NON-STATIONARY ANALYSIS OF THE CONVERGENCE OF THE NON-NEGATIVE LEAST-MEAN-SQUARE ALGORITHM

Jie Chen[⋆], *Cédric Richard*[⋆], *Jose-Carlos M. Bermudez*[‡], *Paul Honeine*[†]

[⋆] Université de Nice Sophia-Antipolis, CNRS, Observatoire de la Côte d'Azur, France
[†] Université de Technologie de Troyes, CNRS, France
[‡] Federal University of Santa Catarina, Florianópolis, Brazil
E-mail: {jie.chen,cedric.richard}@unice.fr, j.bermudez@ieee.org, paul.honeine@utt.fr

ABSTRACT

Non-negativity is a widely used constraint in parameter estimation procedures due to physical characteristics of systems under investigation. In this paper, we consider an LMS-type algorithm for system identification subject to non-negativity constraints, called Non-Negative Least-Mean-Square algorithm, and its normalized variant. An important contribution of this paper is that we study the stochastic behavior of these algorithms in a non-stationary environment, where the unconstrained solution is characterized by a time-variant mean and is affected by random perturbations. Convergence analysis of these algorithms in a stationary environment can be viewed as a particular case of the convergence model derived in this paper. Simulation results are presented to illustrate the performance of the algorithm and the accuracy of the derived models.

Index Terms— Non-negativity constraint, adaptive filtering, non-stationary signal, convergence analysis

1. INTRODUCTION

Optimizing a cost function given a set of constraints is a common objective in parameter estimation problems. The constraints are usually imposed by system specifications which provide *a priori* information on the feasible set of solutions. Estimating parameters subject to constraints poses specific problems in online applications. Non-negativity is one of the most commonly stated constraints. It is often imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. For instance, non-negativity constraints were used for image deblurring [1], deconvolution of system impulse response estimation [2] and audio processing [3]. Another similar problem is the non-negative matrix factorization (NMF), which is now a popular dimension reduction technique [4, 5, 6]. This problem is closely related to blind deconvolution, and has found direct application in neuroscience [7] and hyperspectral imaging [8]. Separation of non-negative mixture of non-negative sources has also been considered in [9, 10].

Over the last fifteen years, a variety of methods have been developed to tackle non-negative least-squares (NNLS) problems. Active set techniques use the fact that if the set of variables which activate constraints is known, then the solution of the constrained least-squares problem can be obtained by solving an unconstrained one that includes only inactive variables. The active set algorithm of Lawson and Hanson [11] is a batch resolution technique for NNLS problems. It has become a standard among the most frequently used methods. In [12], 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. Projected gradient algorithms [13, 14, 15, 16] form another class, which is based on successive projections onto the feasible region. In [17], Lin used this kind of algorithm for NMF problems. Low memory requirements and simplicity make these algorithms 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 [5, 18]. Particularly efficient updates were derived in this way for a large number of problems involving non-negativity constraints [19]. However, these algorithms require batch processing, which is not suitable for online system identification problems.

An LMS-type adaptive algorithm called Non-Negative Least Mean Square (NNLMS), which is suitable for online applications, has been proposed in [20]. It is based on stochastic gradient descent approach combined with a fixed point iteration scheme. The NNLMS algorithm was studied in [20] for operation in stationary environments, and was shown to converge to a solution satisfying the Karush-Kuhn-Tucker conditions. In this paper we study the stochastic behavior of the NNLMS algorithm and its normalized counterpart for operation in a nonstationary environment, where the unconstrained solution is characterized by a time-variant mean and is affected by random perturbations. Convergence analysis of these algorithms in a stationary environment can be viewed as a particular case of the convergence model derived in this paper.

2. NON-NEGATIVE LEAST-MEAN-SQUARE ALGORITHM

Consider the estimation problem where the unknown system is characterized by real-valued observations

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

where $(\cdot)^\top$ denotes the vector transpose, $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*]^\top$ is the vector of model parameters, and $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^\top$ is the input data vector. The input signal $x(n)$ and the additive noise $z(n)$ are assumed stationary and zero-mean.

