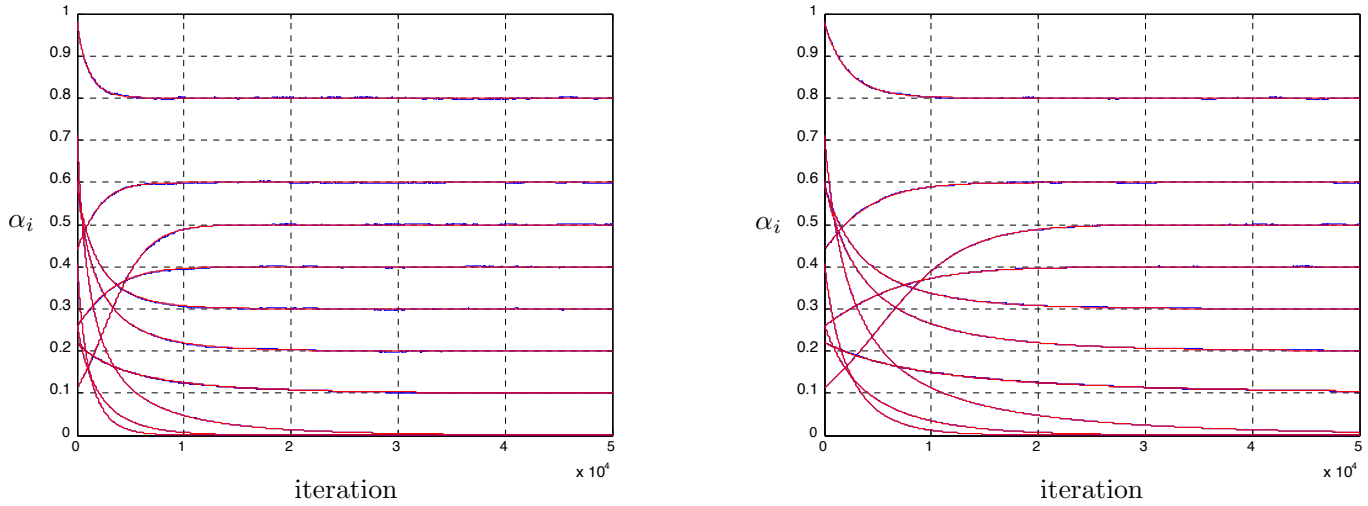

Fig. 3. Convergence of the coefficients $\alpha_i(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered: $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure. The theoretical curves (red line) obtained from the model (30)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

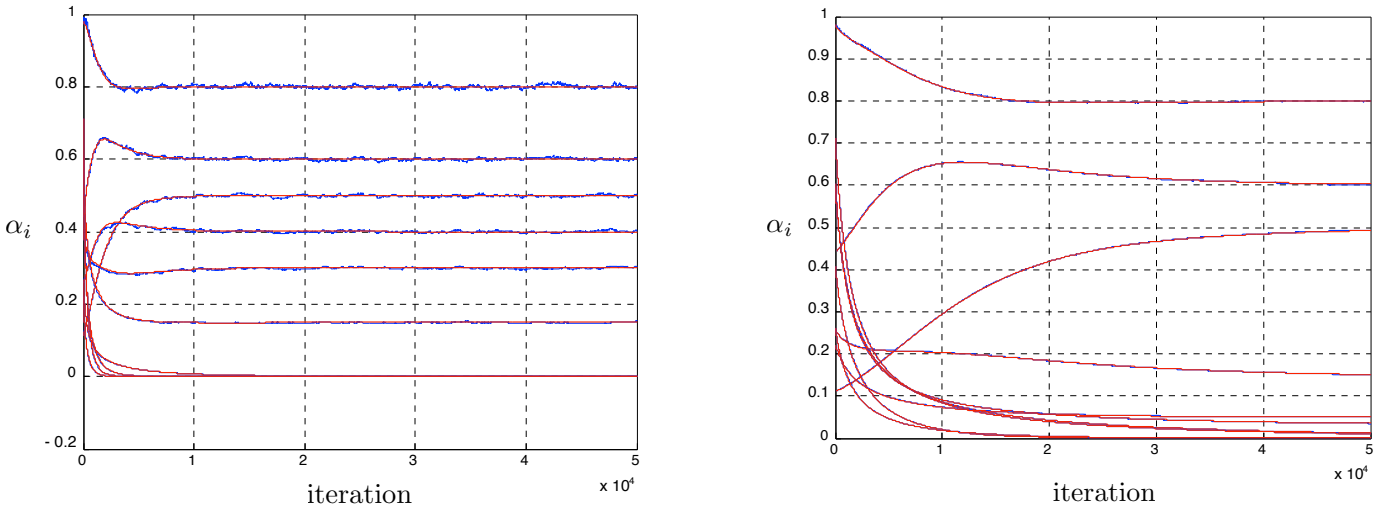

Fig. 4. Same experiment as in Figure 3 except that input sequence $x(n)$ is generated by a first-order AR process. Two different step sizes are considered: $\eta = 2.5 \times 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure.

