Variants of Non-Negative Least-Mean-Square Algorithm and Convergence Analysis

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

***Abstract*—Due to the inherent physical characteristics of systems under investigation, non-negativity is one of the most interesting constraints that can usually be imposed on the parameters to estimate. The Non-Negative Least-Mean-Square algorithm (NNLMS) was proposed to adaptively find solutions of a typical Wiener filtering problem but with the side constraint that the resulting weights need to be non-negative. It has been shown to have good convergence properties. Nevertheless, certain practical applications may benefit from the use of modified versions of this algorithm. In this paper, we derive three variants of NNLMS. Each variant aims at improving the NNLMS performance regarding one of the following aspects: sensitivity of input power, unbalance of convergence rates for different weights and computational cost. We study the stochastic behavior of the adaptive weights for these three new algorithms for non-stationary environments. This study leads to analytical models to predict the first and second order moment behaviors of the weights for Gaussian inputs. Simulation results are presented to illustrate the performance of the new algorithms and the accuracy of the derived models.**

***Index Terms*—Adaptive signal processing, convergence analysis, exponential algorithm, least-mean-square algorithms, non-negativity constraints, normalized algorithm, sign-sign algorithm.**

I. Introduction

OPTIMIZATION of a cost function given a set of constraints is a common objective in signal processing estimation problems. The constraints are usually imposed by system specifications which provide *a priori* information on the feasible set of solutions. The solution of estimation problems under constraints poses special problems for online applications. Common real-time signal processing restrictions on computational complexity and memory requirements tend to rule out several good solutions to the constrained optimization problem.

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. Non-negativity constraints have been 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 used in many applications [4]–[6]. This problem is closely related to blind deconvolution, and has found direct application in neuroscience [7] and in 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-square (NNLS) problems. 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 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]–[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 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 [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. In [20], the problem of online system identification under non-negativity constraints on the parameters to estimate was investigated. An LMS-type adaptive algorithm, called Non-Negative Least-Mean-Square (NNLMS) was proposed to solve the Wiener problem under the constraint that the resulting weights need to be non-negative. It was based on the stochastic gradient descent approach combined with a fixed point iteration which converges to a solution satisfying the Karush-Kuhn-Tucker conditions. The stochastic behavior of this algorithm was also analyzed in [20], [21].

In this paper, we extend the work of [20], [22] and derive useful variants of the NNLMS algorithm. Each of these variants is derived to improve the NNLMS properties in some sense. A

Manuscript received February 11, 2014; revised May 18, 2014; accepted June 03, 2014. Date of publication June 24, 2014; date of current version July 10, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Urbashi Mitra. This work has been partly supported by CNPq Grant No. 307071/2013-8 and by CAPES—Proc. 3367-11-8.

J. Chen and C. Richard are with the Cote d'Azur Observatory, CNRS, University of Nice Sophia-Antipolis, Parc Valrose, 06108 Nice Cedex 2, France (e-mail: jie.chen@unice.fr; cedric.richard@unice.fr).

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

P. Honeine is with the Institute Charles Delaunay, CNRS, University of Technology of Troyes, 10010 Troyes Cedex, France (e-mail: paul.honeine@utt.fr).

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

Digital Object Identifier 10.1109/TSP.2014.2332440

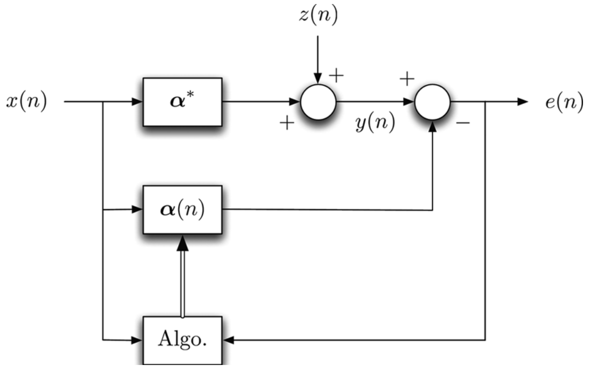


Fig. 1. Adaptive system under study.

normalized algorithm is proposed to reduce the NNLMS performance sensitivity to the input power value. An exponential algorithm is proposed to improve the balance of weight convergence rates. Compared to NNLMS, the new algorithm leads to faster convergence of the weights in the active set (weights for which the inequality constraint is satisfied with the equal sign). Finally, a sign-based algorithm is proposed to reduce implementation cost in critical real-time applications.

The rest of this paper is organized as follows. Section II reviews the system identification problem under non-negative constraints and the NNLMS algorithm. Section III motivates and introduces the NNLMS variants. In Sections IV and V, the transient behavior of each of these variants is analyzed. Analytical models are derived for the mean weight and for the mean-square error behavior. The accuracy of these models is illustrated through simulations. A final example compares the performance of the proposed algorithms with those of NLMS and Projected NLMS in solving an unconstrained non-negative parameter estimation problem.

II. Review of Non-Negative Least-Mean-Square Algorithm

Consider the estimation problem depicted in Fig. 1. The unknown system is characterized by real-valued observations

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

where $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \dots, \alpha_{\mathrm{N}}^*]^\top$ is the vector of the model parameters and $\boldsymbol{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^\top$ 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 of systems under investigation impose a non-negativity constraint on the estimate $\boldsymbol{\alpha}$ of the system parameters. Therefore, the problem of identifying the optimum non-negative model can be formalized as follows

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

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

A. A Fixed-Point Iteration Scheme

To solve the problem (2), let us consider its Lagrangian function $Q(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ given by [23]

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

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 $\boldsymbol{\alpha}^\circ$, $\boldsymbol{\lambda}^\circ$, namely,

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

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

where ∇_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

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

where the extra minus sign is just used to make a gradient descent of $J(\boldsymbol{\alpha})$ apparent. To solve (5) iteratively, two important points have to be noticed. First, $\boldsymbol{D}(-\nabla_\alpha J(\boldsymbol{\alpha}))$ is also a descent direction of $J(\boldsymbol{\alpha})$ if $\boldsymbol{D}$ is a symmetric positive definite matrix. Second, 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 (5) leads to the component-wise gradient descent algorithm

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

where $\eta_i(n)$ a positive step size required to get a contraction scheme and to control the convergence rate, and $f_i(\boldsymbol{\alpha}(n)) > 0$ is the i-th entry of a diagonal matrix $\boldsymbol{D}(n)$. Function $f_i(\boldsymbol{\alpha}(n))$ in (6) is an arbitrary positive function of $\boldsymbol{\alpha}(n)$. Some criteria $J(\boldsymbol{\alpha})$ are defined only for parameter vectors $\boldsymbol{\alpha}$ with positive entries, e.g., the Itakura-Saito distance and the 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 $\alpha_i(n+1)$ in (6) 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

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

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

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

the update (6) can be written in vector form as

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

where the ith entry of the weight adjustment direction $\boldsymbol{d}(n)$ 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 descent direction of $J(\boldsymbol{\alpha})$ 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 algorithm stability.

B. The Non-Negative Least-Mean-Square Algorithm

Let us now consider the mean-square error criterion $J_{\mathrm{mse}}(\boldsymbol{\alpha}) = E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\}$ to be minimized with respect to $\boldsymbol{\alpha}$ so that

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

The gradient of $J_{\mathrm{mse}}(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ is $\nabla_\alpha J(\boldsymbol{\alpha}) = 2\,(\boldsymbol{R}_x\boldsymbol{\alpha} - \boldsymbol{r}_{xy})$, 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 (11) by the instantaneous estimates $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$ and $y(n)\,\boldsymbol{x}(n)$, respectively. Then, choosing $f_i(\boldsymbol{\alpha}(n)) = 1/2$ for all i in (6), a fixed positive step-size η, defining $\boldsymbol{D}_x(n) = \mathrm{diag}\{\boldsymbol{x}(n)\}$ and $\boldsymbol{D}_\alpha(n) = \mathrm{diag}\{\boldsymbol{\alpha}(n)\}$, and noting that $\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n) = \boldsymbol{D}_x(n)\boldsymbol{\alpha}(n)$ leads to the stochastic update given by

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}(n) \tag{13}$$

where $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n)$. A detailed study of this algorithm, named Non-Negative LMS (NNLMS) can be found in [20].

III. Variants of the Non-Negative Least-Mean-Square Algorithm

A. Normalized NNLMS

A direct extension of the original algorithm is the Normalized NNLMS. Conditioned on $\boldsymbol{\alpha}(n)$, the product $e(n)\boldsymbol{D}_x(n)$ in (13) has dimension of signal power. Thus, η is inversely proportional to signal power. Hence, setting a constant value for 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

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)}\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{N}}(n). \tag{14}$$

Like in Normalized LMS (NLMS) algorithm, adding a small positive regularization parameter ϵ to the denominator $\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)$ may be necessary to avoid numerical difficulties when $\boldsymbol{x}^\top(n)\boldsymbol{x}(n)$ becomes very small. The resulting ϵ-Normalized NNLMS will then be

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n) + \epsilon}\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{N}}(n) \tag{15}$$

where we maintained the notation $\boldsymbol{\alpha}_{\mathrm{N}}(n)$ because (14) is a particular case of (15) for $\epsilon = 0$. From now on, we refer to (15) simply as the Normalized NNLMS algorithm.

B. Exponential NNLMS

Each component $\alpha_i(n)$ in the update term of (13) can be viewed as a distinct variable step size adjustment along the ith axis[1]. Hence, each component of $\boldsymbol{\alpha}(n)$ will have a different convergence rate in general. Specifically in the case of weights in the active set (those that tend to zero in steady-state), the convergence rate will progressively reduce in time, becoming very small near steady-state. To alleviate this convergence rate unbalance, we introduce the Exponential NNLMS algorithm.

To achieve a faster convergence for the adaptive coefficients as they get close to zero we propose the use of $f_i(\boldsymbol{\alpha}_{\mathrm{E}}(n)) = \alpha_{\mathrm{E}_i}^{\gamma-1}(n)$ in (11), with parameter γ chosen in order to attract small values of $\alpha_{\mathrm{E}_i}(n)$ towards zero. This leads to the ith weight update equation

$$\alpha_{\mathrm{E}_i}(n+1) = \alpha_{\mathrm{E}_i}(n) + \eta\, e(n)\, x_i(n)\, \alpha_{\mathrm{E}_i}^{\gamma}(n). \tag{16}$$

For $0 < \gamma < 1$, the ith weight update in (16) becomes larger than that in (13) when $|\alpha_{\mathrm{E}_i}(n)| < 1$, thus accelerating convergence towards a null steady-state coefficient value.

The condition for $\alpha_{\mathrm{E}_i}(n+1) \geq 0$ given $\alpha_{\mathrm{E}_i}(n) \geq 0$ can be easily determined from (16) as

$$\eta \leq \frac{1}{e(n)\, \alpha_{\mathrm{E}_i}^{\gamma-1}(n)}, \quad \forall\, i, n. \tag{17}$$

This condition, however, is not useful for design purposes, since it requires *a priori* knowledge of the algorithm behavior. We then propose a modified version of the update (16) that allows for instantaneous negative values of $\alpha_{\mathrm{E}_i}(n)$. The problem with real and negative instantaneous values of $\alpha_{\mathrm{E}_i}(n)$ is that it may lead to a complex value for $\alpha_{\mathrm{E}_i}^{\gamma}$ for $0 < \gamma < 1$. To obtain always real values for $\alpha_{\mathrm{E}_i}^{\gamma}$ we propose to use $\gamma = p/q$ with p and q odd integers and $0 < p < q$. The oddness of p and q guarantees that $\mathrm{sgn}(\alpha_{\mathrm{E}_i}^{\gamma}(n)) = \mathrm{sgn}(\alpha_{\mathrm{E}_i}(n))$. Then, the real solution for $\alpha_{\mathrm{E}_i}^{\gamma}$ can be obtained by calculating $\mathrm{sgn}\{\alpha_{\mathrm{E}_i}\}|\alpha_{\mathrm{E}_i}|^{\gamma}$. This leads to the following weight update equation for the Exponential NNLMS algorithm in vector form:

$$\boldsymbol{\alpha}_{\mathrm{E}}(n+1) = \boldsymbol{\alpha}_{\mathrm{E}}(n) + \eta e(n)\boldsymbol{D}_x(n)\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n) \tag{18a}$$

with the ith component of $\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)$ defined as

$$[\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)]_i = \mathrm{sign}\{\alpha_{\mathrm{E}_i}(n)\}|\alpha_{\mathrm{E}_i}(n)|^{\gamma}. \tag{18b}$$

As in the gamma correction used in image processing, an exponent in the range $0 < \gamma < 1$ reduces the dynamic range of

[1] Note that (13) can be written as $\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\boldsymbol{D}_\alpha(n)\boldsymbol{x}(n)e(n)$. Hence, $\boldsymbol{D}_\alpha(n)$ multiplies the ith gradient component $e(n)x(n-i+1)$ of the classical LMS algorithm by $\alpha_i(n)$.

TABLE I
COMPUTATIONAL COMPLEXITY

Algorithm	Recursion	Computational cost per iteration				Main property
		$+$	$\times$	sgn	$(\cdot)^{\gamma}$	
NNLMS	Equation (13)	$2N$	$3N+1$			Original one, simplicity
Norm. NNLMS	Equation (15)	$3N$	$4N+1$			Insensitivity to input power
Exp. NNLMS	Equation (18)	$2N$	$3N+1$		N	Balance on weight convergence
S-S NNLMS	Equation (19)	N	$2N$	N		Reduced computational cost

each $\alpha_{\mathrm{E}_i}(n)$. Large values of $\alpha_{\mathrm{E}_i}(n)$ will be compressed towards 1 and small values of $\alpha_{\mathrm{E}_i}(n)$ will be increased to prevent from stalling convergence. When $\gamma = 1$, the update equation degenerates into the NNLMS algorithm (13). Using $\gamma > 1$ is generally not recommended, as it tends to spread the vector component values.