In certain applications, inherent physical characteristics impose a non-negativity constraint on the estimate $\boldsymbol{\alpha}$. Therefore, the problem of identifying the optimum non-negative model can be formalized as follows

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

where $J(\boldsymbol{\alpha})$ is a continuously differentiable and strictly convex cost function in $\mathbb{R}^N$, and $\boldsymbol{\alpha}^{\text{o}}$ is the solution to the constrained optimization problem. To solve the problem (2), let us consider its Lagrangian function $Q(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ given by [21]

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

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 $\boldsymbol{\alpha}^{\text{o}}$, $\boldsymbol{\lambda}^{\text{o}}$, namely,

$$\nabla_\alpha Q(\boldsymbol{\alpha}^{\text{o}}, \boldsymbol{\lambda}^{\text{o}}) = 0 \tag{3}$$

$$\alpha_i^{\text{o}} \, [\boldsymbol{\lambda}^{\text{o}}]_i = 0, \quad i = 1, \ldots, N \tag{4}$$

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

$$\alpha_i^{\text{o}} \, [-\nabla_\alpha J(\boldsymbol{\alpha}^{\text{o}})]_i = 0, \quad i = 1, \ldots, N \tag{5}$$

where the extra minus sign is just used to make a gradient descent of $J(\boldsymbol{\alpha})$ apparent. Equations of the form $\varphi(u) = 0$ can be solved with a fixed-point iteration algorithm by considering the problem $u = u + \varphi(u)$ under some conditions on function φ. Implementing this strategy with equation (5) leads to the component-wise gradient descent algorithm

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

with $\eta_i(n)$ a positive step size required to get a contraction scheme and to control the convergence rate.

Consider the criterion $J_{\text{MSE}}(\boldsymbol{\alpha}) = E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\}$ to be minimized with respect to $\boldsymbol{\alpha}$ so that

$$\begin{aligned} \boldsymbol{\alpha}^o = \; & \arg\min_{\boldsymbol{\alpha}} E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\} \\ & \text{subject to } \alpha_i \geq 0, \quad \forall i \end{aligned} \tag{7}$$

The gradient of $J_{\text{MSE}}(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ is $\nabla_\alpha J_{\text{MSE}}(\boldsymbol{\alpha}) = 2\,(\boldsymbol{R}_x \boldsymbol{\alpha} - \boldsymbol{r}_{xy})$, where $\boldsymbol{R}_x$ is the autocorrelation matrix of $\boldsymbol{x}(n)$, and $\boldsymbol{r}_{xy}$ is the correlation vector between $\boldsymbol{x}(n)$ and $y(n)$. Following a stochastic gradient approach, the second-order moments $\boldsymbol{R}_x$ and $\boldsymbol{r}_{xy}$ are replaced in (6) by the instantaneous estimates $\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)$ and $y(n)\,\boldsymbol{x}(n)$, respectively. Choosing a positive fixed step size η, defining $\boldsymbol{D}_{\boldsymbol{\alpha}}(n)$ as the diagonal matrix with diagonal entries given by $\boldsymbol{\alpha}(n)$, leads to the stochastic update given by

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta \, \boldsymbol{D}_{\boldsymbol{\alpha}}(n) \, \boldsymbol{x}(n) \, e(n) \tag{8}$$

where the estimation error $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\,\boldsymbol{x}(n)$, and the algorithm is required to be initialized with all $\alpha_i(0) > 0$. We refer to this method as the Non-Negative LMS algorithm (NNLMS).

Compared with the classical LMS algorithm, the direction of the (instantaneous) gradient vector in (8) is modified by the premultiplication by $\eta \, \boldsymbol{D}_\alpha(n)$, i.e., the ith component is scaled by $\eta \, \alpha_i(n)$. The weight vector update is then no longer in the direction of the gradient, as a consequence of the constraints. This effect enables the weight corrections to reduce gradually to zero for weights approaching zero from the positive phase. These weights tend to converge without becoming negative. If a weight that approaches zero turns negative due to the stochastic update, the instant multiplicative weight $\alpha_i(n)$ turns the update from the gradient descend strategy to a gradient ascend one, and the weigth tends to turn positive again. Hence, differently from LMS, the NNLMS correction term is a nonlinear function of $\boldsymbol{\alpha}(n)$. This causes the two algorithms to have completely different convergence behavrios

