A modified non-negative LMS algorithm and its stochastic behavior analysis

Jie Chen(1,2), Cédric Richard(1), Jose Bermudez(3), Paul Honeine(2)
(1)Universit? de Nice Sophia-Antipolis, France
(2)Universit? de Technologie de Troyes, France
(3)Department of Electrical Engineering, Federal University of Santa Catarina, Florian?polis, SC, Brazil

***Abstract*—In hyperspectral images, pixels are mixtures of spectral components associated to pure materials. Although the linear mixture model is the most studied case, nonlinear models have been taken into consideration to overcome some limitations of the linear model. In this paper, nonlinear hyperspectral unmixing problem is studied through kernel-based learning theory. Endmember components at each band are mapped implicitly in a high feature space, in order to address the nonlinear interaction of photons. Experiment results with both synthetic and real images illustrate the effectiveness of the proposed scheme.**

I. Introduction

Optimizing an objective function subject to constraints is very common in statistical signal and image processing. Constraints represent prior knowledge that may lead to proper solutions, or may reduce the feasible space of solutions when the problem is ill-conditioned. Non-negativity is probably one of the most commonly used constraints. It is often imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. Non-negative least-square problems (NNLS) have been addressed in applications ranging from image deblurring [1] to impulse response estimation [2]. Non-negative matrix factorization (NMF) [3], which is closely related to blind deconvolution problems, have also found direct application in hyperspectral imaging [4].

A variety of methods have been proposed in the literature to tackle the NNLS problem. The Active set algorithm of Lawson and Hanson [5] is a batch resolution technique for NNLS problems, which has become a standard among the most frequently used methods. Projected gradient methods [6], [7] are based on successive projections on the feasible region. Multiplicative algorithms are very popular to solve NMF problems [8]. All these algorithms are however based on batch processing, which is not suitable for online system identification problems. In [9], [10], online system identification subject to non-negativity constraints on the parameters to estimate was addressed. A LMS-type algorithm, called non-negative LMS algorithm (NN-LMS), has been proposed to compute solutions adaptively. It relies on stochastic gradient descent combined with a fixed point iteration scheme to converge to a solution that satisfies to Karush-Kuhn-Tucker conditions.

During convergence of NN-LMS algorithm, it has been observed that weights may have different convergence rates and accuracy, especially those in the active set since they become smaller and finally tend to zero as iterations proceed. In this paper we introduce a modified NN-LMS algorithm in order to alleviate this problem. We also propose analytical models to characterize the stochastic behavior of this algorithm.

II. Modified NN-LMS algorithm

A. Presentation of the NN-LMS algorithm

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

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

with $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^\top$ the vector of 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. Sequence $z(n)$ represents measurement noise and modeling errors. Due to inherent physical characteristics of systems under investigation, here we consider the problem of identifying the optimum model defined by

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

with $J(\boldsymbol{\alpha}) = E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\}$, and $\boldsymbol{\alpha}^o$ the solution of the constrained optimization problem. The KKT conditions at the optimum $\boldsymbol{\alpha}^o$ can be combined into the following expression [9], [10]

$$\alpha_i^o \, [-\nabla_\alpha J(\boldsymbol{\alpha}^o)]_i = 0 \tag{3}$$

where the extra minus sign is just used to make a gradient descent of $J(\boldsymbol{\alpha})$ apparent. Equations of the form $g(u) = 0$ can be solved with a fixed-point algorithm, under some conditions on function g, by considering the problem $u = u + g(u)$. Implementing this fixed-point strategy with (3) leads us to the component-wise gradient descent algorithm

$$\alpha_i(k+1) = \alpha_i(k) + \eta\, \alpha_i(k)[-\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i \tag{4}$$