C. Sign-Sign NNLMS

Like Sign-Sign LMS, which has been included in the CCITT standard for adaptive differential pulse code modulation [24], the motivation for introducing a Sign-Sign NNLMS algorithm is its computational simplicity and its robustness against disturbances [25]. Replacing the input regressor vector and the estimation error in the update term with their signs reduces computation time and dynamic range requirements by replacing multiplications with shifts in real-time implementations. The Sign-Sign NNLMS algorithm is given by

$$\boldsymbol{\alpha}_{\mathrm{S}}(n+1) = \boldsymbol{\alpha}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{e(n)\}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\}\,\boldsymbol{\alpha}_{\mathrm{S}}(n). \quad (19)$$

After the two signs are evaluated, the ith component update is given by

$$\alpha_{\mathrm{S}_i}(n+1) = \alpha_{\mathrm{S}_i}(n) \pm \eta\,\alpha_{\mathrm{S}_i}(n) \quad (20)$$

where the sign before η is determined by $\mathrm{sgn}\{e(n)x_i(n)\}$. The step-size η is usually selected as a power of 2^{-1}, say $\eta = 2^{-m}$ for some integer $m > 0$. In this case, (20) can be efficiently implemented using shift-add operations. Moreover, the non-negativity constraint will be always satisfied if $\boldsymbol{\alpha}_{\mathrm{S}}$ is initialized with a positive vector and $0 < \eta < 1$. Table I compares the computation complexities of NNLMS and its three variants described above. The rightmost column describes the anticipated property of each algorithm, to be verified in the following.

IV. MEAN WEIGHT BEHAVIOR

Convergence in the mean sense of the NNLMS algorithm (13) has been studied in [20] for a stationary environment. We now study the stochastic behavior of the NNLMS variants introduced in Section III for fixed step sizes and for a time variant unconstrained solution given by

$$\boldsymbol{\alpha}^*(n) = \boldsymbol{\alpha}^*_{\mathrm{o}}(n) + \boldsymbol{\xi}(n) \quad (21)$$

where $\boldsymbol{\alpha}^*_{\mathrm{o}}(n)$ is a deterministic time-variant mean and $\boldsymbol{\xi}(n)$ is zero-mean with covariance matrix $\boldsymbol{\Xi} = \sigma_{\xi}^2\boldsymbol{I}$ and independent of any other signal. This simple model provides some information on how the performances of the proposed algorithms are affected by a time variant optimal solution which consists of a deterministic trajectory and a random perturbation. The analysis using more elaborate non-stationarity models such as the random walk model or the autoregressive model [26] leads to mathematically intractable situations. This is due to the extra multiplication of the weight update term by a function of $\boldsymbol{\alpha}(n)$ in (13), (15), (18) and (19), as compared to the LMS algorithm. For the random walk model, the recursive equation for the covariance matrix of the adaptive weight vector becomes a function of the optimal weight covariance matrix, which becomes unbounded as time progresses [26]. For the autoregressive model, a nonlinear term given by the product of the weight error vector and the optimal weight update makes it impossible to determine a recursive adaptive weight vector covariance matrix equation in the state-space form [27]. The model (21) leads to a tractable analysis and still permits inferences about the behavior of the algorithms in randomly time variant environments by varying the power σ_{ξ}^2 of $\boldsymbol{\xi}(n)$. Inferences on the ability of the algorithm to track mean weight variations are also possible but require a different model run for each type of mean time variation of $\boldsymbol{\alpha}^*_{\mathrm{o}}(n)$ to be investigated.

To conserve space and to simplify notation without ambiguity, from now on we use the generic notations $\boldsymbol{\alpha}(n)$ and $\alpha_i(n)$ whenever the given expression is valid for all the algorithms under study. Notations $\boldsymbol{\alpha}_{\mathrm{N}}$, $\boldsymbol{\alpha}_{\mathrm{E}}$ and $\boldsymbol{\alpha}_{\mathrm{S}}$ will be used only for expressions which are specific to the corresponding algorithm. The same notational observation applies to any vector or matrix when referring to a specific algorithm.

For the analyses that follow, we shall define the weight error vector with respect to the unconstrained solution $\boldsymbol{\alpha}^*(n)$ as

$$\tilde{\boldsymbol{v}}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*(n) \quad (22)$$

and the weight error vector with respect to the mean unconstrained solution $\boldsymbol{\alpha}^*_{\mathrm{o}}(n)$ as

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*_{\mathrm{o}}(n). \quad (23)$$

The two vectors are related by $\tilde{\boldsymbol{v}}(n) = \boldsymbol{v}(n) - \boldsymbol{\xi}(n)$.

A. Statistical Assumptions

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 Independence Assumption [26]. This assumption is typical in the study of adaptive algorithms. It is sometimes used for simplification and frequently required for mathematical tractability. The simulation results will show that the resulting analytical models have low sensitivity to this assumption, as they accurately predict the behavior of the three algorithms.

B. Normalized NNLMS Algorithm

Using (22) with the appropriate subscript in (15) and $e(n) = y(n) - \boldsymbol{\alpha}_{\mathrm{N}}^\top(n)\boldsymbol{x}(n) = z(n) - (\boldsymbol{v}_{\mathrm{N}}(n) - \boldsymbol{\xi}(n))^\top\boldsymbol{x}(n)$ yields

$$\begin{aligned}\boldsymbol{v}_{\mathrm{N}}(n+1) = \boldsymbol{v}_{\mathrm{N}}(n) &+ \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,z(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\\ &+ \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,z(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\\ &- \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\\ &- \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\\ &+ \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{\xi}^\top(n)\,\boldsymbol{x}(n)\\ &+ \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\,\boldsymbol{\xi}^\top(n)\,\boldsymbol{x}(n) - \boldsymbol{\Delta}_{\mathrm{N}}(n)\end{aligned} \tag{24}$$

where $\boldsymbol{\Delta}_{\mathrm{N}}(n) = \boldsymbol{\alpha}_{\circ}^*(n+1) - \boldsymbol{\alpha}_{\circ}^*(n)$ is a deterministic vector proportional to the derivative of the mean unconstrained optimal solution.

Taking the expected value of (24) 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

$$\begin{aligned}E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} &= E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\\ &-\eta E\left\{\frac{1}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\}\\ &-\eta E\left\{\frac{1}{\boldsymbol{x}^\top(n)\boldsymbol{x}(n)+\epsilon}\boldsymbol{D}_x(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{x}(n)\right\} - \boldsymbol{\Delta}_{\mathrm{N}}(n).\end{aligned} \tag{25}$$

Using the independence assumption, the second expectation in the r.h.s. of (25) can be written as

$$\begin{aligned}E&\left\{\frac{1}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\}\\ &= \boldsymbol{D}_{\alpha_{\circ}^*}(n)\,E\left\{\frac{\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\right\}\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}.\end{aligned} \tag{26}$$

Evaluation of the first expected value in the r.h.s. of (26) requires approximations. Each numerator element is given by $x(n-i)x(n-j)$. The random part of the denominator is given by $\sum_{k=0}^{N-1} x^2(n-k)$. A common approximation that works well for reasonably large N is to neglect the correlation between these two variates, as the latter tends to vary much slower than the former [28], [29]. Moreover, given its slow variation we approximate $\boldsymbol{x}^\top(n)\boldsymbol{x}(n)$ by its mean value $N\sigma_x^2$, which is reasonable for large values of N. Using these approximations yields

$$\begin{aligned}E&\left\{\frac{1}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)+\epsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\circ}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\}\\ &\approx \frac{1}{N\sigma_x^2+\epsilon}\boldsymbol{D}_{\alpha_{\circ}^*}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}.\end{aligned} \tag{27}$$

Using again $\boldsymbol{x}^\top(n)\boldsymbol{x}(n) \approx N\sigma_x^2$ and removing it from the expected value, the ith component of the second expectation in the r.h.s. of (25) is

$$\begin{aligned}\Big[&\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\Big]_i\\ &= \sum_{j=1}^{N} x(n-i+1)\,v_{\mathrm{N}_i}(n)\,v_{\mathrm{N}_j}(n)\,x(n-j+1).\end{aligned} \tag{28}$$

Taking the expectation, using the independence assumption and defining $\boldsymbol{K}_{\mathrm{N}}(n) = E\{\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\}$ we obtain

$$\begin{aligned}\Big[E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\}\Big]_i &= \sum_{j=1}^{N} r_x(j-i)\,[\boldsymbol{K}_{\mathrm{N}}(n)]_{ij}\\ &= [\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)]_{ii}\end{aligned} \tag{29}$$

which yields

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

where $\operatorname{diag}\{\cdot\}$ denotes the vector of diagonal entries in the matrix. Hence, (25) becomes

$$\begin{aligned}E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} &= \left(\boldsymbol{I} - \frac{\eta}{N\sigma_x^2+\epsilon}\,\boldsymbol{D}_{\alpha_{\circ}^*}(n)\,\boldsymbol{R}_x\right)\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\\ &-\frac{\eta}{N\sigma_x^2+\epsilon}\operatorname{diag}\{\boldsymbol{R}_x\boldsymbol{K}_{\mathrm{N}}(n)\} - \boldsymbol{\Delta}_{\mathrm{N}}(n).\end{aligned} \tag{30}$$

This recursion for $E\{\boldsymbol{v}_{\mathrm{N}}(n)\}$ requires a model for $\boldsymbol{K}_{\mathrm{N}}(n)$. A recursive model will be derived for $\boldsymbol{K}_{\mathrm{N}}(n)$ in Section V, see (44). That model can be used along with (30) to predict the mean weight behavior of the Normalized NNLMS algorithm. Nevertheless, we have found that a sufficiently accurate and more intuitive mean behavior model can be obtained by neglecting the weight error fluctuations and using the following separation approximation

$$\boldsymbol{K}_{\mathrm{N}}(n) \approx E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\,E\{\boldsymbol{v}_{\mathrm{N}}^\top(n)\}. \tag{31}$$

This approximation has been successfully used in [20] to study the mean behavior of the NNLMS algorithm. A discussion about the validity of the approximation can be found in [20]. Extensive simulation results have shown that this approximation achieves adequate accuracy in modeling the mean behavior of the adaptive weights. We thus obtain the following model

$$\begin{aligned}E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} &= \left(\boldsymbol{I} - \frac{\eta}{N\sigma_x^2+\epsilon}\,\boldsymbol{D}_{\alpha_{\circ}^*}(n)\,\boldsymbol{R}_x\right)\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\\ &-\frac{\eta}{N\sigma_x^2+\epsilon}\operatorname{diag}\Big\{\boldsymbol{R}_x\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\,E\{\boldsymbol{v}_{\mathrm{N}}^\top(n)\}\Big\} - \boldsymbol{\Delta}_{\mathrm{N}}(n).\end{aligned} \tag{32}$$

C. Exponential NNLMS Algorithm

Using (22) with the appropriate subscript in (18), $e(n) = y(n) - \boldsymbol{\alpha}_{\mathrm{E}}^\top(n)\boldsymbol{x}(n) = z(n) - (\boldsymbol{v}_{\mathrm{E}}(n) - \boldsymbol{\xi}(n))^\top\boldsymbol{x}(n)$, and considering that $\operatorname{sign}\{\alpha_{\mathrm{E}_i}\}|\alpha_{\mathrm{E}_i}|^\gamma$ is equal to the real solution of $\alpha_{\mathrm{E}_i}^\gamma$, the Exponential NNLMS weight error update equation can be written as

$$\begin{aligned}&\boldsymbol{v}_{\mathrm{E}}(n+1)\\&=\boldsymbol{v}_{\mathrm{E}}(n)+\eta\,e(n)\,\boldsymbol{D}_x(n)\,\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}\\&=\boldsymbol{v}_{\mathrm{E}}(n)+\eta\,z(n)\,\boldsymbol{D}_x(n)\,\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}\\&\quad-\eta\,\boldsymbol{D}_x(n)\,\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}\,\boldsymbol{v}_{\mathrm{E}}^\top(n)\,\boldsymbol{x}(n)\\&\quad+\eta\,\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\,\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}-\boldsymbol{\Delta}_{\mathrm{E}}(n)\end{aligned}\tag{33}$$

where $\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}$ is a real vector.

The nonlinear term $(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)(n))^{(\gamma)}$ on the r.h.s. complicates the evaluation of the expected value of (33) because the statistics of the weight error vector are unknown. We have again found out that using a zero-th order approximation of $\boldsymbol{v}_{\mathrm{E}}(n)$ is sufficient to provide a reasonably good model for the mean weight error behavior. Thus, we make

$$(v_{\mathrm{E}_i}(n)+\alpha^*_{\mathrm{o}_i}(n))^\gamma\approx(E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^\gamma.\tag{34}$$

Using (34) in (33), taking the expected value and considering the statistical properties of $z(n)$ and $\boldsymbol{\xi}(n)$ yields

$$\begin{aligned}&E\{\boldsymbol{v}_{\mathrm{E}}(n+1)\}\\&\approx E\{\boldsymbol{v}_{\mathrm{E}}(n)\}-\eta\,E\{\boldsymbol{D}_x(n)\big(E\{\boldsymbol{v}_{\mathrm{E}}(n)\}+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)}\\&\qquad\cdot\boldsymbol{x}^\top(n)\boldsymbol{v}_{\mathrm{E}}(n)\}-\boldsymbol{\Delta}_{\mathrm{E}}(n)\\&=\Big(\boldsymbol{I}_{\mathrm{N}}-\eta\,\boldsymbol{D}_r(n)\boldsymbol{R}_x\Big)E\{\boldsymbol{v}_{\mathrm{E}}(n)\}-\boldsymbol{\Delta}_{\mathrm{E}}(n)\end{aligned}\tag{35}$$

where $\boldsymbol{I}_{\mathrm{N}}$ is the $N\times N$ identity matrix and $\boldsymbol{D}_r(n)$ is an $N\times N$ diagonal matrix defined as $\boldsymbol{D}_r(n)=\operatorname{diag}\{\boldsymbol{r}(n)\}$ with $\boldsymbol{r}(n)$ being the $N\times 1$ vector whose ith component is $r_i(n)=(E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^\gamma$. It is simple to verify that this model collapses to the NNLMS model derived in [20] for $p=q=1$.

