World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:11, 2008

Mean-Square Performance of Adaptive Filter Algorithms in Nonstationary Environments

Mohammad Shams Esfand Abadi and John Håkon Husøy

***Abstract*—Employing a recently introduced unified adaptive filter theory, we show how the performance of a large number of important adaptive filter algorithms can be predicted within a general framework in nonstationary environment. This approach is based on energy conservation arguments and does not need to assume a Gaussian or white distribution for the regressors. This general performance analysis can be used to evaluate the mean square performance of the *Least Mean Square* (LMS) algorithm, its normalized version (NLMS), the family of *Affine Projection Algorithms* (APA), the *Recursive Least Squares* (RLS), the *Data-Reusing LMS* (DR-LMS), its normalized version (NDR-LMS), the *Block Least Mean Squares* (BLMS), the *Block Normalized LMS* (BNLMS), the *Transform Domain Adaptive Filters* (TDAF) and the *Subband Adaptive Filters* (SAF) in nonstationary environment. Also, we establish the general expressions for the steady-state excess mean square in this environment for all these adaptive algorithms. Finally, we demonstrate through simulations that these results are useful in predicting the adaptive filter performance.

***Keywords*—Adaptive filter, general framework, energy conservation, mean-square performance, nonstationary environment.

I. Introduction

PERFORMANCE analysis of adaptive filtering algorithms in nonstationary environments has been, and still is, an area of active research [1], [2], [3]. When the input signal properties vary with time, the adaptive filters are able to track these variations. The aim of tracking performance analysis is to characterize this tracking ability in nonstationary environments. In this area, many contributions focus on a particular algorithm, making more or less restrictive assumptions on the input signal. For example in [4], [5], the transient performance of the LMS was presented in the nonstationary environments. The former uses a random-walk model for the variations in the optimal weight vector, while the latter assumes deterministic variations in the optimal weight vector. The steady-state performance of this algorithm in the nonstationary environment for the white input is presented in [6]. The tracking performance analysis of the signed regressor LMS algorithm can be found in [7], [8], [9]. Also, the steady-state and tracking analysis of this algorithm without the explicit use of the independence assumptions are presented in [10].

Obviously, a more general analysis encompassing as many different algorithms as possible as special cases, while at the same time making as few restrictive assumptions as possible, is highly desirable. In [11], a unified approach for steady-state and tracking analysis of LMS, NLMS, and some adaptive filters with the nonlinearity property in the error is presented. Their approach was based on energy-conservation relation which was originally derived in [12] and [13]. Also in [14], a unified approach to steady-state performance analysis of a family of affine projection and data-reusing adaptive filter algorithms in the stationary environments and without using the independence assumptions have been presented based on a theory of averaging analysis.

An important recent contribution is the tracking analysis of the Affine Projection Algorithm(s) (APA) [15]. The analysis is based on an energy conservation argument and using the energy relation. Also, the transient and steady-state analysis of data-reusing adaptive algorithms in the stationary environments is presented in [16] based on the weighted energy relation. But the performance of these algorithms in nonstationary environment is not presented. There is a general performance analysis of adaptive filters in [17] and [18]. But again, this analysis was performed in the stationary environments.

We have shown previously [19] that the least mean squares (LMS), the normalized LMS (NLMS), the affine projection algorithm [3], the recursive least squares (RLS), the transform domain adaptive filters (TDAF) [20] and the subband adaptive filters (SAF) [21], [22], [23] can be derived through parameter selections in the generic filter vector update equations presented in [19]. In this paper we extend these generic update equations to show that other adaptive filter algorithms such as the binormalized data-reusing (BNDR-LMS) [24], the NLMS with orthogonal correction factors (NLMS-OCF) [25], the data-reusing adaptive algorithms such as the data-reusing LMS (DR-LMS) [26], the normalized data-reusing LMS (NDR-LMS) [27] and the block adaptive algorithms such as the block LMS (BLMS) and the block NLMS (BNLMS) adaptive algorithms [2] are established through parameter selections in these generic update equations. Accordingly, a general formalism for the mean-square performance analysis of adaptive filters in nonstationary environment is presented. The strategy of the analysis is based on energy conservation arguments and does not need to assume a Gaussian or white distribution for the regressors [3]. Especially, we derive the general expressions for the steady-state mean square error in nonstationary environment for all the adaptive filter algorithms covered by the generic update equations.