978-1-4673-0323-1/11/$26.00 ©2011 IEEE

where η is a positive step size that must be tuned to construct a contraction scheme and to control the convergence rate. Using usual stochastic gradient approximations as in LMS and rewriting the update equation in vectorial form, we obtain the NN-LMS algorithm [9], [10]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{D}_{\alpha}(n)\, \boldsymbol{x}(n) \tag{5}$$

where $\boldsymbol{D}_{\alpha}(n)$ stands for the diagonal matrix with diagonal entries given by $\boldsymbol{\alpha}(n)$, and $e(n) = y(n) - \boldsymbol{\alpha}^{\top}(n)\, \boldsymbol{x}(n)$. The direction of the gradient vector is modified by the premultiplication by $\boldsymbol{D}_{\alpha}(n)$, and the weight vector update is no longer in the direction of the gradient. This is a consequence of the constraints.

B. *Modified NN-LMS*

We note that in NN-LMS algorithm, each component of the gradient vector is scaled by a different value $\alpha_i(n)\eta$. However, these factors that modify the step size along each axis result in different convergence rates and accuracy for each component of $\boldsymbol{\alpha}(n)$, especially for the weights corresponding to the active set as they become smaller through iterations, and finally tend to zero. In order to compensate this unbalance of convergence rates, we now introduce the modified NN-LMS algorithm. Considering KKT condition (3), we can equivalently write

$$[\alpha_i^o]^{\gamma}\, [-\nabla_{\alpha} J(\boldsymbol{\alpha}^o)]_i = 0 \tag{6}$$

with γ a positive real number. Implementing the fixed-point iteration strategy with equation (6) and using instantaneous estimates for the gradient leads us to the following algorithm

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{D}_{x}(n)\, \boldsymbol{\alpha}^{\gamma}(n) \tag{7}$$

In the above equation, with abuse of notation, $\boldsymbol{\alpha}^{\gamma}(n)$ denotes the exponential value γ applied to each component $\alpha_i(n)$ of $\boldsymbol{\alpha}(n)$ as follows: $\alpha_i^{\gamma}(n) = sign\{\alpha_i(n)\}|\alpha_i(n)|^{\gamma}$. We recommend to specify parameter γ in the form $\gamma = \frac{p}{q}$ where p and q are both integer numbers, and $0 < p \leq q$. Although it can be shown that weights are always non-negative thanks to an appropriate step size, the fixed-point 0 can be reached by the left side 0^- with less strict step size selection. See [9] for more details with NN-LMS. Just as the gamma correction in image processing, an exponential value $0 < \gamma < 1$ reduces the dynamic range of parameter vector $\boldsymbol{\alpha}(n)$. Each component $\alpha_i(n)$ will be closer to 1 no matter it is larger or smaller than 1. When $\gamma = 1$, the update equation degenerates to the NN-LMS algorithm in the original form (5). A γ larger than 1 is not recommended in most cases, as it enlarges difference among the components.

III. Algorithm behavior modeling

We now propose models to characterize the stochastic behavior of the modified NN-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 the solution of the least-mean-square problem by $\boldsymbol{\alpha}^*$,

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} E\{[y(n) - \boldsymbol{\alpha}^{\top}\boldsymbol{x}(n)]^2\}. \tag{8}$$

It is assumed that $z(n)$ is stationary, zero-mean with variance σ_z^2 and statistically independent of any other signal. Thus, $E\{z(n)\, \boldsymbol{D}_x(n\} = 0$. In what follows, we shall study the behavior of (7) in the mean sense.

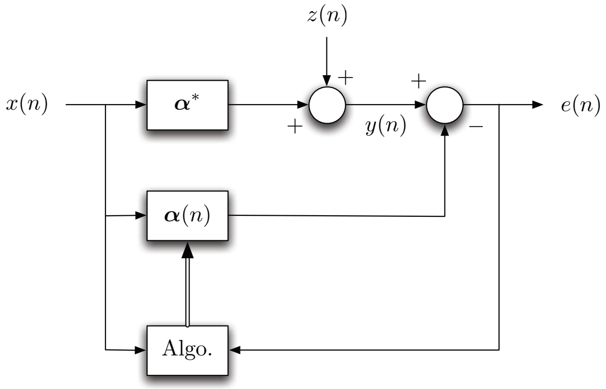

Fig. 1. Adaptive system under study

A. *Mean weight behavior analysis*

Defining the weight-error vector $\boldsymbol{v}(n)$ with respect to the unconstrained solution $\boldsymbol{\alpha}^*$, $\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*$, the update equation (7) can be written as