D. Sign-Sign NNLMS Algorithm

The statistical analysis of the Sign-Sign NNLMS algorithm behavior is complicated by the fact that the weight update term is discontinuous in both the input vector $\boldsymbol{x}(n)$ and the error $e(n)$ [30]. To make that analysis tractable, we consider the case of input signal $x(n)$ zero-mean and Gaussian [25], [30].

Using (22) with the appropriate subscript in (19) and $e(n)=y(n)-\boldsymbol{\alpha}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n)=z(n)-(\boldsymbol{v}_{\mathrm{S}}(n)-\boldsymbol{\xi}(n))^\top\boldsymbol{x}(n)$, the Sign-Sign NNLMS weight error update equation can be written as

$$\begin{aligned}\boldsymbol{v}_{\mathrm{S}}(n+1)=\boldsymbol{v}_{\mathrm{S}}(n)+\eta\operatorname{sgn}\{z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n)+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\}\\\cdot\operatorname{sgn}\{\boldsymbol{D}_x(n)\}\,(\boldsymbol{v}_{\mathrm{S}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n))-\boldsymbol{\Delta}_{\mathrm{S}}(n).\end{aligned}\tag{36}$$

Note that, unlike the former two variants, the non-stationarity effect appears in the weight error update (36) as a nonlinear function of $\boldsymbol{\xi}(n)$. The ith component of (36) is given by

$$\begin{aligned}&v_{\mathrm{S}_i}(n+1)=v_{\mathrm{S}_i}(n)+\eta\operatorname{sgn}\big\{z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n)\\&+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\big\}\operatorname{sgn}\big\{x_i(n)\big\}\,(v_{\mathrm{S}_i}(n)+\alpha^*_{\mathrm{o}_i}(n))-\Delta_{\mathrm{S}_i}(n).\end{aligned}\tag{37}$$

To determine the expected value of (37), we first note that it has been demonstrated in [30], [31] using Price's theorem [32] that

$$E\Big\{\operatorname{sgn}\{\theta_1\}\operatorname{sgn}\{\theta_2\}\Big\}=\frac{2}{\pi}\,\sin^{-1}\left(\frac{E\{\theta_1\theta_2\}}{\sigma_{\theta_1}\sigma_{\theta_2}}\right)\tag{38}$$

for θ_1 and θ_2 two zero-mean jointly Gaussian variables with variances $\sigma^2_{\theta_1}$ and $\sigma^2_{\theta_2}$, respectively. Then, noting that $z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n)+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)$ and $x_i(n)$ are zero-mean Gaussian when conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$, we assume them jointly Gaussian[2] and use the result in (38) to obtain

$$\begin{aligned}&E\Big\{\operatorname{sgn}\{z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\,\boldsymbol{x}(n)+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\}\\&\qquad\cdot\operatorname{sgn}\{x_i(n)\}|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)\Big\}\\&\approx\frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\,\boldsymbol{v}_{\mathrm{S}}(n)-\boldsymbol{R}_i^\top\,\boldsymbol{\xi}(n)}{\sigma_x\;\sigma_{e|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)}}\right)\end{aligned}\tag{39}$$

where $\boldsymbol{R}_i$ the i-th column of $\boldsymbol{R}$ and $\sigma^2_{e|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)}$ is the variance of $e(n)$ when conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$.

Now, since $\sin^{-1}(\cdot)$ is a nonlinear function and the distribution of its argument is unknown, we proceed as we did for the Exponential NNLMS algorithm and replace the nonlinear function by its zero-th order approximation

$$\begin{aligned}&E\Big\{\operatorname{sgn}\{z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\,\boldsymbol{x}(n)+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\}\operatorname{sgn}\{x_i(n)\}\Big\}\\&\qquad\approx\frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\;E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)\end{aligned}\tag{40}$$

with

$$\begin{aligned}&\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}\\&=y\sqrt{\sigma_z^2+\operatorname{tr}\Big\{\boldsymbol{R}_x\;E\{\boldsymbol{v}_{\mathrm{S}}(n)\}E\{\boldsymbol{v}_{\mathrm{S}}^\top(n)\}\Big\}+\operatorname{trace}\{\boldsymbol{R}_x\boldsymbol{\Xi}\}}.\end{aligned}\tag{41}$$

Taking the expected value of (37), using the results (39) and (40) and expressing the result in vector form yields the mean weight error vector behavior model

$$\begin{aligned}E\{\boldsymbol{v}_{\mathrm{S}}(n+1)\}=\Big(\boldsymbol{I}_{\mathrm{N}}+\eta\,\boldsymbol{D}_p(n)\Big)E\{\boldsymbol{v}_{\mathrm{S}}(n)\}\\+\eta\,\boldsymbol{D}_p(n)\,\boldsymbol{\alpha}^*_{\mathrm{o}}(n)-\boldsymbol{\Delta}_{\mathrm{S}}(n)\end{aligned}\tag{42}$$

where $\boldsymbol{D}_p(n)$ is the $N\times N$ diagonal matrix $\boldsymbol{D}_p(n)=\operatorname{diag}\{\boldsymbol{p}(n)\}$ with $\boldsymbol{p}(n)$ being the $N\times 1$ vector whose ith entry is given by (40).

V. Second Moment Analysis

We now study the behavior of the second-order moments of the adaptive weights for the three algorithms proposed in Section III. The analysis is performed under the same statistical hypotheses used in the previous section. The following additional assumptions are used in the subsequent analysis:

A1) The input vector $\boldsymbol{x}(n)$ is 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 [33].

These assumptions are typical in the study of adaptive algorithms. They are sometimes used for simplification and sometimes required for mathematical feasibility.

[2] As $\boldsymbol{x}(n)$ and $z(n)$ are independent and both Gaussian, $[\boldsymbol{x}(n),z(n)]$ is jointly Gaussian. When conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$, $[\boldsymbol{x}^\top(n),z(n)-\boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n)+\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)]$ is jointly Gaussian as a linear transformation of $[\boldsymbol{x}^\top(n),z(n)]$.

Besides assumptions A1 and A2, the following approximation is needed to progress in the analysis as the distribution of $\boldsymbol{v}(n)$ is unknown:

A3) In evaluating higher order (greater than 2) moments of $\boldsymbol{v}(n)$, we approximate $\boldsymbol{v}(n)\boldsymbol{v}^\top(n)$ by its mean value $\boldsymbol{K}(n) = E[\boldsymbol{v}(n)\boldsymbol{v}^\top(n)]$. This approximation preserves the mean (in odd order moments) and fluctuation behaviors of $\boldsymbol{v}(n)$ while keeping the mathematical problem tractable. It has been previously employed with success in analyses of adaptive algorithms with weight updates that are nonlinear with respect to the weight vector [34].

The simulation results will show that assumptions A1 and A2 and approximation A3 lead to analytical models which are accurate enough in predicting the behavior of the algorithms for design purposes.

The excess means square estimation error (EMSE) is given by $\zeta(n) = E\left\{\tilde{\boldsymbol{v}}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\tilde{\boldsymbol{v}}(n)\right\}$. Using the relation between $\tilde{\boldsymbol{v}}(n)$ and $\boldsymbol{v}(n)$, the properties of $\boldsymbol{\xi}(n)$, and noting from (24), (33) and (36) that $\boldsymbol{v}(n)$ and $\boldsymbol{\xi}(n)$ are independent, we can write $\zeta(n)$ as

$$\begin{aligned}\zeta(n) &= E\left\{(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))^\top \boldsymbol{x}(n)\boldsymbol{x}^\top(n)(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))\right\}\\ &= \operatorname{trace}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\} + \operatorname{trace}\{\boldsymbol{R}_x\, \boldsymbol{\Xi}\} \qquad (43)\end{aligned}$$

with $\boldsymbol{K}(n) = E\left\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\right\}$. The term $\operatorname{trace}\{\boldsymbol{R}_x\, \boldsymbol{\Xi}\}$ is the contribution of the random non-stationarity of the system to the EMSE. In the following, we derive recursive models for $\boldsymbol{K}(n)$ for each of the algorithms.

A. Normalized NNLMS Algorithm

Post-multiplying (24) by its transpose, taking the expectation, using the approximation $\boldsymbol{x}^\top(n)\boldsymbol{x}(n) \approx N\sigma_x^2$ in (24), defining $\tilde{\eta} = \eta/(N\sigma_x^2 + \epsilon)$, using A1–A3 and proceeding as in [20] leads to

$$\begin{aligned}\boldsymbol{K}_{\mathrm{N}}(n+1) &= \boldsymbol{K}_{\mathrm{N}}(n)\\ &- \tilde{\eta}\Big(\boldsymbol{P}_{1_{\mathrm{N}}}(n)\boldsymbol{K}_{\mathrm{N}}(n) + \boldsymbol{K}_{\mathrm{N}}(n)\boldsymbol{P}_{1_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{5_{\mathrm{N}}}(n) + \boldsymbol{P}_{5_{\mathrm{N}}}^\top(n)\Big)\\ &+ \tilde{\eta}^2\,\Big(\boldsymbol{P}_{6_{\mathrm{N}}}(n) + \boldsymbol{P}_{7_{\mathrm{N}}}(n) + \boldsymbol{P}_{7_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{8_{\mathrm{N}}}(n)\Big)\\ &+ \tilde{\eta}^2\sigma_z^2\,\Big(\boldsymbol{P}_{2_{\mathrm{N}}}(n) + \boldsymbol{P}_{3_{\mathrm{N}}}(n) + \boldsymbol{P}_{3_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{4_{\mathrm{N}}}(n)\Big)\\ &+ \tilde{\eta}^2\,(\boldsymbol{P}_{9_{\mathrm{N}}}(n) + \boldsymbol{P}_{10_{\mathrm{N}}}(n) + \boldsymbol{P}_{11_{\mathrm{N}}}(n) + \boldsymbol{P}_{11_{\mathrm{N}}}^\top(n))\\ &+ \boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{N}}}(n) \qquad (44)\end{aligned}$$

with

$$\boldsymbol{P}_{1_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_\circ^*(n)\,\boldsymbol{x}(n)\} = \boldsymbol{D}_{\alpha_\circ^*}(n)\,\boldsymbol{R}_x. \qquad (45)$$

$$\begin{aligned}\boldsymbol{P}_{2_{\mathrm{N}}}(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_\circ^*(n)\,\boldsymbol{\alpha}_\circ^*(n)^\top\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{D}_{\alpha_\circ^*}(n)\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha_\circ^*}(n). \qquad (46)\end{aligned}$$

$$\begin{aligned}\boldsymbol{P}_{3_{\mathrm{N}}}(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{\alpha}_\circ^*(n)^\top\boldsymbol{D}_x(n)\}\\ &\approx E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha_\circ^*}(n). \qquad (47)\end{aligned}$$

$$\boldsymbol{P}_{4_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\} \approx \boldsymbol{R}_x \circ \boldsymbol{K}_{\mathrm{N}}(n) \qquad (48)$$

where $\circ$ denotes the so-called Hadamard entry-wise product,

$$\begin{aligned}&\boldsymbol{P}_{5_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{K}_{\mathrm{N}}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}. \qquad (49)\end{aligned}$$

$$\begin{aligned}&\boldsymbol{P}_{6_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_\circ^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{\alpha}_\circ^*(n)^\top\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{D}_{\alpha_\circ^*}(n)\,\boldsymbol{Q}_{\mathrm{N}}(n)\,\boldsymbol{D}_{\alpha_\circ^*}(n) \qquad (50)\end{aligned}$$

where the matrix $\boldsymbol{Q}_{\mathrm{N}}(n)$ is defined by $\boldsymbol{Q}_{\mathrm{N}}(n) = \boldsymbol{D}_{\alpha_\circ^*}(n)\,(2\,\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\boldsymbol{R}_x + \operatorname{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\}\,\boldsymbol{R}_x)$,

$$\begin{aligned}&\boldsymbol{P}_{7_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}_\circ^*(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{D}_{\alpha_\circ^*}(n)\boldsymbol{Q}_{\mathrm{N}}(n)E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}. \qquad (51)\end{aligned}$$

$$\begin{aligned}&\boldsymbol{P}_{8_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{D}_x(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{Q}_{\mathrm{N}}(n) \circ \boldsymbol{K}_{\mathrm{N}}(n). \qquad (52)\end{aligned}$$

$$\begin{aligned}&\boldsymbol{P}_{9_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\\ &\qquad \cdot\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{D}_x(n)\} \qquad (53)\end{aligned}$$

$$\begin{aligned}&\boldsymbol{P}_{10_{\mathrm{N}}}(n)\\ &= E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\\ &\qquad \cdot\boldsymbol{\alpha}_\circ^*(n)\boldsymbol{\alpha}_\circ^{*\top}(n)\boldsymbol{D}_x(n)\} \qquad (54)\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{P}_{11_{\mathrm{N}}}(n) = E\{&\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\\ &\cdot\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{\alpha}_\circ^{*\top}(n)\boldsymbol{D}_x(n)\}. \qquad (55)\end{aligned}$$

In obtaining (44), it was considered that the products of the last two term of (24) by the other terms lead to zero mean values due to the properties of $\boldsymbol{\xi}(n)$.