A direct extension of the NNLMS is the Normalized NNLMS. Conditionally to $\boldsymbol{\alpha}(n)$, the product $e(n)\boldsymbol{D}_x(n)$ in (8) is proportional to the power of the input signal. Hence, setting a constant value for the step size η leads to different weight updates for different signal power levels. This is the same sensitivity to signal power verified in the LMS algorithm. A popular way to address this limitation is to normalize the weight update by the input vector squared ℓ_2-norm which yields the Normalized NNLMS update equation

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n) + \epsilon} \boldsymbol{D}_{\boldsymbol{\alpha}}(n)\,\boldsymbol{\alpha}(n)\,e(n) \tag{9}$$

A small positive regularization parameter ϵ is added to the denominator $\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)$ to avoid numerical difficulties.

3. STOCHASTIC BEHAVIOR STUDY FOR NON-STATIONARY ENVIRONMENTS

The nonlinearity of the weight correction term with respect to the estimated weight vector makes the theoretical analysis of the NNLMS algorithm quite different from that of LMS. Models for the behavior of the algorithm in stationary environments have been studied in [20]. We now study the stochastic behavior of the NNLMS and its normalized variant with fixed step sizes for a non-stationary environment. To this end, instead of considering a constant system weight vector $\boldsymbol{\alpha}^*$ in (1), the system is characterized by a time variant weight vector $\boldsymbol{\alpha}^*(n)$ given by

$$\boldsymbol{\alpha}^*(n) = \boldsymbol{\alpha}_{\text{o}}^*(n) + \boldsymbol{\xi}(n) \tag{10}$$

with $\boldsymbol{\alpha}_{\text{o}}^*(n)$ a deterministic time-variant mean trajectory, and $\boldsymbol{\xi}(n)$ a zero-mean random variable with covariance matrix $\boldsymbol{\Xi} = \sigma_\xi^2 \boldsymbol{I}$ that is independent of any other signal. The deterministic trajectory may result from inherent properties of the system, such as cosine or circular behavior [22]. This model leads to a tractable analysis and permits inferences about the behavior of the algorithms in time variant environments, by varying the trajectory $\boldsymbol{\alpha}_{\text{o}}^*(n)$ and the power σ_ξ^2 of $\boldsymbol{\xi}(n)$. For the analyses that follow, we shall define the weight error vector with respect to the mean unconstrained solution $\boldsymbol{\alpha}_{\text{o}}^*(n)$ as

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}_{\text{o}}^*(n). \tag{11}$$

The following analysis is performed for $x(n)$ and $z(n)$ zero-mean stationary Gaussian, and for $z(n)$ white and statistically independent of any other signal. We assume in the subsequent mean weight behavior analysis that the input and weight vectors are statistically independent, according to the well-known Independence Assumption [23]. We start by studying the behavior of the NNLMS in non-stationary environments. The result will be extended to its normalized variant at the end of this section.

3.1. Mean weight behavior analysis

Considering that the estimation error can also be expressed by $e(n) = z(n) - (\boldsymbol{v}(n) - \boldsymbol{\xi}(n))^\top \boldsymbol{x}(n)$, and using the relation (11) in (9) the weight error update equation of NNLMS can be written as

$$\begin{aligned} \boldsymbol{v}(n+1) &= \boldsymbol{v}(n) + \eta z(n) \boldsymbol{D}_x(n) \boldsymbol{v}(n) \\ &\quad + \eta z(n) \boldsymbol{D}_x(n) \boldsymbol{\alpha}_{\text{o}}^*(n) - \eta \boldsymbol{D}_x(n) \boldsymbol{\alpha}_{\text{o}}^*(n) \boldsymbol{x}^\top(n) \boldsymbol{v}(n) \\ &\quad - \eta \boldsymbol{D}_x(n) \boldsymbol{v}(n) \boldsymbol{v}^\top(n) \boldsymbol{x}(n) + \eta \boldsymbol{D}_x(n) \boldsymbol{v}(n) \boldsymbol{\xi}^\top(n) \boldsymbol{x}(n) \\ &\quad + \eta \boldsymbol{D}_x(n) \boldsymbol{\alpha}_{\text{o}}^*(n) \boldsymbol{\xi}^\top(n) \boldsymbol{x}(n) - \boldsymbol{\Delta}(n). \end{aligned} \tag{12}$$

where $\boldsymbol{D}_x(n)$ is the diagonal matrix with $\boldsymbol{x}(n)$ as diagonal entries, and $\boldsymbol{\Delta}(n) = \boldsymbol{\alpha}_{\text{o}}^*(n+1) - \boldsymbol{\alpha}_{\text{o}}^*(n)$ is a deterministic vector proportional to the derivative of the mean unconstrained optimal solution.