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\, e(n)\, \boldsymbol{D}_x(n) \left(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*\right)^{\gamma}. \tag{9}$$

Due to the nonlinear term $(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*)^{\gamma}$ in the r.h.s of (9), calculations of $\boldsymbol{v}(n{+}1)$ cannot be easily operated. We thus use the first order approximation to linearize this term. Expressed near a point v_i^o, it is written as

$$\begin{aligned}
&(v_i(n) + \alpha_i^*)^{\gamma} \\
\approx&(v_i^o + \alpha_i^*)^{\gamma} + \gamma\, (v_i^o + \alpha_i^*)^{\gamma-1}\, (v_i(n) - v_i^o) \\
=&((v_i^o + \alpha_i^*)^{\gamma} - \gamma\, (v_i^o + \alpha_i^*)^{\gamma-1}\, v_i^o) + \gamma\, (v_i^o + \alpha_i^*)^{\gamma-1}\, v_i(n)
\end{aligned}$$

For sake of notational simplicity, we denote

$$\begin{aligned}
r_i &= (v_i^o + \alpha_i^*)^{\gamma} - \gamma\, (v_i^o + \alpha_i^*)^{\gamma-1}\, v_i^o \\
s_i &= \gamma\, (v_i^o + \alpha_i^*)^{\gamma-1}
\end{aligned}$$

In order to calculate $E\{\boldsymbol{v}(n+1)\}$, the point $\boldsymbol{v}^o$ should be chosen at each iteration. A direct choice of $\boldsymbol{v}^o$ is $E\{\boldsymbol{v}(n)\}$, which can be viewed as a constant during this iteration. Consequently $\boldsymbol{r}$ and $\boldsymbol{s}$ are then denoted by $\boldsymbol{r}(n)$ and $\boldsymbol{s}(n)$ with the instant index n. The expected value $E\{\boldsymbol{v}(n+1)\}$

can be expressed in vector form by

$$\begin{aligned}&E\{\boldsymbol{v}(n+1)\}\\ &=E\{\boldsymbol{v}(n)\} + \eta\, E\{\, z(n)\, \boldsymbol{D}_x(n)\}\, E\{\big(\boldsymbol{r}(n) + \boldsymbol{D}_s(n)\, \boldsymbol{v}(n)\big)\}\\ &\quad - \eta\, E\{\boldsymbol{D}_x(n)\, (\boldsymbol{r}(n) + \boldsymbol{D}_s(n)\, \boldsymbol{v}(n))\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n)\}.\end{aligned}$$

where $\boldsymbol{D}_s(n)$ is the diagonal matrix with $\boldsymbol{s}(n)$ as the diagonal elements. Considering that $E\{z(n)\, \boldsymbol{D}_x(n)\} = 0$, the above expression becomes

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = E\{\boldsymbol{v}(n)\} - \eta\, \boldsymbol{D}_r(n)\, E\{\boldsymbol{x}(n)\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n)\}\\ - \eta \boldsymbol{D}_s(n)\, E\{\boldsymbol{D}(x)\boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\}\end{aligned} \tag{10}$$

where $\boldsymbol{D}_r(n)$ is the diagonal matrix with $\boldsymbol{r}(n)$ as the diagonal elements. The second expectation in the r.h.s of (10) is directly given by $E\{\boldsymbol{x}(n)\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n)\} = \boldsymbol{R}_x E\{\boldsymbol{v}(n)\}$. Then using the result we have obtained in the analysis of NN-LMS yields [9]