Expected values $\boldsymbol{P}_{1_{\mathrm{N}}}(n)$ through $\boldsymbol{P}_{8_{\mathrm{N}}}(n)$ correspond to the terms of the weight error vector recursive equation derived for the NNLMS algorithm in [20] with $\tilde{\eta}$ substituted for η. Thus, we use the results from [20] and indicate their values directly in (45) through (52). We now derive expressions for $\boldsymbol{P}_{9_{\mathrm{N}}}(n)$ through $\boldsymbol{P}_{11_{\mathrm{N}}}(n)$. These terms convey the effect of the random part of the environment non-stationarity.

Computing (i,j)th entry of $\boldsymbol{P}_{9_{\mathrm{N}}}(n)$ yields

$$\begin{aligned}&[\boldsymbol{P}_{9_{\mathrm{N}}}]_{ij}(n)\\ &= E\left\{\sum_k\sum_l \xi_k(n)\xi_l(n)x_k(n)x_l(n)x_i(n)v_{\mathrm{N}_i}(n)v_{\mathrm{N}_j}(n)x_j(n)\right\}\\ &= \sum_k\sum_l E\left\{\xi_k(n)\xi_l(n)\right\}\, E\left\{v_{\mathrm{N}_i}(n)v_{\mathrm{N}_j}(n)\right\}\\ &\quad\cdot E\left\{x_k(n)x_l(n)x_i(n)x_j(n)\right\}. \qquad (56)\end{aligned}$$

As $E\left\{\xi_k(n)\xi_l(n)\right\} \neq 0$ only for $k = l$, $[\boldsymbol{P}_{9_{\mathrm{N}}}]_{ij}(n) = \sigma_\xi^2\sum_k[\boldsymbol{K}_{\mathrm{N}}(n)]_{ij}E\left\{x_k^2(n)x_i(n)x_j(n)\right\}$. Using the Gaussian moment factorizing theorem yields $\sum_k E\left\{x_k^2(n)x_i(n)x_j(n)\right\} = \left([\boldsymbol{R}_x]_{ij}[\boldsymbol{R}_x]_{kk} + [\boldsymbol{R}_x]_{ik}[\boldsymbol{R}_x]_{jk}\right) = \left[\boldsymbol{R}_x\operatorname{trace}\{\boldsymbol{R}_x\} + 2\boldsymbol{R}_x\boldsymbol{R}_x\right]_{ij}$. This enables us to write the result in matrix form

$$\boldsymbol{P}_{9_{\mathrm{N}}}(n) = \sigma_\xi^2\,\boldsymbol{K}_{\mathrm{N}}(n) \circ (\boldsymbol{R}_x\,\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x)\,. \qquad (57)$$

Similarly, we have

$$\boldsymbol{P}_{10_{\mathrm{N}}}(n) = \sigma_\xi^2\,(\boldsymbol{\alpha}_\circ^*(n)\,\boldsymbol{\alpha}_\circ^{*\top}(n)) \circ (\boldsymbol{R}_x\,\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \qquad (58)$$

$$\begin{aligned}\boldsymbol{P}_{11_{\mathrm{N}}}(n) &= \sigma_\xi^2\,(E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\boldsymbol{\alpha}_\circ^{*\top}(n))\\ &\qquad \circ (\boldsymbol{R}_x\,\operatorname{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x). \qquad (59)\end{aligned}$$

The last term $\boldsymbol{K}_{\boldsymbol{\Delta}_{\rm N}}(n)$ conveys the effect of deterministic variation of the mean of system weights. Observing the terms multiplied with $\boldsymbol{\Delta}_{\rm N}(n)$, we have

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{\Delta}_{\rm N}}(n) =& \boldsymbol{\Delta}_{\rm N}(n)\boldsymbol{\Delta}_{\rm N}^\top(n) - \boldsymbol{\Delta}_{\rm N}(n)(E\{\boldsymbol{v}_{\rm N}(n+1)\} + \boldsymbol{\Delta}_{\rm N}(n))^\top \\ & - (E\{\boldsymbol{v}_{\rm N}(n+1)\} + \boldsymbol{\Delta}_{\rm N}(n))\boldsymbol{\Delta}_{\rm N}^\top(n) \\ =& - \boldsymbol{\Delta}_{\rm N}(n)\boldsymbol{\Delta}_{\rm N}^\top(n) - \boldsymbol{\Delta}_{\rm N}(n)E\{\boldsymbol{v}_{\rm N}(n+1)\}^\top \\ & - E\{\boldsymbol{v}_{\rm N}(n+1)\}\boldsymbol{\Delta}_{\rm N}^\top(n). \end{aligned} \tag{60}$$

B. Exponential NNLMS Algorithm

The second order moment analysis of the Exponential NNLMS algorithm requires an improvement on approximation (34) for the nonlinearity $(\boldsymbol{v}_{\rm E}(n) + \boldsymbol{\alpha}_{\circ}^*(n))^{(\gamma)}$ in (33). We use instead the following first order approximation for the real-valued solution of $(v_{{\rm E}_i}(n) + \alpha_{\circ_i}^*(n))^\gamma$:

$$\begin{aligned} & v_{{\rm E}_i}(n) + \alpha_{\circ_i}^*(n))^\gamma \\ & \quad \approx (E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^\gamma + \gamma\, g(E\{v_{{\rm E}_i}(n)\}) \\ & \quad \cdot (E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^{\gamma-1}\,(v_{{\rm E}_i}(n) - E\{v_{{\rm E}_i}(n)\}) \end{aligned} \tag{61}$$

where

$$\begin{aligned} g(E\{v_{{\rm E}_i}(n)\}) = 1 - \Big[& u\big(E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n) + \varepsilon\big) \\ & -u\big(E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n) - \varepsilon\big)\Big] \end{aligned} \tag{62}$$

with $u(\cdot)$ being the unit step function and ε a small constant. The reason to include the gate function g about $E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n)$ in the regular Taylor series is that the derivative of $(v_{{\rm E}_i}(n) + \alpha_{\circ_i}^*(n))^\gamma$ tends to infinity if $v_{{\rm E}_i}(n)$ approaches $-\alpha_{\circ_i}^*(n)$. It is simple to verify that $\lim_{v_{{\rm E}_i}(n)\rightarrow -\alpha_{\circ_i}^*(n)} g(E\{v_{{\rm E}_i}(n)\})(E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^{\gamma-1} = 0$. The zero-th order approximation is sufficient about the point where the function is equal to zero. With this new approximation, the term on $v_{{\rm E}_i}(n)$ in (61) will include moments of the weight error vector which are necessary to proper modeling its fluctuations.

To use vector notation, we define two deterministic vectors $\boldsymbol{r}(n)$ and $\boldsymbol{s}(n)$ whose ith entries are respectively

$$\begin{aligned} r_i =& (E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^\gamma \\ & - \gamma g(E\{v_{{\rm E}_i}(n)\})(E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^{\gamma-1} E\{v_{{\rm E}_i}(n)\} \\ s_i =& \gamma\, g(E\{v_{{\rm E}_i}(n)\})(E\{v_{{\rm E}_i}(n)\} + \alpha_{\circ_i}^*(n))^{\gamma-1} \end{aligned}$$

respectively. We define also the corresponding diagonal matrices $\boldsymbol{D}_r(n) = {\rm diag}\{\boldsymbol{r}(n)\}$ and $\boldsymbol{D}_s(n) = {\rm diag}\{\boldsymbol{s}(n)\}$. With these new definitions, the linear approximation can be written in vector form as

$$(\boldsymbol{v}_{\rm E}(n) + \boldsymbol{\alpha}_{\circ}^*(n))^{(\gamma)} \approx \boldsymbol{r}(n) + \boldsymbol{D}_s(n)\boldsymbol{v}_{\rm E}(n). \tag{63}$$

Post-multiplying (33) by its transpose, using (63), taking the expected value, using A1–A3 and defining matrix $\boldsymbol{D}_{v_{\rm E}}(n) = {\rm diag}\{\boldsymbol{v}_{\rm E}(n)\}$ yields, after simple algebraic manipulations as done in [20]

$$\begin{aligned} & \boldsymbol{K}_{\rm E}(n+1) \\ & = \boldsymbol{K}_{\rm E}(n) - \eta\,(\boldsymbol{P}_{1_{\rm E}}(n)\,\boldsymbol{K}_{\rm E}(n) + \boldsymbol{K}_{\rm E}(n)\,\boldsymbol{P}_{1_{\rm E}}^\top(n)) \\ & \quad - \eta\,(\boldsymbol{P}_{5_{\rm E}}(n)\boldsymbol{K}_{\rm E}(n) + \boldsymbol{K}_{\rm E}(n)\boldsymbol{P}_{5_{\rm E}}^\top(n)) \\ & \quad + \eta^2\,(\boldsymbol{P}_{6_{\rm E}}(n) + \boldsymbol{P}_{7_{\rm E}}(n) + \boldsymbol{P}_{7_{\rm E}}^\top(n) + \boldsymbol{P}_{8_{\rm E}}(n)) \\ & \quad + \eta^2\sigma_z^2(\boldsymbol{P}_{2_{\rm E}}(n) + \boldsymbol{P}_{3_{\rm E}}(n) + \boldsymbol{P}_{3_{\rm E}}^\top(n) + \boldsymbol{P}_{4_{\rm E}}(n)) \\ & \quad + \eta^2\boldsymbol{P}_{9_{\rm E}}(n) + \boldsymbol{K}_{\boldsymbol{\Delta}_{\rm E}}(n) \end{aligned} \tag{64}$$

with the eight moments $\boldsymbol{P}_{1_{\rm E}}(n) - \boldsymbol{P}_{8_{\rm E}}(n)$ given by

$$\boldsymbol{P}_{1_{\rm E}}(n) = E\Big\{\boldsymbol{D}_x(n)\boldsymbol{r}(n)\boldsymbol{x}^\top(n)\Big\} = \boldsymbol{D}_r(n)\boldsymbol{R}_x. \tag{65}$$

$$\boldsymbol{P}_{2_{\rm E}}(n) = E\{\boldsymbol{D}_x(n)\boldsymbol{r}(n)\boldsymbol{r}^\top(n)\boldsymbol{D}_x(n)\} \approx \boldsymbol{D}_r(n)\boldsymbol{R}_x\boldsymbol{D}_r(n). \tag{66}$$

$$\begin{aligned} \boldsymbol{P}_{3_{\rm E}}(n) &= E\{\boldsymbol{D}_x(n)\boldsymbol{r}(n)\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{D}_s(n)\boldsymbol{D}_x(n)\} \\ &\approx \boldsymbol{D}_r(n)\boldsymbol{R}_x E\{\boldsymbol{D}_{v_{\rm E}}(n)\}\boldsymbol{D}_s(n) \end{aligned} \tag{67}$$

$$\begin{aligned} \boldsymbol{P}_{4_{\rm E}}(n) &= E\{\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\boldsymbol{v}_{\rm E}(n)\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{D}_s(n)\boldsymbol{D}_x(n)\} \\ &\approx \boldsymbol{D}_s(n)(\boldsymbol{R}_x \circ \boldsymbol{K}_{\rm E}(n))\boldsymbol{D}_s(n). \end{aligned} \tag{68}$$

$$\begin{aligned} \boldsymbol{P}_{5_{\rm E}}(n) &= \{\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{D}_s(n)\} \\ &= {\rm diag}\Big\{\boldsymbol{R}_x E\{\boldsymbol{v}_{\rm E}(n)\}\Big\}\boldsymbol{D}_s(n) \end{aligned} \tag{69}$$

$$\begin{aligned} \boldsymbol{P}_{6_{\rm E}}(n) &= E\{\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)\boldsymbol{r}^\top(n)\boldsymbol{D}_x(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\rm E}(n)\} \\ &\approx \boldsymbol{D}_r(n)\boldsymbol{Q}_{\rm E}(n)\boldsymbol{D}_r(n) \end{aligned} \tag{70}$$

where the matrix $\boldsymbol{Q}_{\rm E}(n)$ in above equations is defined by $\boldsymbol{Q}_{\rm E}(n) = 2\,\boldsymbol{R}_x\boldsymbol{K}_{\rm E}(n)\boldsymbol{R}_x + {\rm trace}\{\boldsymbol{R}_x\boldsymbol{K}_{\rm E}(n)\}\,\boldsymbol{R}_x$

$$\begin{aligned} & \boldsymbol{P}_{7_{\rm E}}(n) \\ & = E\{\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{r}(n)\boldsymbol{v}_{\rm E}^\top(n)\boldsymbol{D}_s(n)\boldsymbol{D}_x(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\rm E}(n)\} \\ & \approx \boldsymbol{D}_r(n)\,\boldsymbol{Q}_{\rm E}(n)\,E\{\boldsymbol{D}_{v_{\rm E}}(n)\}\,\boldsymbol{D}_s(n) \end{aligned} \tag{71}$$

and

$$\begin{aligned} \boldsymbol{P}_{8_{\rm E}}(n) = & E\{\boldsymbol{v}_{\rm E}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\boldsymbol{v}_{\rm E}(n)\,\boldsymbol{v}_{\rm E}^\top(n) \\ & \cdot \boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\rm E}(n)\} \\ & \approx \boldsymbol{D}_s(n)\,\Big(\boldsymbol{Q}_{\rm E}(n) \circ \boldsymbol{K}_{\rm E}(n)\Big)\,\boldsymbol{D}_s(n). \end{aligned} \tag{72}$$

The expectation $\boldsymbol{P}_{9_{\rm E}}(n)$ conveys the non-stationarity effects and is given by

$$\begin{aligned} \boldsymbol{P}_{9_{\rm E}}(n) = E\Big\{ & (\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n))^2\,(\boldsymbol{v}_{\rm E}(n) + \boldsymbol{\alpha}_{\circ}^*(n))^{(\gamma)} \\ & \cdot(\boldsymbol{v}_{\rm E}(n) + \boldsymbol{\alpha}_{\circ}^*(n))^{(\gamma)\top}\Big\}. \end{aligned} \tag{73}$$