Taking the expected value of (12) and noting that the expectations of the second, third, sixth and seventh terms on the r.h.s. are equal to zero by virtue of the natures of $z(n)$ and $\boldsymbol{\xi}(n)$ yields

$$E\{\boldsymbol{v}(n+1)\} = E\{\boldsymbol{v}(n)\} - \eta E\Big\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}_{\rm o}^*(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\Big\} - \eta E\Big\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\Big\} - \boldsymbol{\Delta}(n). \tag{13}$$

where $E\{\cdot\}$ denotes the expected value. Using the independence assumption, the second expectation in the r.h.s. of (13) can be written as

$$E\Big\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}_{\rm o}^*(n)\boldsymbol{x}^\top(n)\boldsymbol{v}(n)\Big\} = \boldsymbol{D}_{\alpha_{\rm o}^*}(n)\boldsymbol{R}_x E\{\boldsymbol{v}(n)\} \tag{14}$$

The third term is given by

$$E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\} = \operatorname{diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} \tag{15}$$

with $\boldsymbol{K}(n)$ the correlation matrix of the weight error $\boldsymbol{K}(n) = E\{\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\}$. Hence the mean behavior model (13) is expressed by

$$\begin{aligned} E\{\boldsymbol{v}(n+1)\} &= \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\,\boldsymbol{R}_x\big)\;E\{\boldsymbol{v}(n)\} \\ &\quad - \eta\operatorname{diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} - \boldsymbol{\Delta}(n) \end{aligned} \tag{16}$$

This recursion in the variable $E\{\boldsymbol{v}(n)\}$ requires a model for $\boldsymbol{K}(n)$. A recursive model will be derived for $\boldsymbol{K}(n)$ in the later subsection. We have observed that a sufficiently accurate and more intuitive mean behavior model can be obtained by neglecting the weight error fluctuation terms and by using the following separation approximation $\boldsymbol{K}(n) \approx E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}$. A discussion about the validity of this approximation can be found in [20]. We thus obtain the following first-order model

$$\begin{aligned} E\{\boldsymbol{v}(n+1)\} &= \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} \\ &\quad - \eta\operatorname{diag}\Big\{\boldsymbol{R}_x\,E\{\boldsymbol{v}(n)\}E\{\boldsymbol{v}^\top(n)\}\Big\} - \boldsymbol{\Delta}(n) \end{aligned} \tag{17}$$

3.2. Second-order moment analysis

The excess means square estimation error (EMSE) is given by $J_{\rm EMSE}(n) = E\left\{(\boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*(n))^\top\boldsymbol{x}(n)\boldsymbol{x}^\top(n)(\boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*(n))\right\}$. Using (10)-(11), the properties of $\boldsymbol{\xi}(n)$, we can write $J_{\rm EMSE}(n)$ as

$$\begin{aligned} J_{\rm EMSE}(n) &= E\Big\{(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))^\top\boldsymbol{x}(n)\boldsymbol{x}^\top(n)(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))\Big\} \\ &= \operatorname{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} + \operatorname{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\} \end{aligned} \tag{18}$$

The term $\operatorname{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\}$ is the direct contribution of the random nonstationarity of the system to the EMSE. In order to estimate the EMSE, we need to derive a recursive model for $\boldsymbol{K}(n)$. The same statistic assumptions A1–A4 as in [20] are used in the following derivation.