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = E\{\boldsymbol{v}(n)\} - \eta\, \boldsymbol{D}_r(n)\, \boldsymbol{R}_x\, E\{\boldsymbol{v}(n)\}\\ - \eta \boldsymbol{D}_s(n)\, \text{diag}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\}.\end{aligned} \tag{11}$$

where $\boldsymbol{K}(n)$ is the covariance matrix of $\boldsymbol{v}(n)$. This equation requires second-order moments defined by $\boldsymbol{K}(n)$ in order to update the first-order one $E\{\boldsymbol{v}(n)\}$. To simplify the analysis, we use the following separation approximation

$$\boldsymbol{K}(n) \approx E\{\boldsymbol{v}(n)\}\, E\{\boldsymbol{v}^\top(n)\} \tag{12}$$

We thus obtain the result

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = E\{\boldsymbol{v}(n)\} - \eta\, \boldsymbol{D}_r(n)\, \boldsymbol{R}_x\, E\{\boldsymbol{v}(n)\}\\ - \eta \boldsymbol{D}_s(n)\, \text{diag}\{\boldsymbol{R}_x\, E\{\boldsymbol{v}(n)\}\, E\{\boldsymbol{v}^\top(n)\}\}.\end{aligned} \tag{13}$$

The justification of this approximation can be found in [9] and extensive simulation results have shown us that approximations used in the derivation provide sufficient accuracy in modeling the mean behavior of the weights.

B. Excess mean square error analysis

We now present the model of second-order behavior. 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 $z(n)$ leads to the following expression for the mean-square estimation error (MSE):

$$\begin{aligned}E\{e^2(n)\} &= E\{(z(n) - \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n))(z(n) - \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n))\}\\ &= \sigma_z^2 + E\{\boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n)\}\\ &= \sigma_z^2 + \text{trace}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\}.\end{aligned} \tag{14}$$

The excess mean square error is defined by $J_{emse} = \text{trace}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\}$. We thus derive a recursion for $\boldsymbol{K}(n) = E\{\boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\}$ starting from the weight error update equation (9). Premultiplying equation (9) by its transpose, we obtain

$$\begin{aligned}\boldsymbol{v}(n+1)\boldsymbol{v}^\top(n+1) =&\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\\ &-\eta\, (\widetilde{\boldsymbol{P}}_1(n)) + \widetilde{\boldsymbol{P}}_1^\top(n)) + \eta^2\, \widetilde{\boldsymbol{P}}_2(n)\\ &+\eta^2\, (\widetilde{\boldsymbol{P}}_3(n) + \widetilde{\boldsymbol{P}}_3^\top(n)) + \eta^2\, \widetilde{\boldsymbol{P}}_4(n)\\ &-\eta\, (\widetilde{\boldsymbol{P}}_5(n) + \widetilde{\boldsymbol{P}}_5^\top(n)) + \eta^2\, \widetilde{\boldsymbol{P}}_6(n)\\ &+\eta^2\, (\widetilde{\boldsymbol{P}}_7(n) + \widetilde{\boldsymbol{P}}_7^\top(n)) + \eta^2\, \widetilde{\boldsymbol{P}}_8(n)\end{aligned} \tag{15}$$

As we approximate the nonlinear term $(\alpha_i^* + v_i(n))^\gamma$ using the first-order approximation which is of the form $(r_i(n) + s_i(n)v_i(n))$, the terms $\widetilde{\boldsymbol{P}}_1$ to $\widetilde{\boldsymbol{P}}_8$ are then given by

$$\begin{aligned}\widetilde{\boldsymbol{P}}_1 &= E\{\boldsymbol{v}(n)(\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_2 &= E\{z(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)(z(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_3 &= E\{z(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)(z(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_4 &= E\{z(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n)(z(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_5 &= E\{\boldsymbol{v}(n)(\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_6 &= E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)(\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_7 &= E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)\\ &\quad (\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n))^\top\}\\ \widetilde{\boldsymbol{P}}_8 &= E\{\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n)\\ &\quad (\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}(n))^\top\}\end{aligned}$$