Using the first order approximation (63) and simple manipulations yields

$$\begin{aligned} & \boldsymbol{P}_{9_{\rm E}}(n) \\ & = {\rm trace}\{\boldsymbol{\Xi}\boldsymbol{R}_x\}\Big\{\boldsymbol{r}(n)\boldsymbol{r}^\top(n) + \boldsymbol{D}_s(n)\boldsymbol{K}_{\rm E}(n)\boldsymbol{D}_s(n) \\ & \quad + \boldsymbol{r}(n)E\{\boldsymbol{v}_{\rm E}^\top(n)\}\boldsymbol{D}_s(n) + \boldsymbol{D}_s(n)E\{\boldsymbol{v}_{\rm E}(n)\}\boldsymbol{r}^\top(n)\Big\}. \end{aligned} \tag{74}$$

The last term $\boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{E}}}(n)$ is obtained in the same form of (60)

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{E}}}(n) = -\boldsymbol{\Delta}_{\mathrm{E}}(n)\boldsymbol{\Delta}_{\mathrm{E}}^{\top}(n) - \boldsymbol{\Delta}_{\mathrm{E}}(n)E\{\boldsymbol{v}_{\mathrm{E}}(n+1)\}^{\top}\\ -E\{\boldsymbol{v}_{\mathrm{E}}(n+1)\}\boldsymbol{\Delta}_{\mathrm{E}}^{\top}(n). \quad (75)\end{aligned}$$

C. Sign-Sign NNLMS Algorithm

Using the weight error vector definition $\boldsymbol{v}_{\mathrm{S}}(n) = \boldsymbol{\alpha}_{\mathrm{S}}(n) - \boldsymbol{\alpha}_{\circ}^{*}(n)$ in (19) and $\mathrm{sgn}\{e(n)\}\mathrm{sgn}\{\boldsymbol{D}_x(n)\} = \mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}$ yields

$$\begin{aligned}\boldsymbol{v}_{\mathrm{S}}(n+1) = \boldsymbol{v}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{v}_{\mathrm{S}}(n)\\ +\eta\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{\alpha}_{\circ}^{*}(n) - \boldsymbol{\Delta}_{\mathrm{S}}(n). \quad (76)\end{aligned}$$

Post-multiplying (76) by its transpose, taking the expected value and rearranging the terms leads to

$$\begin{aligned}&\boldsymbol{K}_{\mathrm{S}}(n+1)\\ &= \boldsymbol{K}_{\mathrm{S}}(n) + \eta\,(\boldsymbol{P}_{1_{\mathrm{S}}}(n) + \boldsymbol{P}_{1_{\mathrm{S}}}^{\top}(n)) + \eta\,(\boldsymbol{P}_{2_{\mathrm{S}}}(n) + \boldsymbol{P}_{2_{\mathrm{S}}}^{\top}(n))\\ &\quad + \eta^2\,(\boldsymbol{P}_{3_{\mathrm{S}}}(n) + \boldsymbol{P}_{4_{\mathrm{S}}}(n) + \boldsymbol{P}_{4_{\mathrm{S}}}^{\top}(n) + \boldsymbol{P}_{5_{\mathrm{S}}}(n)) + \boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{S}}(n)}\end{aligned} \quad (77)$$

where

$$\boldsymbol{P}_{1_{\mathrm{S}}}(n) = E\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{\alpha}_{\circ}^{*\top}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \quad (78)$$

$$\boldsymbol{P}_{2_{\mathrm{S}}}(n) = E\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \quad (79)$$

$$\begin{aligned}\boldsymbol{P}_{3_{\mathrm{S}}}(n) &= E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\boldsymbol{\alpha}_{\circ}^{*}(n)\boldsymbol{\alpha}_{\circ}^{*\top}\\ &\qquad\cdot\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\}\end{aligned} \quad (80)$$

$$\begin{aligned}\boldsymbol{P}_{4_{\mathrm{S}}}(n) &= E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\boldsymbol{\alpha}_{\circ}^{*}(n)\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\\ &\qquad\cdot\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\}\end{aligned} \quad (81)$$

$$\begin{aligned}\boldsymbol{P}_{5_{\mathrm{S}}}(n) &= E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\boldsymbol{v}(n)\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\\ &\qquad\cdot\mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}\}.\end{aligned} \quad (82)$$

These expected values are calculated in the following for $\boldsymbol{x}(n)$ Gaussian.

Expected value $\boldsymbol{P}_{1_{\mathrm{S}}}(n)$:

Using the properties of statistical expectation $\boldsymbol{P}_{1_{\mathrm{S}}}(n)$ can be written as

$$\begin{aligned}\boldsymbol{P}_{1_{\mathrm{S}}}(n) = E_v\Big\{&\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{\alpha}_{\circ}^{*\top}(n)\\ &E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\mid\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)\}\}\Big\}. \quad (83)\end{aligned}$$

The conditional expectation in (83) is given by (39), which must be approximated. Approximation (40) for the ith element of (39) is too simple to predict the weight error fluctuations. A more suitable approximation is given by a first order Taylor series expansion:

$$\begin{aligned}&\frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^{\top}\,\boldsymbol{v}_{\mathrm{S}}(n)}{\sigma_x\sigma_{e|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)}}\right)\\ &\approx\frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^{\top}\,E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)\\ &\quad-\frac{2}{\pi}\,\frac{\boldsymbol{R}_i^{\top}(\boldsymbol{v}_{\mathrm{S}}(n)-E\{\boldsymbol{v}_{\mathrm{S}}(n)\})}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}\sqrt{1-\left(\frac{\boldsymbol{R}_i^{\top}E\{\boldsymbol{v}_{\mathrm{S}}(n)\})}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)^2}}\\ &= q_i(n) + \boldsymbol{s}_i^{\top}(n)\boldsymbol{v}_{\mathrm{S}}(n)\end{aligned} \quad (84)$$

where the scalar $q_i(n)$ and the vector $\boldsymbol{s}_i(n)$ are deterministic variables defined respectively as

$$\begin{aligned}q_i(n) &= \frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^{\top}E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)\\ &\quad+\frac{2}{\pi}\,\frac{\boldsymbol{R}_i^{\top}E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi})}\sqrt{1-\left(\frac{\boldsymbol{R}_i^{\top}E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)^2}}\end{aligned} \quad (85)$$

$$\boldsymbol{s}_i(n) = -\frac{2}{\pi}\,\frac{\boldsymbol{R}_i}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}\sqrt{1-\left(\frac{\boldsymbol{R}_i^{\top}E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)^2}} \quad (86)$$

with $\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}$ defined in (41).

Using (84) in (83) and defining the $N \times 1$ vector $\boldsymbol{q}(n)$ with ith element $q_i(n)$ and the $N \times N$ matrix $\boldsymbol{S}(n)$ with ith column $\boldsymbol{s}_i(n)$, (83) becomes, after simple manipulations,

$$\begin{aligned}\boldsymbol{P}_{1_{\mathrm{S}}}(n) \approx E\{\boldsymbol{v}_{\mathrm{S}}(n)\}\boldsymbol{\alpha}_{\circ}^{*\top}(n)\,\mathrm{diag}\{\boldsymbol{q}(n)\}\\ +\boldsymbol{K}_{\mathrm{S}}(n)\,\boldsymbol{S}(n)\,\boldsymbol{D}_{\alpha_{\circ}^{*}}(n). \quad (87)\end{aligned}$$

Expected value $\boldsymbol{P}_{2_{\mathrm{S}}}(n)$:

Similar to $\boldsymbol{P}_{1_{\mathrm{S}}}(n)$, we first express $\boldsymbol{P}_{2_{\mathrm{S}}}(n)$ in the form

$$\boldsymbol{P}_{2_{\mathrm{S}}}(n) = E_v\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\mid\boldsymbol{v}_{\mathrm{S}}(n)\}\}\}. \quad (88)$$

Then, using (39) and (84) we obtain

$$\begin{aligned}\boldsymbol{P}_{2_{\mathrm{S}}}(n) \approx \boldsymbol{K}_{\mathrm{S}}(n)\mathrm{diag}\{\boldsymbol{q}(n)\}\\ +E\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\mathrm{diag}\{\boldsymbol{S}^{\top}(n)\boldsymbol{v}_{\mathrm{S}}(n)\}\}. \quad (89)\end{aligned}$$

The (i,j)th element of the expectation in (89) is given by

$$\begin{aligned}&E\{[\boldsymbol{v}_{\mathrm{S}}(n)\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\mathrm{diag}\{\boldsymbol{S}^{\top}(n)\boldsymbol{v}(n)\}]_{ij}\}\\ &\qquad= \sum_{k=1}^{N} S_{kj}E\{v_{\mathrm{S}_k}(n)v_{\mathrm{S}_i}(n)v_{\mathrm{S}_j}(n)\}. \quad (90)\end{aligned}$$

Evaluation of the third order moment in (90) requires further approximation, as the distribution of $\boldsymbol{v}_{\mathrm{S}}(n)$ is unknown. We assume that the distribution of $\boldsymbol{v}(n)$ can be approximated by a Gaussian distribution about its mean value. Then, using the properties of Gaussian variables [35] and defining the centered variable $\bar{v}_p(n) = v_p(n) - E\{v_p(n)\}$ for $p = i, j, k$, we have

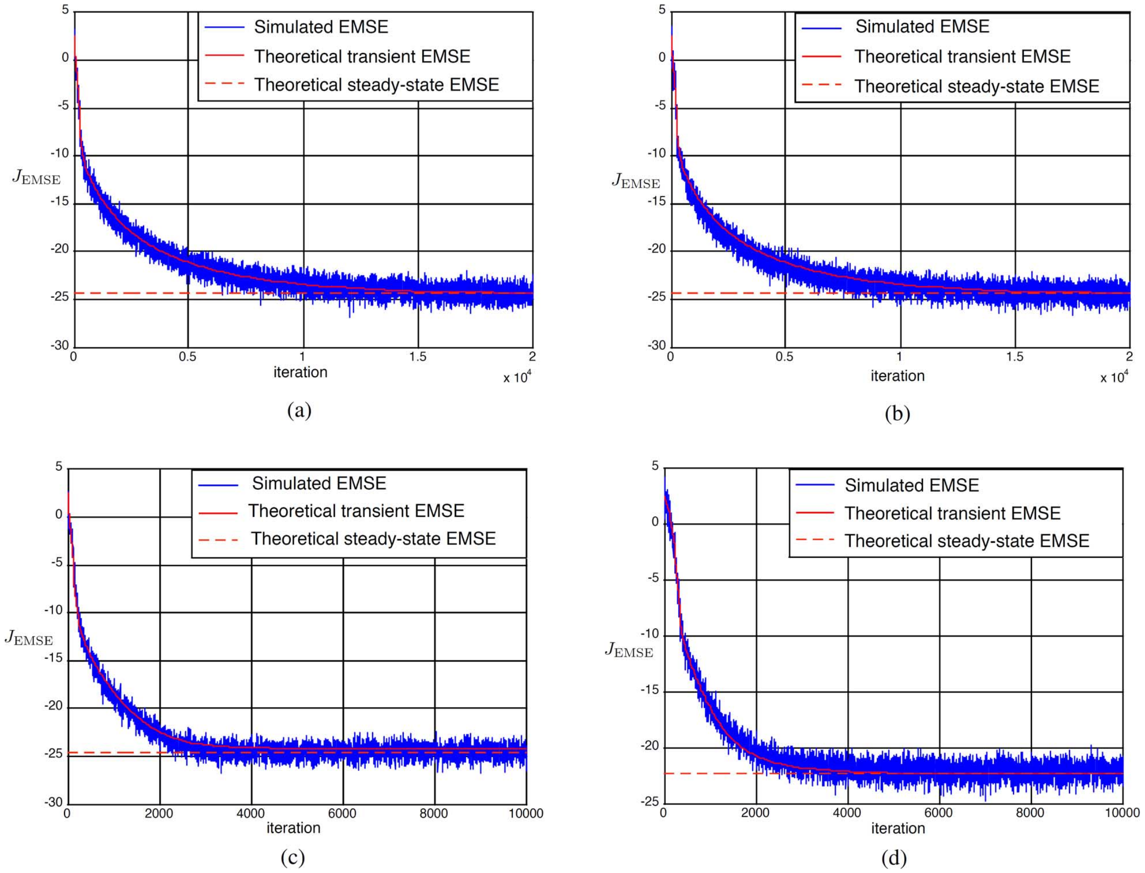
$$\begin{aligned}&E\{v_i(n)v_j(n)v_k(n)\}\\ &\approx E\{\tilde{v}_i(n)\tilde{v}_j(n)\}E\{v_k(n)\}+E\{\tilde{v}_i(n)\tilde{v}_k(n)\}E\{v_j(n)\}\\ &\quad+E\{\tilde{v}_j(n)\tilde{v}_k(n)\}E\{v_i(n)\}+E\{v_i(n)\}E\{v_j(n)\}E\{v_k(n)\}\\ &= K_{ij}(n)E\{v_k(n)\}+K_{ik}(n)E\{v_j(n)\}+K_{jk}(n)\,E\{v_i(n)\}\\ &\quad-2E\{v_i(n)\}E\{v_j(n)\}E\{v_k(n)\}\end{aligned} \quad (91)$$

which completes the derivation of (89).