Post-multiplying (12) by its transpose, taking the expectation, it can be observed that

$$\boldsymbol{K}(n+1) = \boldsymbol{K}(n) + \overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1) + \boldsymbol{K}_{\boldsymbol{\xi}}(n+1) + \boldsymbol{K}_{\boldsymbol{\Delta}}(n+1) \tag{19}$$

Matrix $\overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1)$ consists of the expectations of products between the pairs of the first to fifth terms in (12), and is not affected by the random fluctuation $\boldsymbol{\xi}(n)$. Matrix $\boldsymbol{K}_{\boldsymbol{\xi}}(n+1)$ consists of cross-products where $\boldsymbol{\xi}(n)$ is involved. The last term $\boldsymbol{K}_{\boldsymbol{\Delta}}(n)$ conveys the effect of deterministic variation of the mean of system weights.

Decoupled with the random perturbation, matrix $\overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1)$ evolution has the same form as the recursive equation of the weight error covariance matrix derived in [20] for the case of a stationary environment, except that the system weight $\boldsymbol{\alpha}^*$ has to be replaced here by its time variant counterpart $\boldsymbol{\alpha}_o^*(n)$. This leads to the following expression for $\overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1)$

$$\begin{aligned} \overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1) &= \eta^2(\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\,\boldsymbol{Q}(n)\,\boldsymbol{D}_{\alpha_{\rm o}^*}(n) + \boldsymbol{Q}(n)\circ\boldsymbol{K}(n)) \\ &\quad + \eta\sigma_z^2(\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\boldsymbol{R}_x\boldsymbol{D}_{\alpha_{\rm o}^*}(n) + \boldsymbol{R}_x\circ\boldsymbol{K}(n)) + (\boldsymbol{P}_1(n) + \boldsymbol{P}_1^\top(n)) \end{aligned}$$

with $\boldsymbol{Q}(n) = 2\boldsymbol{R}_x\boldsymbol{K}(n)\boldsymbol{R}_x + \operatorname{trace}\{\boldsymbol{R}_x\boldsymbol{K}(n)\}\boldsymbol{R}_x$, and

$$\begin{aligned} \boldsymbol{P}_1(n) &= -\eta\,(\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\,\boldsymbol{R}_x\boldsymbol{K}(n) + \boldsymbol{K}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_v(n)\}) \\ &\quad + \eta^2\boldsymbol{D}_{\alpha_{\rm o}^*}(n)\,\boldsymbol{Q}(n)\,E\{\boldsymbol{D}_v(n)\} + \eta^2\sigma_z^2 E\{\boldsymbol{D}_v(n)\}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha_{\rm o}^*}(n), \end{aligned}$$

where $\circ$ denotes the so-called Hadamard entry-wise product. Notice that the cross-products of the sixth and seventh terms of (12) with the other terms lead to zero mean values. This leaves only auto-product terms

$$\boldsymbol{K}_{\boldsymbol{\xi}}(n+1) = \eta^2\,(\boldsymbol{P}_2(n) + \boldsymbol{P}_3(n) + \boldsymbol{P}_4(n) + \boldsymbol{P}_4^\top(n)) \tag{20}$$

with

$$\begin{aligned} \boldsymbol{P}_2(n) &= E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{D}_x(n)\} \\ \boldsymbol{P}_3(n) &= E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{\alpha}_{\rm o}^*(n)\boldsymbol{\alpha}_{\rm o}^{*\top}(n)\boldsymbol{D}_x(n)\} \\ \boldsymbol{P}_4(n) &= E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{\alpha}_{\rm o}^{*\top}(n)\boldsymbol{D}_x(n)\} \end{aligned}$$

These terms convey the effect of the random part of the environment non-stationarity. Computing the (i,j)-th entry of $\boldsymbol{P}_2(n)$ leads to

$$\begin{aligned} [\boldsymbol{P}_2]_{ij}(n) = \sum_k\sum_l & E\{\xi_k(n)\xi_l(n)\}\;E\{v_i(n)v_j(n)\} \\ & \cdot E\{x_k(n)x_l(n)x_i(n)x_j(n)\} \end{aligned} \tag{21}$$

As $E\{\xi_k(n)\xi_l(n)\} \neq 0$ only for $k = l$, using the Gaussian moment factorizing theorem for the term with fourth-order statistics of $x(n)$, yields $\sum_k E\left\{x_k^2(n)x_i(n)x_j(n)\right\} = \big[\boldsymbol{R}_x\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x\big]_{ij}$. We can thus write the result in matrix form

$$\boldsymbol{P}_2(n) = \sigma_\xi^2\,\boldsymbol{K}(n)\circ(\boldsymbol{R}_x\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \tag{22}$$