We have organized our paper as follows: In the following section we briefly present and extend the generic adaptive filter update equations of [19] forming the basis of our analysis. In the next section, the general mean square performance analysis of adaptive filters in nonstationary environment and the general expression for the steady-state mean square error in this environment are established. We conclude the paper by showing a comprehensive set of simulations supporting the

Mohammad Shams Esfand Abadi and John Håkon Husøy are with the University of Stavanger, Department of Electrical and Computer Engineering, N-4036 Stavanger, Norway, Email: John.H.Husoy@uis.no.

validity of our results.

Throughout the paper, the following notations are adopted:

$\|.\|$ Euclidean norm of a vector.
$\|\underline{t}\|^2_{\mathbf{\Sigma}}$ $\mathbf{\Sigma}$-Weighted Euclidean norm of a column vector $\underline{t}$ defined as $\underline{t}^T\mathbf{\Sigma}\underline{t}$.
vec($\mathbf{T}$) Creating an $M^2 \times 1$ column vector $\underline{t}$ through stacking the columns of the $M \times M$ matrix $\mathbf{T}$.
vec($\underline{t}$) Creating an $M \times M$ matrix $\mathbf{T}$ from the $M^2 \times 1$ column vector $\underline{t}$.
$\mathbf{A} \otimes \mathbf{B}$ Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$.
Tr(.) Trace of a matrix.
$(.)^T$ Transpose of a vector or a matrix.
diag{...}Diagonal matrix of its entries {...}.
$E\{.\}$ Expectation operator.

II. The Generic Adaptive Filter Update Equations and Adaptive Filter Algorithms

In Figure 1 we show the prototypical adaptive filter setup, where $x(n)$, $d(n)$ and $e(n)$ are the input, desired and output error signals, respectively. $\underline{h}(n)$ is the $M \times 1$ column vector of filter coefficients at time n. From [19], the generic filter vector update equation can be stated as,

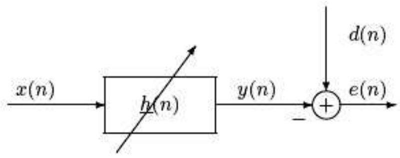

Fig. 1. Prototypical adaptive filter setup

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{C}(n)\mathbf{X}(n)\underline{e}(n). \tag{1}$$

where

$$\underline{e}(n) = \underline{d}(n) - \mathbf{X}^T(n)\underline{h}(n), \tag{2}$$

is the output error vector. The matrix $\mathbf{X}(n)$ is the $M \times K$ input signal matrix defined as

$$\mathbf{X}(n) = [\underline{x}(nL), \underline{x}(nL-D), \ldots, \underline{x}(nL-(K-1)D)], \tag{3}$$

where $\underline{x}(nL) = [x(nL), x(nL-1), \ldots, x(nL-M+1)]^T$, and $\underline{d}(n)$ is a $K \times 1$ vector of desired signal which is defined as

$$\underline{d}(n) = [d(nL), d(nL-D), \ldots, d(nL-(K-1)D)]^T. \tag{4}$$

The parameter K is a positive integer (usually, but not necessarily $K \leq M$), L is the block length[1], and D is the positive integer parameter ($D \geq 1$) that can increase the separation, and consequently reduce the correlation among the regressors in $\mathbf{X}(n)$[2].

The desired signal arise from the following data model

$$d(n) = \underline{x}^T(n)\underline{h}_t(n) + v(n), \tag{5}$$

where $v(n)$ is the measurement noise and assumed to be zero mean, white, Gaussian, and independent of the input signal matrix $\mathbf{X}(n)$ and $\underline{h}_t(n)$ is the true time-variant unknown column vector. We assume that the variation of $\underline{h}_t(n)$ is according to the random walk model [1], [2], [3]

$$\underline{h}_t(n+1) = \underline{h}_t(n) + \underline{q}(n). \tag{6}$$

In Eq. 6, the sequence of $\underline{q}(n)$ is an independent and identically distributed sequence with autocorrelation matrix $\mathbf{Q} = E\{\underline{q}(n)\underline{q}^T(n)\}$ and independent of other sequences.