where $\boldsymbol{D}_r(n)$ and $\boldsymbol{D}_s(n)$, as defined previously, are deterministic values at each iteration. Taking the expected value of (15), and using the statistical properties of $z(n)$, with very similar calculations to the analysis of NN-LMS algorithm in [9], we can obtain the recursion of $\boldsymbol{K}(n+1)$

$$\begin{aligned}\boldsymbol{K}(n+1) = \boldsymbol{K}(n)&\\ &- \eta\, (\boldsymbol{P}_1(n) + \boldsymbol{P}_1^\top(n)) + \eta^2\sigma_z^2\, \boldsymbol{P}_2(n)\\ &+ \eta^2\sigma_z^2\, (\boldsymbol{P}_3(n) + \boldsymbol{P}_3^\top(n)) + \eta^2\sigma_z^2\, \boldsymbol{P}_4(n)\\ &- \eta\, (\boldsymbol{P}_5(n) + \boldsymbol{P}_5^\top(n)) + \eta^2\, \boldsymbol{P}_6(n)\\ &+ \eta^2\, (\boldsymbol{P}_7(n) + \boldsymbol{P}_7^\top(n))\\ &+ \eta^2\, \boldsymbol{P}_8(n)\end{aligned} \tag{16}$$

with the terms $\boldsymbol{P}_1$ to $\boldsymbol{P}_8$ defined as

$$\boldsymbol{P}_1 = \boldsymbol{D}_r(n)\, \boldsymbol{R}_x\, \boldsymbol{K}(n) \tag{17}$$

$$\boldsymbol{P}_2 = \boldsymbol{D}_r(n)\, \boldsymbol{R}_x\, \boldsymbol{D}_r(n) \tag{18}$$

$$\boldsymbol{P}_3 = \boldsymbol{D}_r(n)\, \boldsymbol{R}_x\, E\{\boldsymbol{D}_v(n)\}\, \boldsymbol{D}_s(n) \tag{19}$$

$$\boldsymbol{P}_4 = \boldsymbol{D}_r(n)\, \boldsymbol{R}_x \circ \boldsymbol{K}(n)\, \boldsymbol{D}_r(n) \tag{20}$$

$$\boldsymbol{P}_5 = \boldsymbol{K}(n)\, \boldsymbol{R}_x\, E\{\boldsymbol{D}_v(n)\}\, \boldsymbol{D}_s(n) \tag{21}$$

$$\boldsymbol{P}_6 = \boldsymbol{D}_r(n)\, \boldsymbol{Q}(n)\, \boldsymbol{D}_r(n) \tag{22}$$

$$\boldsymbol{P}_7 = \boldsymbol{D}_r(n)\, \boldsymbol{Q}(n)\, E\{\boldsymbol{D}_v(n)\}\, \boldsymbol{D}_s(n) \tag{23}$$

$$\boldsymbol{P}_8 = \boldsymbol{D}_s(n)\, (\boldsymbol{Q}(n) \circ \boldsymbol{K}(n))\, \boldsymbol{D}_s(n) \tag{24}$$

where $\circ$ is the Hadamard product, and the matrix $\boldsymbol{Q}$ is defined as

$$\boldsymbol{Q}(n) = \boldsymbol{D}_{\alpha^*}\,(2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x)$$

Using $\boldsymbol{P}_1$ to $\boldsymbol{P}_8$, we can obtain the expression for $\boldsymbol{K}(n+1)$ as a function of $\boldsymbol{K}(n)$ for this modified algorithm to calculate the excess mean square error.

IV. Simulation Results

In this section, simulations were firstly conducted to verify the validity of the first-order and second-order behavior models of the modified NN-LMS algorithm. After that, we compared the performance of the modified algorithm with that of the original NN-LMS.