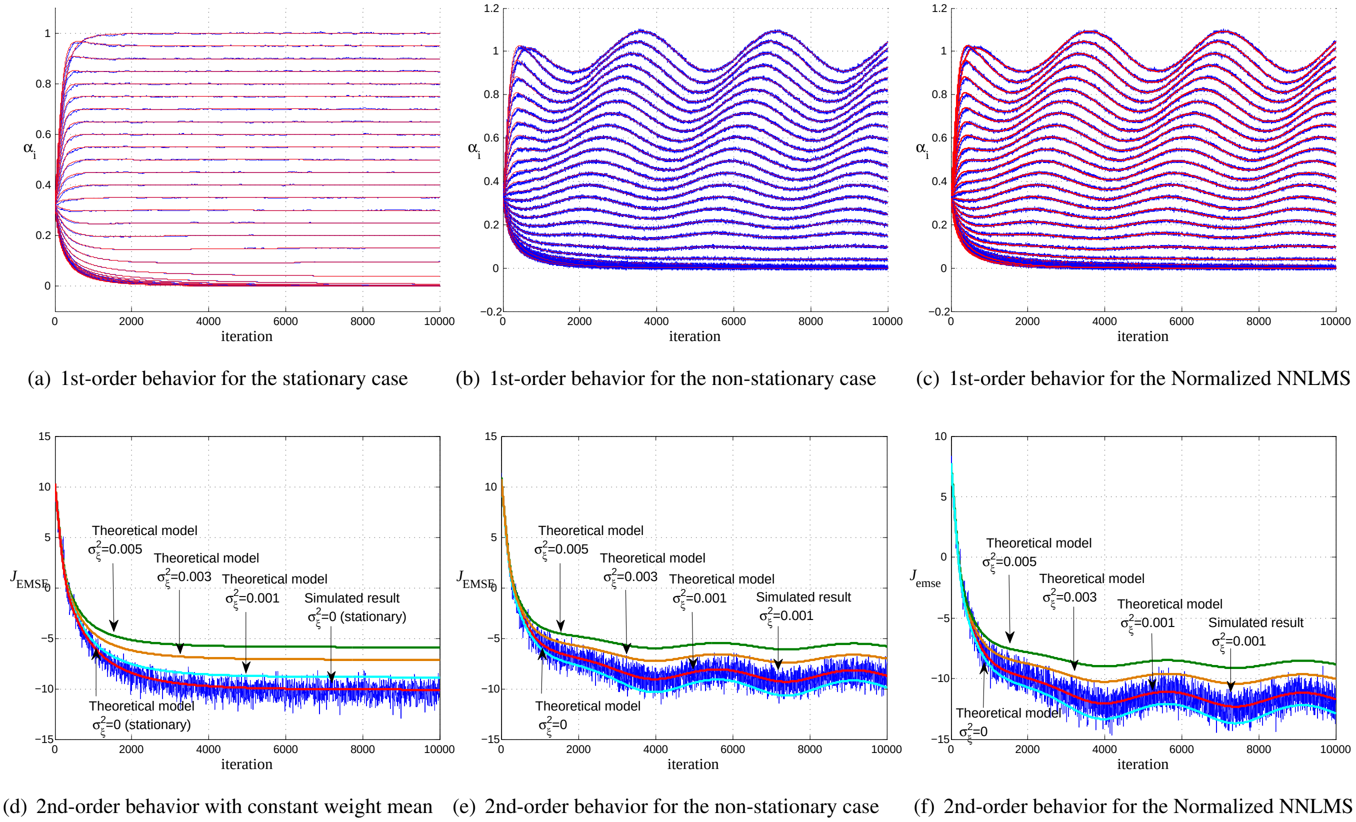
Similarly, we have the expected values for $\boldsymbol{P}_3(n)$ and $\boldsymbol{P}_4(n)$

$$\begin{aligned} \boldsymbol{P}_3(n) &= \sigma_\xi^2\,\Big(\boldsymbol{\alpha}_{\rm o}^*(n)\,\boldsymbol{\alpha}_{\rm o}^{*\top}(n)\Big)\circ(\boldsymbol{R}_x\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \\ \boldsymbol{P}_4(n) &= \sigma_\xi^2\,\Big(E\{\boldsymbol{v}_{\rm N}(n)\}\,\boldsymbol{\alpha}_{\rm o}^{*\top}(n)\Big)\circ(\boldsymbol{R}_x\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \end{aligned}$$