The matrix $\mathbf{C}(n)$ is some $M \times M$ invertible matrix called the *preconditioner*. Selecting $\mathbf{C}(n)$ as an approximate inverse of the autocorrelation matrix, we can improve the convergence speed dramatically relative to the case when no preconditioner is employed [19]. One strategy for selecting the matrix $\mathbf{C}(n)$ is using the regularized inverse of the estimated autocorrelation matrix as a preconditioner. In this case, by using the matrix inversion lemma, we can write

$$\mathbf{C}(n)\mathbf{X}(n) = \mathbf{X}(n)\mathbf{W}(n) \tag{7}$$

where $\mathbf{W}(n)$ is the $K \times K$ invertible matrix called the *weighting* matrix [17], [18]. For more details please refer to [19]. From this one might argue that in some cases, a suitable alternative form of the generic adaptive filter of Eq. 1 can be stated as:

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{X}(n)\mathbf{W}(n)\underline{e}(n). \tag{8}$$

We are now in the position to make specific choices for the preconditioner matrix $\mathbf{C}(n)$ or the weighting matrix $\mathbf{W}(n)$ as well as for the parameters K, L, and D. Different adaptive filter algorithms can now be seen as specific instantiations of the generic adaptive filter update equations (Eq. 1 and Eq. 8). These algorithms are the least mean squares (LMS), the normalized LMS (NLMS), ϵ-NLMS[3], family of affine projection algorithms (APA) such as the standard version of APA, the regularized APA (R-APA) [28], the binormalized data-reusing LMS (BNDR-LMS) [24], the NLMS with orthogonal correction factors (NLMS-OCF) [25], the data-reusing adaptive algorithms such as the data-reusing LMS (DR-LMS), and the normalized DR-LMS (NDR-LMS) [27], the recursive least squares (RLS)[4], the transform domain adaptive filter (TDAF) algorithms[5][20], and the subband adaptive filters (SAF)[6]. The particular choices and their corresponding algorithms are summarized in Table I. It is interesting to note that the adaptive filter algorithms in [21], [22], [23], while derived from different points of view, are the same [18]. Selecting the parameters in the generic adaptive filter according to Table I for the SAF and setting $\epsilon = 0$, result in Eq. 8 from [22].

[1]Setting $L = 1$ we get sample-by-sample algorithms whereas selecting $L > 1$ results in block-based algorithms in which chunks of L samples are input to the algorithm for each coefficient update.

[2]The choice $D \geq 1$ and $L = 1$ corresponds to NLMS-OCF adaptive filter algorithm [25].

[3]ϵ is the regularization parameter and $\mathbf{I}$ is the identity matrix.

[4]The signal matrix $\tilde{\mathbf{X}}(n)$ has the same structure as $\mathbf{X}(n)$, but with horizontal dimension exceeding the vertical dimension M and $0 \ll \lambda \leq 1$.

[5]The matrix $\mathbf{T}$ is an $M \times M$ orthogonal transform matrix.

[6]$\mathbf{F}$ is the $K \times L$ matrix whose columns are unit pulse responses of a L channel orthogonal perfect reconstruction critically sampled filter bank system. In this case, L is the number of subband and K is the length of the channel filters of the analysis filter bank.

TABLE I
THE MOST COMMON FAMILIES OF ADAPTIVE FILTER ALGORITHMS CAN BE DESCRIBED THROUGH EITHER $\underline{h}(n+1) = \underline{h}(n) + \mu \mathbf{C}(n)\mathbf{X}(n)\underline{e}(n)$ OR $\underline{h}(n+1) = \underline{h}(n) + \mu \mathbf{X}(n)\mathbf{W}(n)\underline{e}(n)$.