Expected value $\boldsymbol{P}_{3_{\mathrm{S}}}$:

The (i,j)-th entry of matrix $\boldsymbol{P}_{3_{\mathrm{S}}}(n)$ is given by

$$\begin{aligned}&[\boldsymbol{P}_{3_{\mathrm{S}}}(n)]_{ij}\\ &= E\{\mathrm{sgn}\{x_i(n)\,e(n)\}\,[\boldsymbol{\alpha}_{\circ}^{*}(n)\,\boldsymbol{\alpha}_{\circ}^{*\top}(n)]_{ij}\,\mathrm{sgn}\{x_j(n)\,e(n)\}^{\top}\}\\ &= E\{[\boldsymbol{\alpha}_{\circ}^{*}(n)\,\boldsymbol{\alpha}_{\circ}^{*\top}(n)]_{ij}\,\mathrm{sgn}\{x_i(n)\,x_j(n)\}\}.\end{aligned} \quad (92)$$



(a)

(b)

(c)

(d)

Fig. 2. Steady-state EMSE model validation for NNLMS and its variants. (a) Original NNLMS. (b) Normalized NNLMS. (c) Exponential NNLMS. (d) Sign-Sign NNLMS.

Using (38), $E\{\mathrm{sgn}\{x_i(n)\,x_j(n)\}\} = (2/\pi)\sin^{-1}([\boldsymbol{R}_x]_{ij}/\sigma_x^2)$ and

$$[\boldsymbol{P}_{3_\mathrm{S}}(n)]_{ij} = [\boldsymbol{\alpha}^*_\circ(n)\boldsymbol{\alpha}^{*\top}_\circ(n)]_{ij}\,\frac{2}{\pi}\sin^{-1}\left(\frac{[\boldsymbol{R}_x]_{ij}}{\sigma_x^2}\right). \tag{93}$$

Finally, expressing the result in the matrix form yields

$$\boldsymbol{P}_{3_\mathrm{S}}(n) = \frac{2}{\pi}\left(\boldsymbol{\alpha}^*_\circ(n)\,\boldsymbol{\alpha}^{*\top}_\circ(n)\right)\circ\boldsymbol{T} \tag{94}$$

where the (i,j)th elements of the $N\times N$ matrix $\boldsymbol{T}$ is given by $[\boldsymbol{T}]_{ij} = \sin^{-1}([\boldsymbol{R}_x]_{ij}/\sigma_x^2)$.

Expected values $\boldsymbol{P}_{4_\mathrm{S}}(n)$ and $\boldsymbol{P}_{5_\mathrm{S}}(n)$:

Using the same reasoning and approximations used to evaluate $\boldsymbol{P}_{1_\mathrm{S}}(n)$ to $\boldsymbol{P}_{3_\mathrm{S}}(n)$ yields

$$\boldsymbol{P}_{4_\mathrm{S}}(n) = \frac{2}{\pi}(\boldsymbol{\alpha}^*_\circ(n)E\{\boldsymbol{v}_\mathrm{S}^\top(n)\})\circ\boldsymbol{T} \tag{95}$$

$$\boldsymbol{P}_{5_\mathrm{S}}(n) = \frac{2}{\pi}(\boldsymbol{K}_\mathrm{S}(n))\circ\boldsymbol{T}. \tag{96}$$

The last term $\boldsymbol{K}_{\boldsymbol{\Delta}_\mathrm{S}}(n)$ writes

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{\Delta}_\mathrm{S}}(n) = -\boldsymbol{\Delta}_\mathrm{S}(n)\boldsymbol{\Delta}_\mathrm{S}^\top(n) - \boldsymbol{\Delta}_\mathrm{S}(n)E\{\boldsymbol{v}_\mathrm{S}(n+1)\}^\top\\ -E\{\boldsymbol{v}_\mathrm{S}(n+1)\}\boldsymbol{\Delta}_\mathrm{S}^\top(n).\end{aligned} \tag{97}$$

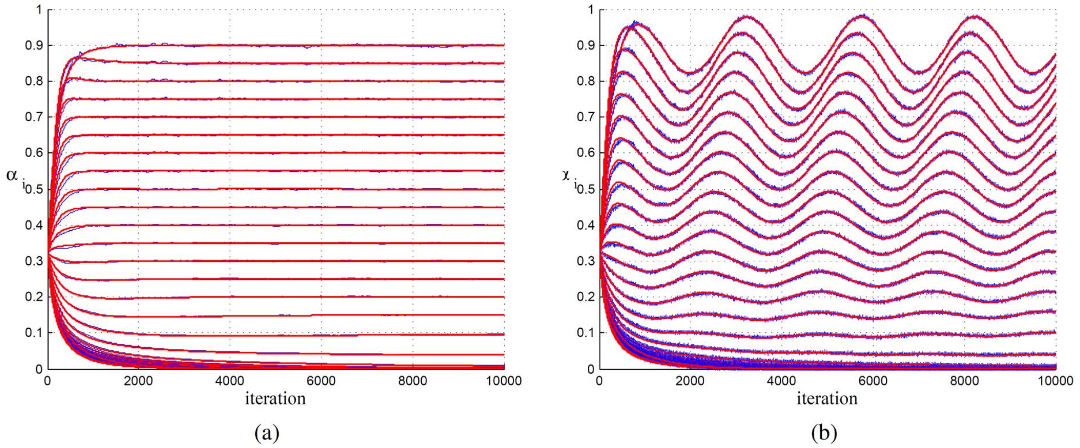
VI. Steady-State Behavior

We have derived also analytical models for the steady-state behavior of the algorithms proposed in Section III for a stationary environment. The analysis is not shown here for space limitations. The interested reader is referred to [36]. We present here simulation examples that illustrate the accuracy of the derived model.

Consider the system in Fig. 1 with an unknown system of order $N = 15$ and weights defined by

$$\begin{aligned}\boldsymbol{\alpha}^* = [0.8,\ 0.6,\ 0.5, -0.05,\ 0.4, -0.04,\ 0.3, -0.03,\\ 0.2, -0.02,\ 0.1, -0.01,\ 0,\ 0,\ 0]^\top,\end{aligned} \tag{98}$$

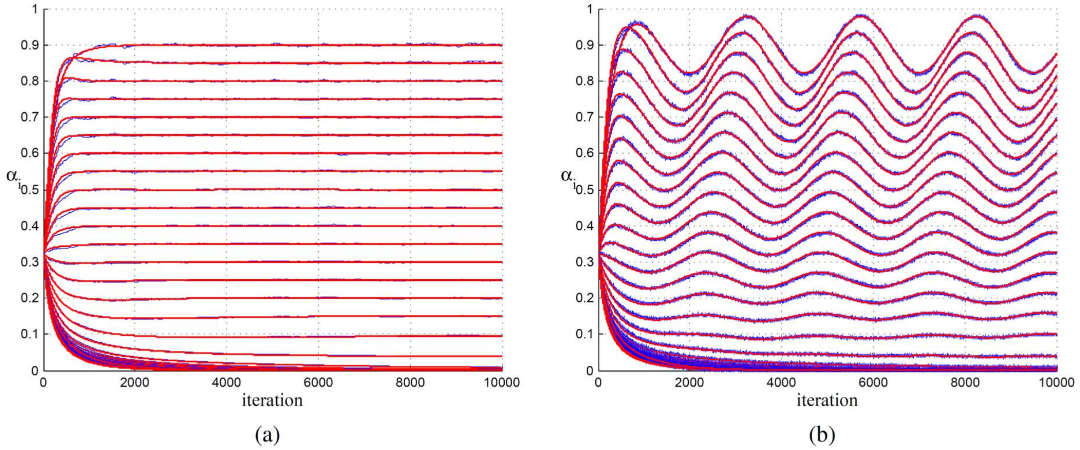
where negative coefficients were explicitly included to activate the non-negativity constraint. The input signal was the first-order AR progress given by $x(n) = 0.5x(n-1) + w(n)$, where $w(n)$ is an i.i.d. zero-mean Gaussian sequence with variance $\sigma_w^2 = 0.75$ (so that $\sigma_x^2 = 1$) and independent of any other signal. The additive independent noise $z(n)$ was zero-mean i.i.d. Gaussian with variance $\sigma_z^2 = 0.01$. The adaptive weights were initialized with $\alpha_i(0) = 0.1$ for $i = 1,\ldots,N$. The step sizes were equal to $\eta = 0.01N\sigma_x^2$ for NNLMS and $\eta = 0.01$ for the NNLMS, Exponential NNLMS and Sign-Sign NNLMS algorithms. Monte Carlo simulation results were obtained by averaging 100 runs. Fig. 2 shows the simulation results and the behavior predicted by the analytical models. The theoretical transient EMSE behaviors were obtained using results obtained in Sections IV and V and the theoretical steady-state EMSE (horizontal dashed lines) were calculated using the models derived in [36]. These figures clearly validate the proposed theoretical results.



(a)

(b)

Fig. 3. Evolution of the coefficients $\alpha_i(n)$ for the normalized NNLMS algorithm in stationary and nonstationary environments for $\sigma_x^2 = 1$. Theoretical curves are from (32): (a) stationary case, $\sigma_x^2 = 1$ and (b) nonstationary case, $\sigma_x^2 = 1$.



(a)

(b)

Fig. 4. Evolution of the coefficients $\alpha_i(n)$ for the normalized NNLMS algorithm in stationary and nonstationary environments for $\sigma_x^2 = 0.5$. Theoretical curves are from (32): (a) stationary case, $\sigma_x^2 = 0.5$ and (b) nonstationary case, $\sigma_x^2 = 0.5$.

VII. Simulation Results and Discussion

We now present simulation examples to illustrate the properties of the three algorithms and the accuracy of the models derived in Sections IV and V. The parameters for these examples were chosen to illustrate several properties of the three algorithms while conserving space. Similar results have been obtained using a variety of parameter sets. For all examples, $N = 31$. The unknown stationary system is defined as

$$\alpha^{*(\text{stat.})}_{\text{o}_i} = \begin{cases} 0.9 - 0.05\, i, & i = 0,\dots,18 \\ -0.01\,(i-18) & i = 19,\dots,31. \end{cases} \tag{99}$$

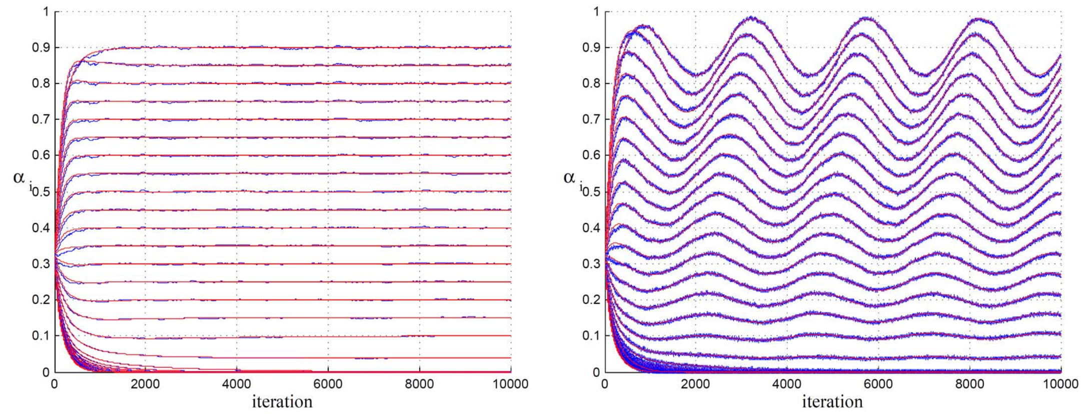
For the non-stationary case, we consider an unknown response defined by

$$\begin{aligned} &\alpha^{*(\text{nonstat.})}_{\text{o}_i}(n) \\ &= \boldsymbol{\alpha}^{*(\text{stat.})}_{\text{o}_i} + \frac{|\boldsymbol{\alpha}^{*(\text{stat.})}_{\text{o}_i}|}{10}\sin\left(\frac{2\,\pi}{T}\,n + 2\,\pi\frac{i-1}{N}\right) + \xi_i(n) \end{aligned} \tag{100}$$

where the period T of the deterministic sinusoidal component was set to 2500. $\boldsymbol{\xi}(n)$ is a zero-mean Gaussian random vector with correlation matrix $\sigma_\xi^2\boldsymbol{I}$ with $\sigma_\xi^2 = 5 \times 10^{-4}$. The input signal is a first-order AR process given by $x(n) = 0.5x(n-1) + w(n)$, with $w(n)$ i.i.d. zero-mean Gaussian with variance σ_w^2, adjusted to obtain the desired input power $\sigma_x^2 = 1$. The noise $z(n)$ is zero-mean i.i.d. Gaussian with variance $\sigma_z^2 = 10^{-2}$. The adaptive weights in $\boldsymbol{\alpha}_i(0)$ were all initialized at $10/N$ for all realizations. The step size was always set to $\eta = 0.005$ for all but the normalized variant. For the latter we used $\eta = 0.005\,N\sigma_x^2$, which leads to an equivalent step size $\tilde{\eta} = 0.005$. Monte Carlo simulations were obtained by averaging 100 runs.