Finally, it can be easily observed that the last term in (19) writes

$$\begin{aligned} \boldsymbol{K}_{\boldsymbol{\Delta}}(n+1) &= \boldsymbol{\Delta}(n)\boldsymbol{\Delta}^\top(n) - \boldsymbol{\Delta}(n)\big(E\{\boldsymbol{v}(n+1)\} + \boldsymbol{\Delta}(n)\big)^\top \\ &\quad - \big(E\{\boldsymbol{v}(n+1)\} + \boldsymbol{\Delta}(n)\big)\boldsymbol{\Delta}^\top(n) \end{aligned} \tag{23}$$

With these closed form expressions for $\overline{\boldsymbol{K}}_{\boldsymbol{\xi}}(n+1)$, $\boldsymbol{K}_{\boldsymbol{\xi}}(n+1)$, and $\boldsymbol{K}_{\boldsymbol{\Delta}}(n+1)$, we can characterize the transient behavior of EMSE via (18). Taking $n \to \infty$ the steady state performance can also be studied.



(a) 1st-order behavior for the stationary case

(b) 1st-order behavior for the non-stationary case

(c) 1st-order behavior for the Normalized NNLMS

(d) 2nd-order behavior with constant weight mean

(e) 2nd-order behavior for the non-stationary case

(f) 2nd-order behavior for the Normalized NNLMS

Fig. 1. Evolution of system coefficients and excess mean square error. First column: NNLMS in a stationary environment (with additional theoretical curves as a function of $\boldsymbol{\xi}(n)$ in the EMSE figure). Second column: NNLMS in non-stationary environments. Third column: Behavior of Normalized NNLMS in non-stationary environments.

3.3. Models for Normalized NNLMS

Evaluation of the expected values of the first and second-order update equations for the Normalized NNLMS (9) involves terms containing the denominator $\boldsymbol{x}^\top(n)\boldsymbol{x}(n)+\epsilon$. A common approximation that works well for reasonably large N is to neglect the correlation between this term and the others, as it tends to vary much slower [24, 25]. Given this slow variation and ϵ usually very small, we approximate $\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon$ by $N\sigma_x^2$, which is also reasonable for large values of N. Therefore, all the models that have been derived above can be used for Normalized NNLMS by replacing the step size η by the equivalent step size $\tilde{\eta}=\eta\,N\sigma_x^2$.

4. SIMULATION EXAMPLES

In this section, we present simulation examples to illustrate the properties of the algorithms and the accuracy of the derived models. In these examples, the system order is $N=31$. The unknown stationary system is defined by

$$\alpha_{o_i}^{*\,(\text{st.})}=\begin{cases}1-0.05\,i & i=0,\dots,20\\ -0.01\,(i-18) & i=21,\dots,30\end{cases} \tag{24}$$

The last ten negative coefficients are used to activate the non-negativity constraint. For the non-stationary case, we consider an unknown response defined by

$$\alpha_{o_i}^{*\,(\text{nst.})}(n)=\alpha_{o_i}^{*\,(\text{st.})}+\frac{|\alpha_{o_i}^{*\,(\text{st.})}|}{10}\cos\left(\frac{2\pi n}{T}+\frac{2\pi(i-1)}{N}\right)+\xi_i(n)$$

where the period T of the deterministic sinusoidal component is set to 3500, and $\boldsymbol{\xi}(n)$ is a zero-mean Gaussian random vector with correlation matrix $\sigma_\xi^2\boldsymbol{I}$ and $\sigma_\xi^2=0.001$. The input $x(n)$ is a correlated signal generated by a first-order AR process defined as follows: $x(n)=0.5\,x(n-1)+w(n)$, with $w(n)$ an i.i.d. zero-mean Gaussian variable. The variance σ_w^2 was set to $1-0.5^2$ so that $\sigma_x^2=1$. The noise $z(n)$ is zero-mean i.i.d. Gaussian with variance $\sigma_z^2=10^{-2}$. The adaptive weights in $\alpha_i(0)$ were all initialized at $10/N$ for all the realizations. Monte Carlo simulation results were obtained by averaging 100 runs.