Algorithm	K	L	D	$\mathbf{C}(n)/\mathbf{W}(n)$
LMS	$K=1$	$L=1$	$D=1$	$\mathbf{C}(n) = \mathbf{I}$
NLMS	$K=1$	$L=1$	$D=1$	$\mathbf{W}(n) = [1/\Vert\underline{x}(n)\Vert^2].\mathbf{I}$
$\epsilon -$ NLMS	$K=1$	$L=1$	$D=1$	$\mathbf{W}(n) = (\epsilon + \underline{x}^T(n)\underline{x}(n))^{-1}.\mathbf{I}$ or $\mathbf{C}(n) = (\epsilon\mathbf{I} + \underline{x}(n)\underline{x}^T(n))^{-1}$
APA	$K \leq M$	$L=1$	$D=1$	$\mathbf{W}(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}$
BNDR $-$ LMS	$K=2$	$L=1$	$D=1$	$\mathbf{W}(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}$
R $-$ APA	$K \leq M$	$L=1$	$D=1$	$\mathbf{W}(n) = (\epsilon\mathbf{I} + \mathbf{X}^T(n)\mathbf{X}(n))^{-1}$ or $\mathbf{C}(n) = (\epsilon\mathbf{I} + \mathbf{X}(n)\mathbf{X}^T(n))^{-1}$
NLMS $-$ OCF	$K \leq M$	$L=1$	$D \geq 1$	$\mathbf{W}(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}$
DR $-$ LMS	$K \leq M$	$L=1$	$D=1$	$\mathbf{C}(n) = \mathbf{I}$
BLMS	$K=L$	$L <=> M$	$D=1$	$\mathbf{C}(n) = \mathbf{I}$
NDR $-$ LMS	$K \leq M$	$L=1$	$D=1$	$\mathbf{W}(n) = \mathrm{diag}\{1/\Vert\underline{x}(n)\Vert^2, ..., 1/\Vert\underline{x}(n-K+1)\Vert^2\}$
BNLMS	$K=L$	$L <=> M$	$D=1$	$\mathbf{W}(n) = \mathrm{diag}\{1/\Vert\underline{x}(nL)\Vert^2, ..., 1/\Vert\underline{x}(nL-K+1)\Vert^2\}$
SAF	K	L	$D=1$	$\mathbf{W}(n) = \mathbf{F}\{\epsilon\mathbf{I} + \mathrm{diag}\{\mathbf{F}^T\mathbf{X}^T(n)\mathbf{X}(n)\mathbf{F}\}\}^{-1}\mathbf{F}^T$
RLS	$K=1$	$L=1$	$D=1$	$\mathbf{C}(n) = [\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)]^{-1}$ (sliding window) or $\mathbf{C}(n) = [\sum_{i=0}^{n}\lambda^{n-i}\underline{x}(i)\underline{x}^T(i)]^{-1}$ (exp.weighted window)
TDAF	$K=1$	$L=1$	$D=1$	$\mathbf{C}(n) = \mathbf{T}.\{\mathrm{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\mathbf{T}]\}^{-1}.\mathbf{T}^T$ or $\mathbf{C}(n) = \mathbf{T}.\{\mathrm{diag}[\sum_{i=0}^{n}\lambda^{n-i}\mathbf{T}^T\underline{x}(i)\underline{x}^T(i)\mathbf{T}]\}^{-1}.\mathbf{T}^T$

III. GENERAL MEAN-SQUARE PERFORMANCE ANALYSIS OF ADAPTIVE FILTER ALGORITHMS IN NONSTATIONARY ENVIRONMENT

In this section based on the generic update equations, we present the general mean-square performance analysis and develop the general expression for the steady-state excess mean square error (EMSE) in nonstationary environment.