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 $E\{e^2(n)\} = 3.82$ dB at convergence of the non-negative 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):

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

Equation (26) clearly shows that the mean behavior of each coefficient is a function of a single diagonal entry of 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 equation (21).

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

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

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

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

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

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

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

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

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

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

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

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

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

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 assuption **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**.

Expected value $\boldsymbol{P}_1$

This expected value has been already calculated in (23). Remember that

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

Expected value $\boldsymbol{P}_2$

Basic linear algebra gives

$$\boldsymbol{P}_2 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\} = E\{\boldsymbol{D}_{\alpha^*}\,\boldsymbol{x}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{D}_{\alpha^*}\} = \boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha^*}. \tag{50}$$

Expected value $\boldsymbol{P}_3$

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

$$\boldsymbol{P}_3 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\} \approx E\{\boldsymbol{D}_v(n)\}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha^*}. \tag{51}$$

Expected value $\boldsymbol{P}_4$

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

$$[\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)]_{ij} = x(n-i+1)\,v_i(n)\,v_j(n)\,x(n-j+1). \tag{52}$$

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 entry-wise product.

Expected value $\boldsymbol{P}_5$

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

$$E\{\boldsymbol{v}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} = E\{\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{D}_v(n)\}. \tag{54}$$

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

$$E\{\boldsymbol{v}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} \approx E\{\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^{\top}(n)\}\,E\{\boldsymbol{D}_v(n)\}. \tag{55}$$

Finally, using again **A2** we obtain

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

Expected value $\boldsymbol{P}_6$

Basic linear algebra gives

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

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

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

Expected value $\boldsymbol{P}_7$

Using basic algebra, **A2** and **A3** as done to obtain (55), we have

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

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

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

Expected value $\boldsymbol{P}_8$

Using basic algebra we obtain

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

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

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

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, equations (7)–(9)], using results from [25]. Following the same procedure as in [24] yields

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

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

$$E\{(\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n))^2\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\} \approx \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x + 2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x. \tag{64}$$

Then we obtain the final result

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

Expected value $\boldsymbol{P}_9$

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

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

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

$$E\{x(n-i+1)\,x(n-k+1)\,x(n-\ell+1)\,x(n-j+1)\} = r_x(k-i)\,r_x(j-\ell)+r_x(\ell-i)\,r_x(j-k)+r_x(j-i)\,r_x(\ell-k). \tag{67}$$

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 first note that

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

Now, writing $v_i(n+1)\,v_j(n+1) = \big(v_i(n) + \eta\,\Delta v_i(n)\big)\,\big(v_j(n) + \eta\,\Delta v_j(n)\big)$, 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 finally 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

$$E\{v_i(n)\,v_k(n)\,v_\ell(n)\,v_j(n)\} \approx E\{v_i(n)\,v_j(n)\}\,E\{v_k(n)\,v_\ell(n)\}. \tag{69}$$

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

$$\begin{aligned}[\boldsymbol{P}_9]_{ij} \approx\; & r_x(j-i) \sum_\ell \sum_k r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_\ell \sum_k r_x(k-i)\,r_x(j-\ell)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_\ell \sum_k r_x(\ell-i)\,r_x(j-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij}.\end{aligned} \tag{70}$$

The first right-hand term of Equation (70) can be expressed as follows

$$\begin{aligned} r_x(j-i)\sum_{\ell}\sum_{k} r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} &= [\boldsymbol{R}_x]_{ij}\sum_{k}(\sum_{\ell}[\boldsymbol{R}_x]_{k\ell}[\boldsymbol{K}(n)]_{k\ell})[\boldsymbol{K}(n)]_{ij} \\ &= [\mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x]_{ij}\,[\boldsymbol{K}(n)\}]_{ij}. \end{aligned} \tag{71}$$

The second and third right-hand terms write

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

This leads to the following close-form expression

$$[\boldsymbol{P}_9] = (\mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x + 2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x)\circ\boldsymbol{K}(n). \tag{73}$$

Using the expected values $\boldsymbol{P}_1$ to $\boldsymbol{P}_9$ in Equation (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 non-negative LMS algorithm behavior.

B. Simulation examples for the second-order moment analysis

This section presents simulation examples to verify the accuracy of the model (39). Figures 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 conducted in Section III-C. The simulation curves (blue line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (red line) were obtained from model (39). Note the model's accuracy even for step sizes as large as $\eta_{\max}/2$ (left side of Figure 6). Also note that the theoretical value of the minimum excess mean-square error $J_{emse_{min}}$ is represented in Figure 5.[5] It can be observed that $J_{emse}(n)$ tends to $J_{emse_{min}}$ as n goes to infinity. Figure 7 highlights the performance of the model for uncorrelated and correlated input signals $x(n)$ through the same experimental setup as 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 non-negative LMS algorithm in practical applications.