A. Consistency of models

We illustrate the accuracy of the model (13) through an example where inputs $x(n)$ are correlated in time. Consider a first-order AR model given by $x(n) = a\,x(n-1) + w(n)$, with $a = \frac{1}{2}$. The noise $w(n)$ was i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_w^2 = 1 - \frac{1}{4}$, so that $\sigma_x^2 = 1$. The noise $z(n)$ was i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_z^2 = 0.1$. The impulse response $\boldsymbol{\alpha}^*$ was given by

$$\boldsymbol{\alpha}^* = [0.8 \quad 0.6 \quad 0.5 \quad 0.4 \quad 0.3 \quad 0.2 \quad 0.1 \; -0.1 \; -0.3 \; -0.6]^\top \tag{25}$$

Note that it contains negative entries in order to better test the behavior of the algorithm. In this simulation, the parameter γ was set to $\frac{3}{7}$, namely

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\,e(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^{\frac{3}{7}}$$

The initial impulse response $\boldsymbol{\alpha}(0)$ was drawn from the uniform distribution $\mathcal{U}([0;1])$, and kept unchanged for all the simulations. Two step sizes $\eta = 10^{-2}$ and $\eta = 3\times10^{-3}$ were tested. The evolution of the mean value $E\{\alpha_i(n)\}$ was determined by

$$E\{\alpha_i(n)\} = E\{\boldsymbol{v}(n)\} + \boldsymbol{\alpha}^*$$

The simulation result is shown in Fig. 2. The simulation curves (blue line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (red line) were obtained from the models. One can notice that all the curves are perfectly superimposed.

The excess mean square error of these simulations are shown in Fig. 3. The theoretical results were obtained by $J_{emse} = \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}$ with $\boldsymbol{K}(n)$ calculated recursively by (16). Again one can notice that all the curves are perfectly superimposed.

B. Comparison with NN-LMS

We then compared NN-LMS and modified NN-LMS with adjusted step sizes so that they reached the same steady state error. The first order and second behavior of the two algorithms are shown in the Fig. 4. It can be observed that in NN-LMS algorithm small weights, especially those converging to zero, have much slower convergence rates than the others. However, unlike NN-LMS, our modified NN-LMS has more balanced weight convergence rates. The modified algorithm also has better faster convergence rate of excess mean square error than the original one. The comparison shows the advantage of the modified algorithm.

V. Conclusion

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 derived a modified NN-LMS algorithm in order to alleviate the unbalanced weight convergence rates of the original NN-LMS. We also proposed analytical models to characterize the stochastic behavior of this algorithm. Simulation illustrated the accuracy of the model and its advantage over the original algorithm. In future research efforts, we intend to further explore other variants of NN-LMS algorithm to adapt different practical requirement.

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

[4] 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.

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

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

[7] 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, 2011.

[8] 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.

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

[10] J. Chen, C. Richard, P. Honeine, H. Lantéri, and C. Theys, "System identification under non-negativity constraints," in *Proceedings of European Signal Processing Conference*, Aalborg, Denmark, 2010.