A. General mean-square performance analysis of adaptive filter algorithms in nonstationary environment based on Eq. 1

In the mean-square performance analysis, we need to study the time evolution of the $E\{\Vert\underline{\epsilon}(n)\Vert^2_{\mathbf{\Sigma}}\}$, where $\mathbf{\Sigma}$ is any Hermitian and positive-definite matrix[7], and $\underline{\epsilon}(n)$ is the weight-error vector which is defined as

$$\underline{\epsilon}(n) = \underline{h}_t(n) - \underline{h}(n). \tag{9}$$

From Eq. 9, the generic weight-error vector update equation based on Eq. 1 can be stated as

$$\underline{\epsilon}(n+1) = \underline{\epsilon}(n) + \underline{q}(n) - \mu\mathbf{C}(n)\mathbf{X}(n)\underline{e}(n). \tag{10}$$

From Eq. 2 and Eq. 5, the output estimation error vector $\underline{e}(n)$ can be represented as

$$\underline{e}(n) = \mathbf{X}^T(n)\underline{\epsilon}(n) + \underline{v}(n). \tag{11}$$

[7]When $\mathbf{\Sigma} = \mathbf{I}$, the Mean Square Deviation (MSD) and when $\mathbf{\Sigma} = \mathbf{R}$, where $\mathbf{R} = E\{\underline{x}(n)\underline{x}^T(n)\}$ is the autocorrelation matrix of the input signal, the Excess Mean Square Error (EMSE) expressions are established.

Substitute Eq. 11 in Eq. 10, we obtain

$$\underline{\epsilon}(n+1) = \underline{\epsilon}(n) + \underline{q}(n) - \mu\mathbf{C}(n)\mathbf{X}(n)(\mathbf{X}^T(n)\underline{\epsilon}(n) + \underline{v}(n)). \tag{12}$$

Now taking the $\mathbf{\Sigma}$-weighted norm from both sides of Eq. 12,

$$\Vert\underline{\epsilon}(n+1)\Vert^2_{\mathbf{\Sigma}} = \Vert\underline{\epsilon}(n)\Vert^2_{\mathbf{\Sigma}'} + \Vert\underline{q}(n)\Vert^2_{\mathbf{\Sigma}} + \mu^2\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n) + \{Some\ Cross\ Terms\}, \tag{13}$$

where

$$\mathbf{\Sigma}' = \mathbf{\Sigma} - \mu\mathbf{\Sigma}\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n) - \mu\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\mathbf{\Sigma} + \mu^2\mathbf{X}(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\mathbf{X}^T(n) \tag{14}$$

and

$$\mathbf{X}^{\mathbf{\Sigma}}(n) = \mathbf{X}^T(n)\mathbf{C}^T(n)\mathbf{\Sigma}\mathbf{C}(n)\mathbf{X}(n). \tag{15}$$

Taking the expectation from both sides of Eq. 13

$$E\{\Vert\underline{\epsilon}(n+1)\Vert^2_{\mathbf{\Sigma}}\} = E\{\Vert\underline{\epsilon}(n)\Vert^2_{\mathbf{\Sigma}'}\} + E\{\Vert\underline{q}(n)\Vert^2_{\mathbf{\Sigma}}\} + \mu^2E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\}, \tag{16}$$

we obtain the time evolution of the weight-error variance. The expectation of $\Vert\underline{\epsilon}(n)\Vert^2_{\mathbf{\Sigma}'}$ is difficult to calculate because of dependency of $\mathbf{\Sigma}'$ on $\mathbf{C}(n)$, $\mathbf{X}(n)$, and of $\underline{\epsilon}(n)$ on prior regressors. To solve this problem we need to use the following independence assumptions [15].

1) The matrix sequence $\mathbf{X}(n)$ is independent and identically distributed. This assumption guarantees that $\underline{\epsilon}(n)$ is independent of both $\mathbf{\Sigma}'$ and $\mathbf{X}(n)$.

2) $\underline{\epsilon}(n)$ is independent of $\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)$.

Using these independence assumptions, the final results is

$$E\{\|\underline{\epsilon}(n+1)\|^2_{\mathbf{\Sigma}}\} = E\{\|\underline{\epsilon}(n)\|^2_{\mathbf{\Sigma}'}\} + E\{\|\underline{q}(n)\|^2_{\mathbf{\Sigma}}\} + \mu^2 E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\}, \quad (17)$$

where now $\mathbf{\Sigma}'$ is

$$\begin{aligned}\mathbf{\Sigma}' = \mathbf{\Sigma} - \mu\mathbf{\Sigma}E\{\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)\} \\ -\mu E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}\mathbf{\Sigma} \\ +\mu^2 E\{\mathbf{X}(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\mathbf{X}^T(n)\}. \quad (18)\end{aligned}$$