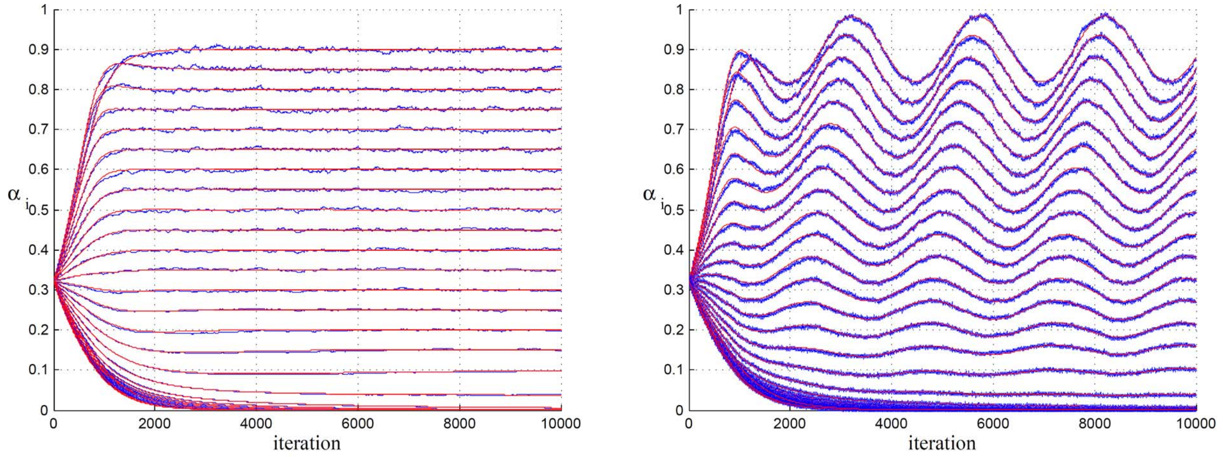
1) Example 1: Figs. 3 and 4 show the results for the Normalized NNLMS. The parameter ϵ was set to 0. Blue curves show simulation results and red curves show the theoretical predictions from (32). Fig. 3 is for $\sigma_x^2 = 1$ and Fig. 4 is for $\sigma_x^2 = 0.5$. It can be verified that the model (32) accurately predicts the algorithm behavior, and that normalization has made the algorithm performance basically independent of the input power.

2) Example 2: Fig. 5 illustrates the results for the Exponential NNLMS algorithm. The parameter $(p, q) = (5, 7)$ was used. Compared with Fig. 3, these figures clearly show that the coefficients that tend to zero in steady-state had their convergence rate significantly improved by the Exponential NNLMS algorithm. Also, the accuracy of the theoretical model (35) can be verified.



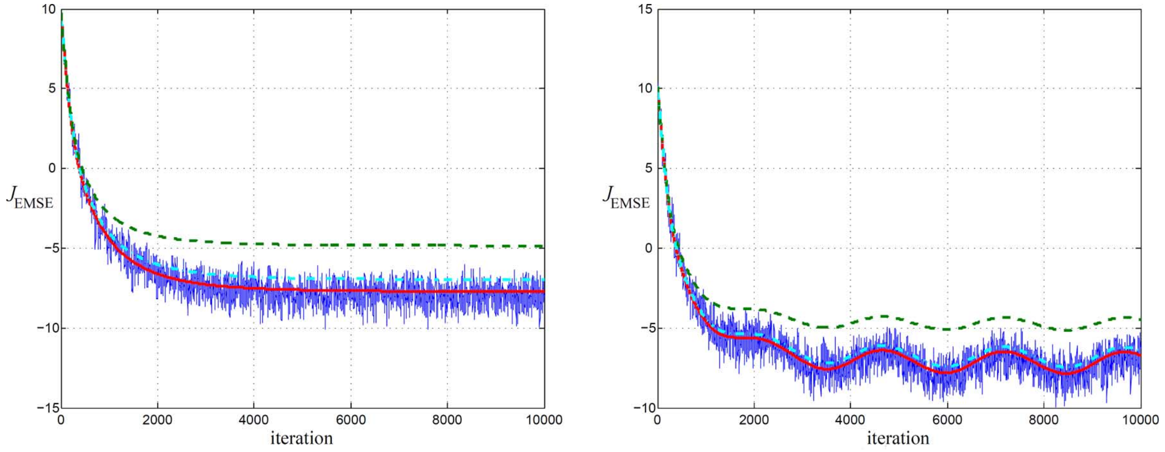
(a) (b)

Fig. 5. Evolution of the coefficients $\alpha_i(n)$ for the Exponential NNLMS algorithm in stationary and nonstationary environments. Theoretical curves are from (35): (a) stationary case and (b) nonstationary case.



(a) (b)

Fig. 6. Evolution of the coefficients $\alpha_i(n)$ for the Sign-Sign NNLMS algorithm in stationary and nonstationary environments. Theoretical curves are from (42): (a) stationary case and (b) nonstationary case.



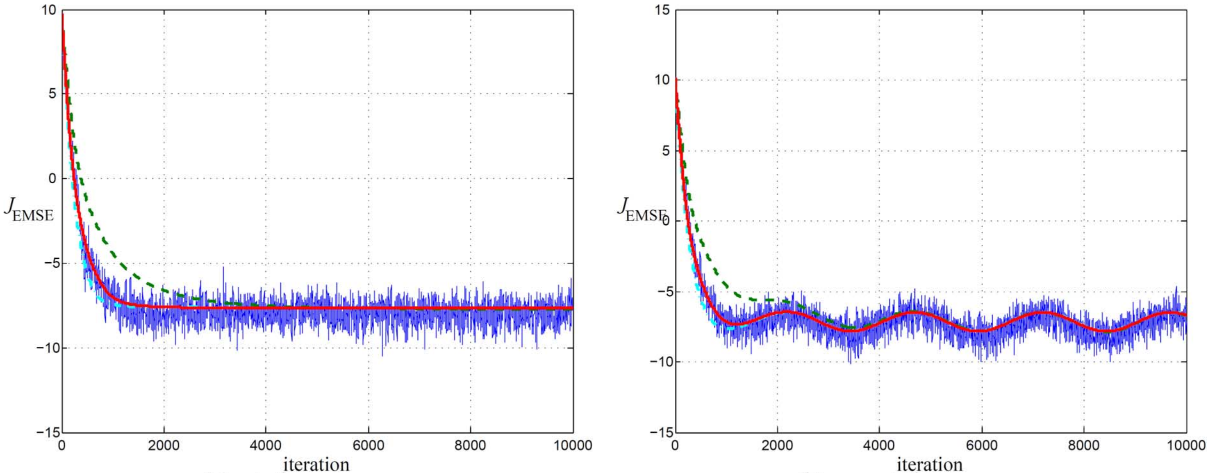
(a) (b)

Fig. 7. Evolution of the EMSE for the Normalized NNLMS algorithm in stationary and nonstationary environment. Light blue dash-dot line and green dashed line show the theoretical results for $\sigma_\xi^2 = 10^{-3}$ and 5×10^{-3}, respectively: (a) curves for the stationary case and two theoretical curves with added $\boldsymbol{\xi}(n)$ and (b) i'nonstationary case.

3) Example 3: Fig. 6 illustrate the result of the Sign-Sign NNLMS under stationary and nonstationary environment. These figures illustrate the accuracy of the model (42). It is also clear that the Sign-Sign NNLMS coefficients converge much slower than those for the NNLMS algorithm as expected.

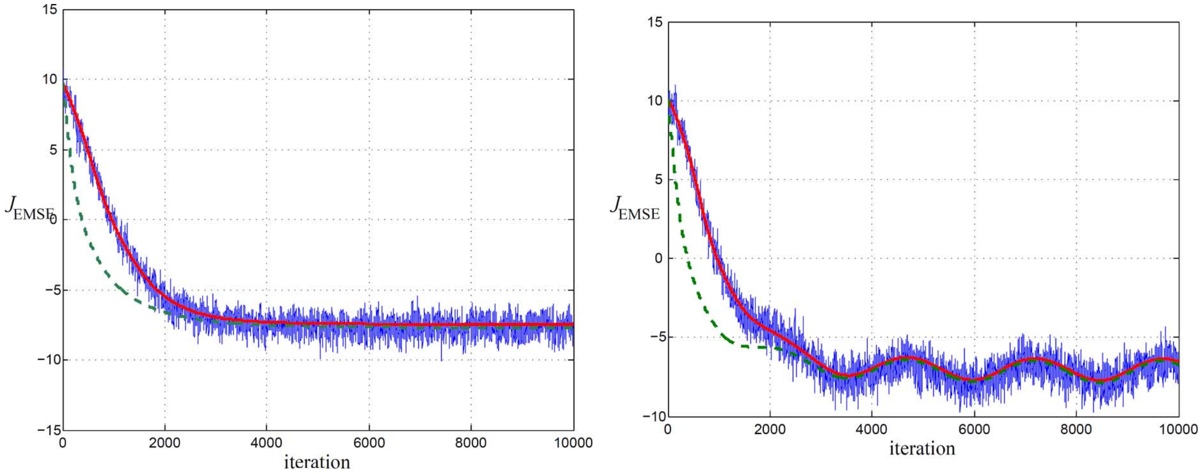
A. Second Moment Behavior

We now illustrate the EMSE behavior of the NNLMS variants. Again, blue curves were obtained from Monte Carlo simulation and red curves show the theoretical predictions.



(a) stationary case (b) nonstationary case

Fig. 8. Evolution of the EMSE for the Exponential NNLMS algorithm in stationary and nonstationary environment. Light blue dash-dot line and green dashed line show the theoretical results for $(p,q)=(3,5)$ and $(p,q)=(1,1)$ (original NNLMS), respectively: (a) stationary case and (b) nonstationary case.



(a) (b)

Fig. 9. Evolution of the EMSE for the Sign-Sign NNLMS algorithm in stationary and nonstationary environment. Theoretical evolution of original NNLMS is represented by the green dashed line: (a) stationary case and (b) nonstationary case.

1) Example 4: Fig. 7 illustrates the EMSE behavior of the Normalized NNLMS algorithm. The accuracy of model (44) can be easily verified. Two more curves are added to each plot to illustrate the effect of the random non-stationarity parameter σ_ξ^2. Random perturbations with different variances were also added to $\alpha_{\text{o}_i}^{*(\text{stat.})}$ (Fig. 7(a)) and to the nonstationary case (Fig. 7(b)). The light blue (dash-dot) lines show the theoretical EMSE for $\sigma_\xi^2 = 10^{-3}$, while the green (dash) lines show the theoretical EMSE for $\sigma_\xi^2 = 5\times 10^{-3}$. These curves illustrate the expected extra EMSE due to tracking of the random optimal solution variations. Simulation curves coincide with the theoretical ones, but are not shown to preserve the visibility of the other curves.

2) Example 5: Fig. 8 illustrates the EMSE behavior of the Exponential NNLMS algorithm. The blue and red curves show again the simulation results and the accurate theoretical predictions using (64) for $(p,q)=(5,7)$. The light blue (dash-dot) and the green (dash) curves show the theoretical predictions of the EMSE behavior for $(p,q)=(3,5)$ and $(p,q)=(1,1)$ (original NNLMS), respectively. The simulation results agree with these curves but are not shown for clarity. These results confirm that the Exponential NNLMS algorithm accelerates the convergence of the adaptive weights when compared to NNLMS.

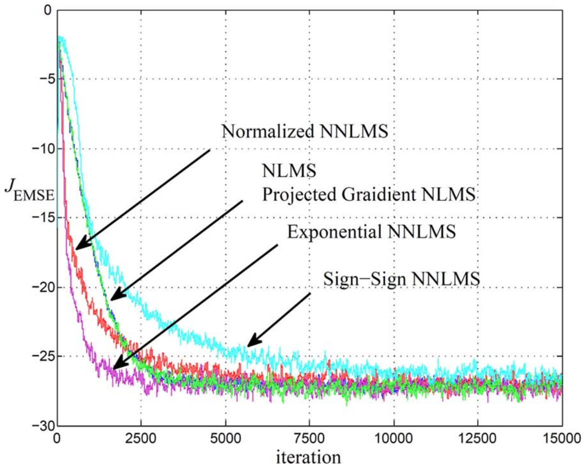
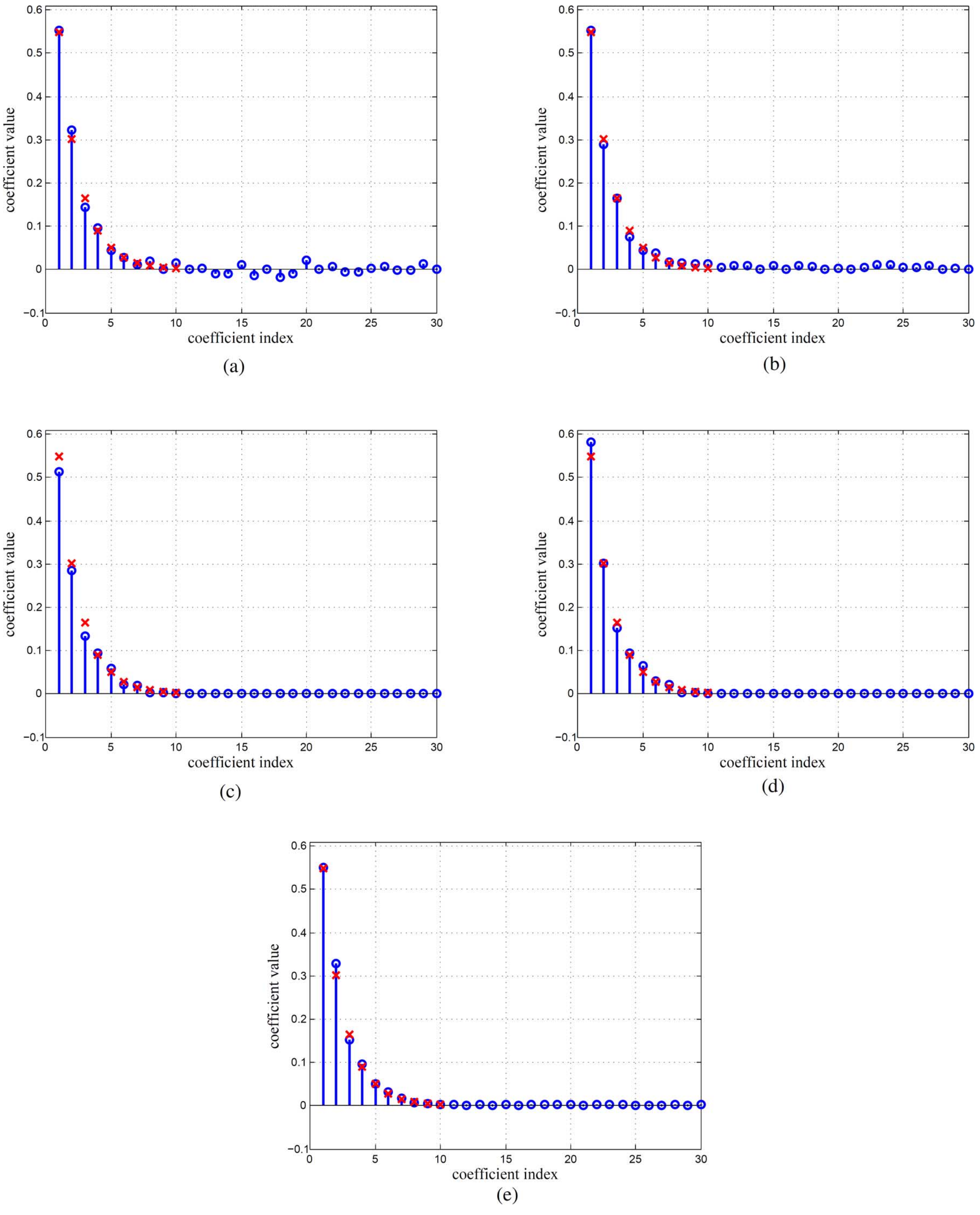


Fig. 10. EMSE (db) for the four algorithms compared.