Firstly, we ran the NNLMS algorithm and the theoretical models for both stationary and non-stationary environments with the step size $\eta=0.005$. In Figs. 1(a), 1(b), 1(d), 1(e), blue curves show simulation results and red curves show the theoretical predictions. It can be verified that the models accurately predict the algorithm behavior. In Figs. 1(a) and 1(b), all the coefficients satisfy the non-negativity constraint. Three more theoretical curves have been added to the second-order plots in Figs. 1(d) and 1(e) to illustrate the effect of the random parameter σ_ξ^2. These curves illustrate the expected extra EMSE due to tracking of the variations of the random optimal solution. To preserve visibility of the graph, simulation curves are not presented but it should be noticed that they coincide with the theoretical ones.

Secondly, the Normalized NNLMS algorithm was tested to highlight its properties. Only the results obtained in non-stationary environments are given here. The step size $\tilde{\eta}$ of the normalized algorithm was set to $\tilde{\eta} = \eta N \sigma_x^2 = 0.005 \times 30 \times 1 = 0.45$ in order to make its performance comparable with that of the NNLMS algorithm depicted in Figs. 1(b) and 1(e), where $\sigma_x^2 = 1$. For the Normalized NNLMS, variance σ_w^2 was set in order that $\sigma_x^2 = 0.5$. The blue curves show simulation results and red curves represent the theoretical predictions. It can be verified that the proposed models accurately predict the algorithm behavior. Moreover, comparison between Figs. 1(b), 1(e) and Figs. 1(c), 1(f) shows that normalization allows the algorithm to basically converge in the same manner, independently of the input power. As previously, three theoretical curves have been added to Fig. 1(f) in order to illustrate the effect of random perturbations on Normalized NNLMS.

5. CONCLUSION AND PERSPECTIVE

In this paper, we made further inspection into the NNLMS algorithm to address the online system identification problem under nonnegativity constraints. The main contribution concerns the derivation of the first-order and the second-order stochastic behavior models for non-stationary environments. These models include as a particular case the algorithm behavior in a stationary environment [20]. The accuracy of these models will enable us to use them as important design tools to estimate the transient, steady-state and tracking performances of the proposed algorithms.

6. 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, Feb. 2010.

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

[3] A. Cont and S. Dubinov, "Realtime multiple pitch and multiple-instrument recognition for music signals unsing sparse non-negative constraints," in *Proc. of the 10th Intl. Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sep. 2007.

[4] 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, Oct. 1999.

[5] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, Apr. 2001.

[6] J. Mairal, F. Bach, J. Ponce, and G. Saphiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, , no. 11, pp. 19–60, Jan. 2010.

[7] A. Cichocki, R. Zdunek, and A.H. Phan, *Nonnegative matrix and tensor factorizations: applications to exploratory multiway data analysis and blind source separation*, Wiley, 2009.

[8] 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, Sep. 2007.

[9] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 534–543, Mar. 2003.

[10] S. Moussaoui, D. Brie, A. Mohammad-Djafari, and C. Carteret, "Separation of non-negative mixture of nonnegative sources using a bayesian approach and MCMC sampling," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4133–4145, Nov. 2006.

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

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

[13] 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, Mar. 1960.

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

[15] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, Jan. 1988.

[16] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97 –123, Jan. 2011.

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

[18] 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, Nov. 2007.

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

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

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

[22] X. Zhao, S-Y Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and nonstationary data," *IEEE Transactions on Signal ProcessingI*, vol. 60, no. 7, pp. 3460–3475, 2012.

[23] Ali Sayed, *Adaptive Filters*, Wiley-Interscience, New York, 2008.

[24] C. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithms," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 31, no. 10, pp. 1177–1191, Oct. 1983.

[25] S. J. M Almeida, J. C. M. Bermudez, and N. J. Bershad, "A statistical analysis of the affine projection algorithm for unity step size and autoregressive inputs," *IEEE Transactions on Circuits and Systems Part I: Fundamental Theory and Applications*, vol. 52, no. 7, pp. 1394–1405, Jul. 2005.