Looking only at the second term of the right hand side of Eq. 17 we obtain

$$\begin{aligned}E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\} &= E\{\mathrm{Tr}(\underline{v}(n)\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n))\} \\ &= \mathrm{Tr}(E\{\underline{v}(n)\underline{v}^T(n)\}E\{\mathbf{X}^{\mathbf{\Sigma}}(n)\}). \quad (19)\end{aligned}$$

Since $E\{\underline{v}(n)\underline{v}^T(n)\} = \sigma_v^2\mathbf{I}$, where σ_v^2 is the variance of the measurement noise, and $E\{\|\underline{q}(n)\|^2_{\mathbf{\Sigma}}\} = \mathrm{Tr}(\mathbf{Q\Sigma})$, Eq. 17 can be stated as

$$E\{\|\underline{\epsilon}(n+1)\|^2_{\mathbf{\Sigma}}\} = E\{\|\underline{\epsilon}(n)\|^2_{\mathbf{\Sigma}'}\} + \mathrm{Tr}(\mathbf{Q\Sigma}) + \mu^2\sigma_v^2\mathrm{Tr}(E\{\mathbf{X}^{\mathbf{\Sigma}}(n)\}), \quad (20)$$

Applying the vec(.) operator [29] on both sides of Eq. 18 yields

$$\begin{aligned}\mathrm{vec}(\mathbf{\Sigma}') = \mathrm{vec}(\mathbf{\Sigma}) - \mu\mathrm{vec}(\mathbf{\Sigma}E\{\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)\}) - \\ \mu\mathrm{vec}(E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}\mathbf{\Sigma}) + \\ \mu^2\mathrm{vec}(E\{\mathbf{X}(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\mathbf{X}^T(n)\}). \quad (21)\end{aligned}$$

Since, in general $\mathrm{vec}(\mathbf{P\Sigma Q}) = (\mathbf{Q}^T \otimes \mathbf{P})\mathrm{vec}(\mathbf{\Sigma})$ [29], we find that Eq. 21 can be written as

$$\begin{aligned}\underline{\sigma}' = \underline{\sigma} - \mu(E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\} \otimes \mathbf{I}).\underline{\sigma} - \\ \mu(\mathbf{I} \otimes E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}).\underline{\sigma} + \\ \mu^2(E\{(\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)) \otimes (\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n))\}).\underline{\sigma}, \quad (22)\end{aligned}$$

where $\underline{\sigma}' = \mathrm{vec}(\mathbf{\Sigma}')$ and $\underline{\sigma} = \mathrm{vec}(\mathbf{\Sigma})$. With definition of the $M^2 \times M^2$ matrix $\mathbf{G}$,

$$\begin{aligned}\mathbf{G} = \mathbf{I} - \mu E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\} \otimes \mathbf{I} - \\ \mu\mathbf{I} \otimes E\{\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\} + \\ \mu^2 E\{(\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)) \otimes (\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n))\}, \quad (23)\end{aligned}$$

Eq. 22 can be stated as

$$\underline{\sigma}' = \mathbf{G}.\underline{\sigma}. \quad (24)$$

The second term of the right hand side of Eq. 20 can be written as

$$\mathrm{Tr}(E\{\mathbf{X}^{\mathbf{\Sigma}}(n)\}) = \mathrm{Tr}(E\{\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}.\mathbf{\Sigma}). \quad (25)$$

Defining $\underline{\gamma}$ through

$$\underline{\gamma} = \mathrm{vec}(E\{\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}) \quad (26)$$

we have

$$\mathrm{Tr}(E\{\mathbf{C}(n)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{C}^T(n)\}) = \underline{\gamma}^T.\underline{\sigma} \quad (27)$$

With the above considerations, the recursion of Eq. 20 can now be stated as

$$E\{\|\underline{\epsilon}(n+1)\|^2_{\underline{\sigma}}\} = E\{\|\underline{\epsilon}(n)\|^2_{\mathbf{G}\underline{\sigma}}\} + \mu^2\sigma_v^2\underline{\gamma}^T\underline{\sigma} + \mathrm{Tr}(\mathbf{Q\Sigma}). \quad (28)$$