3) Example 6: Fig. 9 illustrates EMSE behavior of the Sign-Sign NNLMS algorithm. Once more, the red curves and blue curves illustrate the accuracy of the model (77). The



(a) (b) (c) (d) (e)

Fig. 11. Weights estimated by NLMS, Projected NLMS, Normalized NNLMS, and Exponential NNLMS at $n = 15000$ for a single realization. Real weights are marked by ×. The NNLMS variants determine clearly the support of the response: (a) NLMS algorithm; (b) projected NLMS algorithm; (c) normalized NNLMS algorithm; (d) exponential NNLMS algorithm; and (e) sign-sign NNLMS algorithm.

green (dashed) curves show the performance of the original NNLMS in the same conditions. These curves illustrate the slower convergence rate of Sign-Sign NNLMS when compared to NNLMS, the price paid for a reduced computational complexity.

B. A Comparative Example

This example compares the performance of the NNLMS algorithm and its variants with that of unconstrained algorithms in solving the unconstrained solution problem of identifying an unknown weight vector $\boldsymbol{\alpha}^*$ with non-negative coefficients. This is an interesting application, as in this case the unconstrained algorithm will converge in the mean to the optimal solution. Though the problem description may guarantee that the optimal weights are positive, often in practice one do not have accurate information about the number of coefficients in the optimal solution. A common approach is to set the adaptive filter with a sufficient number of coefficients, usually larger than the actual unknown number. This examples illustrates the performance of the different algorithms in this case.

Consider a non-negative unknown optimal solution

$$\alpha_i^* = \exp(-0.6\, i) \tag{101}$$

with $i = 1, \ldots, 10$ and adaptive filters with $N = 30$ coefficients. Five algorithms were tested: NLMS [26], Projected Gradient NLMS [37], Normalized NNLMS, Exponential NNLMS and Sign-Sign NNLMS. In Projected Gradient NLMS, the coefficients which activate the non-negativity constraints are projected into the feasible region, i.e., set to 0, at each iteration. The input signal was given by $x(n) = 0.5x(n-1) + w(n)$ with $\sigma_w^2 = 3/4$ so that $\sigma_x^2 = 1$. The initial weights $\boldsymbol{\alpha}(0)$ were drawn from the uniform distribution $\mathcal{U}(0,1)$. The additive noise $z(n)$ was i.i.d. Gaussian with $\sigma_z^2 = 0.1$. The step sizes were chosen for each algorithm by experimentation so that all would reach approximately the same steady-state EMSE with the value of 2×10^{-3}. The step sizes were $\eta = 0.035$ for both NLMS and Projected Gradient NLMS, $\eta = 0.8750$ for Normalized NNLMS, $\eta = 0.022$ for Exponential NNLMS and $\eta = 0.007$ for Sign-Sign NNLMS. Fig. 10 shows the EMSE evolution for the five algorithms (Monte Carlo simulation averaged over 100 realizations). Fig. 11 shows the estimated weights for a single realization of the input signal at $n = 15000$. Although the unconstrained NLMS algorithm is able to converge to the optimal solution in the mean sense, it does not provide a good estimation of the zero-valued coefficients in a single realization. NNLMS-type algorithms, including the Sign-Sign algorithm (which has not even converged to the steady-state at $n = 15000$) do a better job in determining the support of the actual response.

VIII. Conclusion

Many real-life systems require non-negative coefficients when their physical behavior is parameterized. In such cases, a non-negativity constraint should be imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. The Non-Negative Least-Mean-Square (NNLMS) algorithm has been recently proposed to solve such a constrained Wiener problem online. In this paper, we proposed three variants of NNLMS, each addressing a different issue that may affect NNLMS under given circumstances. The performances of the Normalized NNLMS, Exponential NNLMS and Sign-Sign NNLMS algorithms were studied for nonstationary environments. The optimal unconstrained solution was modeled by a time-variant mean plus a random fluctuation. The derived analytical models were shown to accurately predict both the mean and the mean-square behavior of the algorithms. Their performances were compared and their advantages in potential applications discussed. Future research efforts will further explore these NNLMS variants properties and apply them in practical situations where efficient adaptive solutions to non-negative filtering problem are required.

References

[1] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero, "Nonnegative leastsquares image deblurring: improved gradient projection approaches," *Inverse Probl.*, vol. 26, no. 1, Feb. 2010.

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

[3] A. Cont and S. Dubinov, "Realtime multiple pitch and multipleinstrument recognition for music signals unsing sparse non-negative constraints," in *Proc. DAFx*, Bordeaux, France, Sep. 2007, pp. 85–92.

[4] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative 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," *Adv. Neural Inf. Process. Syst.*, vol. 13, pp. 556–562, Apr. 2001.

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, no. 11, pp. 19–60, Jan. 2010.

[7] A. Cichocki, R. Zdunek, and A. H. Phan, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. New York, NY, USA: 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," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 155–173, Sep. 2007.

[9] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," *IEEE Trans. Neural Netw.*, 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 non-negative sources using a Bayesian approach and MCMC sampling," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4133–4145, Nov. 2006.

[11] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1995.

[12] R. Bro and S. D. Jong, "A fast non-negativity-constrained least squares algorithm," *J. Chemometr.*, 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," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 1, pp. 181–217, Mar. 1960.

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

[15] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Number. Anal.*, 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 Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[17] C. J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, 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 Trans. Neural Netw.*, 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 Process.*, 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 Trans. Signal Process.*, vol. 59, no. 11, pp. 5225–5235, Nov. 2011.

[21] J. Chen, C. Richard, and J.-C. M. Bermudez, "Non-stationary analysis of the convergence of the non-negative least-mean-square algorithm," in *Proc. EUSIPCO*, 2013.

[22] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, "A modified non-negative LMS algorithm and its stochastic behavior analysis," in *Proc. ASILOMAR*, 2011, pp. 542–546.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, MA, USA: Cambridge Univ. Press, 2004.

[24] *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*, TU-T recommendation G.726 (former CCLTT recommendation G.721), 1994.

[25] S. Koike, "Analysis of the sign-sign algorithm based on Gaussian distributed tap weights," presented at the Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Seattle, USA, May 1998.

[26] A. H. Sayed, *Adaptive Filters*. New York, NY, USA: Wiley-Interscience, 2008.

[27] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1994.

[28] C. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 10, pp. 1177–1191, Oct. 1983.

[29] 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 Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1394–1405, Jul. 2005.

[30] E. Eweda, "Transient and tracking performance bounds of the sign-sign algorithm," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2200–2210, Aug. 1999.

[31] S. Dasgupta, C. R. Johnson Jr, and A. M. Baksho, "Sign-sign LMS convergence with independent stochastic inputs," *IEEE Trans. Inf. Theory*, vol. 36, no. 1, pp. 197–201, Jan. 1990.

[32] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inf. Theory*, vol. 4, no. 2, pp. 69–72, Jun. 1958.

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

[34] M. H. Costa, J.-C. M. Bermudez, and N. J. Bershad, "Stochastic analysis of the filtered-X LMS algorithm in systems with nonlinear secondary paths," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1327–1342, 2002.

[35] K. S. Miller*, Multidimensional Gaussian Distributions*. New York, NY, USA: Wiley, 1964.

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

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

Jie Chen (S'12–M'14) was born in Xi'an, China, in 1984. He received the B.S. degree in information and telecommunication engineering from the Xi'an Jiaotong University, Xi'an, in 2006 and the Dipl.-Ing. and the M.S. degrees in information and telecommunication engineering in 2009 from the University of Technology of Troyes (UTT), Troyes, France, and from the Xi'an Jiaotong University, respectively. In 2013, he received the Ph.D. degree in systems optimization and security from the UTT.

From April 2013 to March 2014, he was a Postdoctoral Research Fellow at the Cote d'Azur Observatory, University of Nice Sophia Antipolis, Nice, France. Since April 2014, he works as a Postdoctoral Research Fellow with the department of Electrical Engineering and Computer Science of University of Michigan, Ann Arbor, USA.

His current research interests include adaptive signal processing, supervised and unsupervised learning, distributed optimization, hyperspectral image analysis, and bio-signal processing. He is a reviewer for several journals, including IEEE Transactions on Image processing, IEEE Transactions on Geoscience and Remote Sensing and Elsevier Signal Processing.

Cédric Richard (S'98–M'01–SM'07) was born January 24, 1970 in Sarrebourg, France. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from the University of Technology of Compiègne, France, all in electrical and computer engineering. From 1999 to 2003, he was an Associate Professor at the University of Technology of Troyes (UTT), France. From 2003 to 2009, he was a Full Professor at UTT. Since September 2009, he has been a Full Professor in the Lagrange Laboratory (University of Nice Sophia Antipolis, CNRS, Observatoire de la Cote d'Azur). In winter 2009 and 2014, and autumns 2010, 2011 and 2013, he was a Visiting Researcher with the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianopolis, Brazil. He is a junior member of the Institut Universitaire de France since October 2010.

His current research interests include statistical signal processing and machine learning.

Cédric Richard is the author of over 220 papers. He was the General Chair of the XXIth francophone conference GRETSI on Signal and Image Processing that was held in Troyes, France, in 2007, and of the IEEE Statistical Signal Processing Workshop (IEEE SSP'11) that was held in Nice, France, in 2011. He will be the Technical Chair of EUSIPCO 2015. Since 2005, he is a member of GRETSI association board and of the EURASIP society, and Senior Member of the IEEE. In 2006–2010, he served as an associate editor of the IEEE Transactions on Signal Processing. Actually, he serves as an Associate Editor of *Signal Processing* Elsevier, and of the IEEE Signal Processing Letters. He is an EURASIP liaison local officer. He is a member of the Signal Processing Theory and Methods (SPTM TC) Technical Committee, and of the Machine Learning for Signal Processing (MLSP TC) Technical Committee, of the IEEE Signal Processing Society.

Paul Honeine and Cédric Richard won the Best Paper Award for "Solving The Preimage Problem in Kernel Machines: A Direct Method" at the 2009 IEEE International Workshop on Machine Learning for Signal Processing.

Jose-Carlos M. Bermudez (M'85–SM'02) received the B.E.E. degree from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, the M.Sc. degree in electrical engineering from COPPE/UFRJ, and the Ph.D. degree in electrical engineering from Concordia University, Montreal, Canada, in 1978, 1981, and 1985, respectively. He joined the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 1985. He is currently a Professor of Electrical Engineering. In the winter of 1992, he was a Visiting Researcher with the Department of Electrical Engineering, Concordia University. In 1994, he was a Visiting Researcher with the Department of Electrical Engineering and Computer Science, University of California, Irvine (UCI). He was a Visiting Researcher with Institut National Polytechnique de Toulouse, France, in winter 2007, spring 2009, and spring 2011. He was a Visiting Researcher with Université Nice Sophia-Antipolis, France, for spring 2010 and for winter 2014. In 2012 he spent a sabbatical year at Institut National Polytechnique de Toulouse, France. His research interests have involved analog signal processing using continuous-time and sampled-data systems. His recent research interests are in digital signal processing, including linear and nonlinear adaptive filtering, active noise and vibration control, echo cancellation, image processing, and speech processing. Prof. Bermudez served as an Associate Editor of the IEEE Transactions on Signal Processing in the area of adaptive filtering from 1994 to 1996 and from 1999 to 2001. He also served as an Associate Editor of the EURASIP *Journal of Advances on Signal Processing* from 2006 to 2010. He was a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society from 1998 to 2004, and is again member of the Committee since January 2014.

Paul Honeine (M'07) was born in Beirut, Lebanon, on October 2, 1977. He received the Dipl.-Ing. degree in mechanical engineering in 2002 and the M.Sc. degree in industrial control in 2003, both from the Faculty of Engineering, the Lebanese University, Lebanon. In 2007, he received the Ph.D. degree in Systems Optimization and Security from the University of Technology of Troyes, France, and was a Postdoctoral Research associate with the Systems Modeling and Dependability Laboratory, from 2007 to 2008. Since September 2008, he has been an assistant Professor at the University of Technology of Troyes, France. His research interests include nonstationary signal analysis and classification, nonlinear and statistical signal processing, sparse representations, machine learning. Of particular interest are applications to (wireless) sensor networks, biomedical signal processing, hyperspectral imagery and nonlinear adaptive system identification. He is the co-author (with C. Richard) of the 2009 Best Paper Award at the IEEE Workshop on Machine Learning for Signal Processing. Over the past five years, he has published more than 100 peer-reviewed papers.