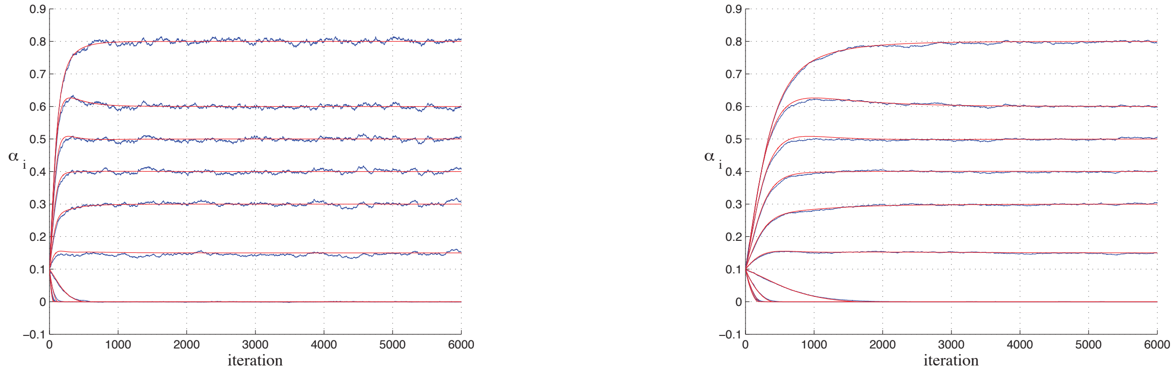

Fig. 2. Convergence of the coefficients $\alpha_i(n)$ for the modified NN-LMS, in the case of correlated input $x(n)$. Two different step sizes are considered: $\eta = 10^{-2}$ on the left, and $\eta = 3 \cdot 10^{-3}$ on the right. The theoretical curves (red line) obtained from the model (13) and simulation curves (blue line) are perfectly superimposed.

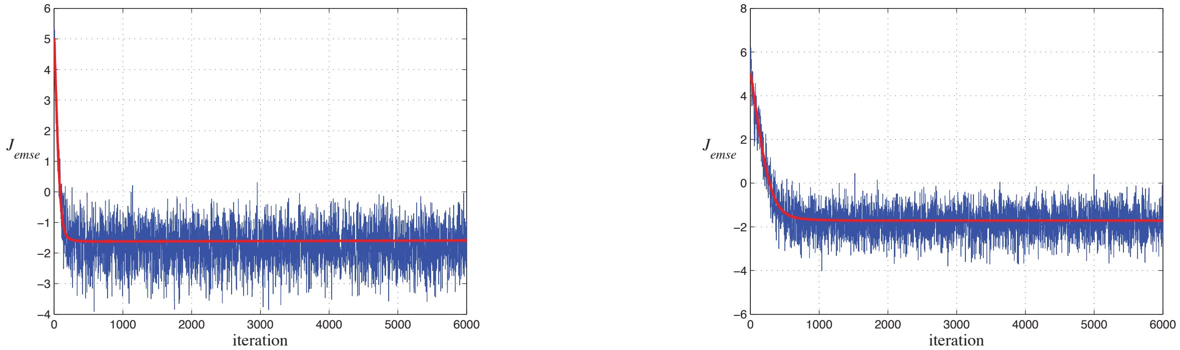

Fig. 3. Corresponding convergence of $J_{emse}(n)$ of the modified NN-LMS in these experiments. The theoretical curves (red line) were obtained with the model (16).

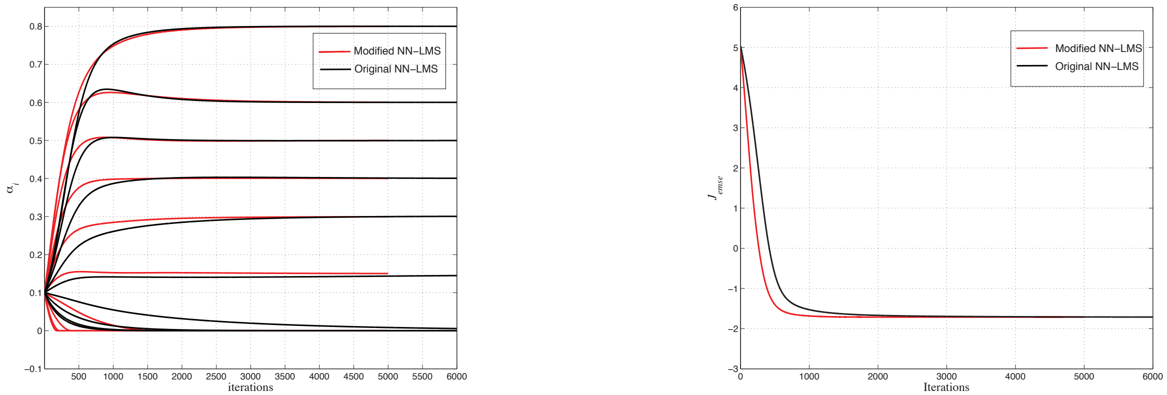

Fig. 4. Comparison of NN-LMS and modified NN-LMS. Left: first-order performance. The modified algorithm has a more balanced weight convergence rate. Right: Excess mean square error.