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

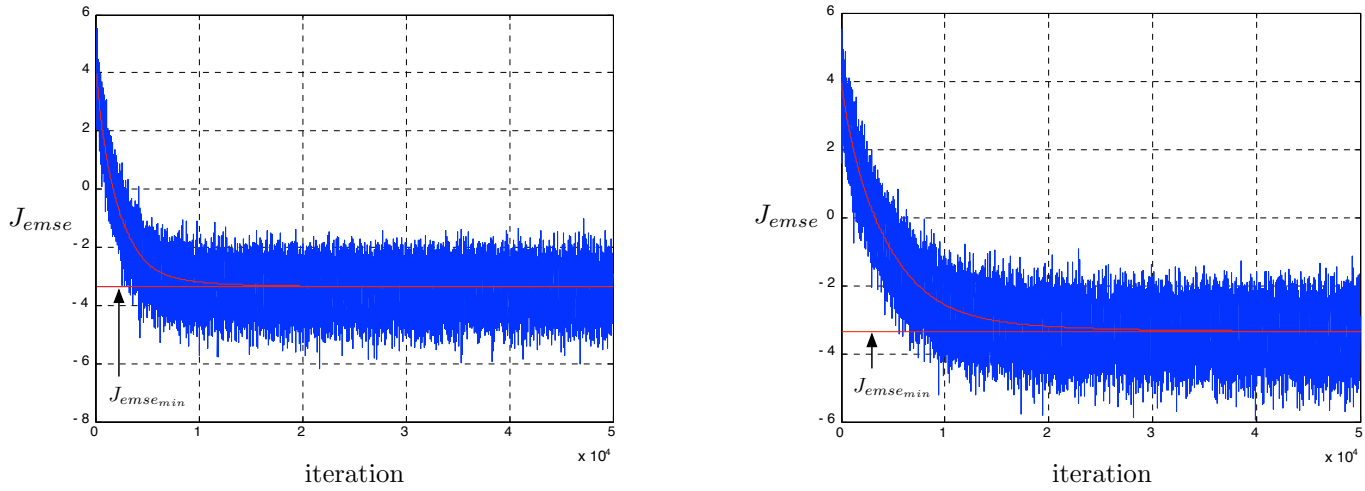

Fig. 5. Convergence of $J_{emse}(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered: $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure. The theoretical curves (red line) obtained from (38) and the model (39) and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

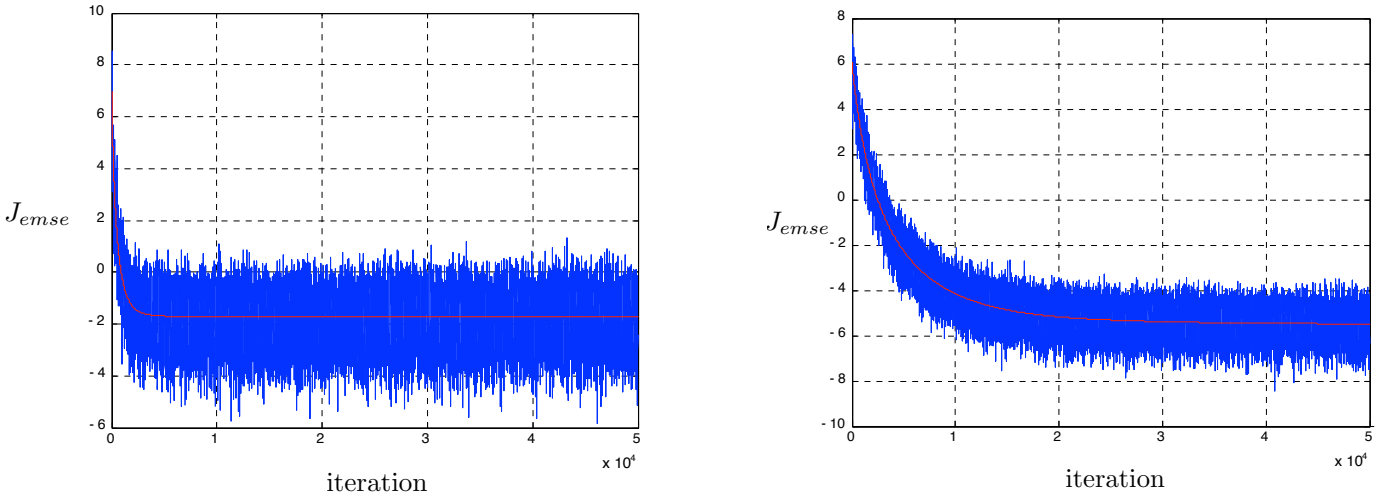

Fig. 6. Same experiment as in Figure 5 except that input sequence $x(n)$ is generated by a first-order AR process. Two different step sizes are considered: $\eta = 2.5 \times 10^{-3}$ on the left figure, and $\eta = 5 \times 10^{-4}$ on the right figure.

V. Conclusion

In many real-life phenomena, due to the inherent physical characteristics of systems under investigation, non-negativity 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 non-negativity constraints, and we derived the so-called non-negative LMS based on stochastic gradient descent. This algorithm switches automatically between a gradient descent mechanism and a gradient ascent one depending whether the non-negativity 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.

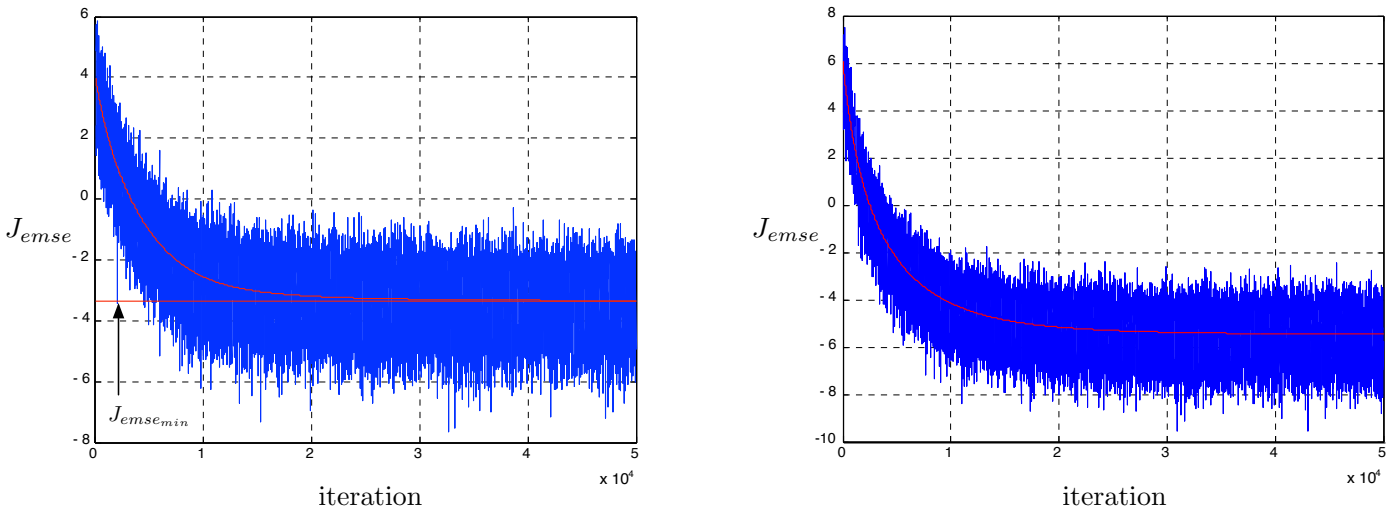

Fig. 7. Convergence of $J_{emse}(n)$ with step size $\eta = 5 \times 10^{-4}$, in the case where input $x(n)$ is i.i.d. on the left figure, and generated by a first-order AR process on the right figure. Compared to Figure 5 (right) and 6 (right), the variance of the noise $z(n)$ has been increased from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

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," *Journal of Chemometrics*, 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," *Advances in neural information processing systems, NIPS*, pp. 556–562, 2001.

[5] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations*, 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," *Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[7] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," *IEEE Transactions on Neural Networks*, 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 Transactions on Signal Processing*, vol. 54, no. 11, pp. 4133–4145, 2006.

[9] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics, 1995.

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

[11] J. B. Rosen, "The gradient projection method for nonlinear programming. part 1: Linear constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 181–217, 1960.

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

[13] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, 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 Processing Magazine*, vol. 28, no. 1, pp. 97 –123, January 2011.

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

[16] 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, 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 Processing*, vol. 81, no. 5, pp. 945–974, 2001.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, University Press, Cambridge, 2004.

[19] Ali Sayed, *Adaptive Filters*, Wiley-Interscience, New York, 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*, Springer-Verlag, 1997.

[22] Daniel Perrin, "La suite logistique et le chaos," Tech. Rep., Département de Mathématiques d'Orsay, Université de Paris-Sud, France, 2008.

[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, pp. 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 Transactions on Signal Processing*, vol. 51, no. 3, pp. 664 – 671, March 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 Transactions on Signal Processing*, vol. 47, no. 3, pp. 675 –689, March 1999.

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