From this recursion, we will be able to evaluate the steady-state excess mean square error (EMSE). When n goes to infinity, we obtain

$$E\{\|\underline{\epsilon}(\infty)\|^2_{\underline{\sigma}}\} = E\{\|\underline{\epsilon}(\infty)\|^2_{\mathbf{G}\underline{\sigma}}\} + \mu^2\sigma_v^2\underline{\gamma}^T\underline{\sigma} + \mathrm{Tr}(\mathbf{Q\Sigma}). \quad (29)$$

therefore

$$E\{\|\underline{\epsilon}(\infty)\|^2_{(\mathbf{I}-\mathbf{G})\underline{\sigma}}\} = \mu^2\sigma_v^2\underline{\gamma}^T\underline{\sigma} + \mathrm{Tr}(\mathbf{Q\Sigma}). \quad (30)$$

If $(\mathbf{I}-\mathbf{G})\underline{\sigma} = \mathrm{vec}(\mathbf{I})$ and $(\mathbf{I}-\mathbf{G})\underline{\sigma} = \mathrm{vec}(\mathbf{R}) = \underline{r}$, the steady-state MSD and EMSE expressions in nonstationary environment are established respectively. Doing this, the final results are

$$\mathrm{EMSE} = \mu^2\sigma_v^2\underline{\gamma}^T(\mathbf{I}-\mathbf{G})^{-1}\underline{r} + \mathrm{Tr}(\mathbf{Q}\mathrm{vec}((\mathbf{I}-\mathbf{G})^{-1}\underline{r})), \quad (31)$$

and the mean square coefficient deviation (MSD) is given by

$$\begin{aligned}\mathrm{MSD} = \mu^2\sigma_v^2\underline{\gamma}^T(\mathbf{I}-\mathbf{G})^{-1}\mathrm{vec}(\mathbf{I}) + \\ \mathrm{Tr}(\mathbf{Q}\mathrm{vec}((\mathbf{I}-\mathbf{G})^{-1}\mathrm{vec}(\mathbf{I}))). \quad (32)\end{aligned}$$

Also, from Eq. 11, we know that $e(n) = \underline{x}^T(n)\underline{\epsilon}(n) + v(n)$. Therefore, the steady-state MSE is given by

$$\mathrm{MSE} = \mathrm{EMSE} + \sigma_v^2. \quad (33)$$

From the general expression (Eq. 31), we will be able to predict the steady-state performance of LMS, ϵ-NLMS, R-AP, BLMS, DR-LMS, RLS, and the transform domain adaptive filter algorithms in nonstationary environment.

B. General mean-square performance analysis of adaptive filter algorithms in nonstationary environment based on Eq. 8

Following the same approach in previous section for the generic update equation (Eq. 8), the steady-state EMSE is given by:

$$\mathrm{EMSE} = \mu^2\sigma_v^2\underline{\varrho}^T(\mathbf{I}-\mathbf{Z})^{-1}\underline{r} + \mathrm{Tr}(\mathbf{Q}\mathrm{vec}((\mathbf{I}-\mathbf{Z})^{-1}\underline{r})), \quad (34)$$

where

$$\begin{aligned}\mathbf{Z} = \mathbf{I} - \mu E\{\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\} \otimes \mathbf{I} - \\ \mu\mathbf{I} \otimes E\{\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\} + \\ \mu^2 E\{(\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)) \otimes (\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n))\}, \quad (35)\end{aligned}$$

and

$$\underline{\varrho} = \mathrm{vec}(E\{\mathbf{X}(n)\mathbf{W}^2(n)\mathbf{X}^T(n)\}). \quad (36)$$

Also, the mean square coefficient deviation (MSD) in the steady-state is obtained by

$$\begin{aligned}\mathrm{MSD} = \mu^2\sigma_v^2\underline{\varrho}^T(\mathbf{I}-\mathbf{Z})^{-1}\mathrm{vec}(\mathbf{I}) + \\ \mathrm{Tr}(\mathbf{Q}\mathrm{vec}((\mathbf{I}-\mathbf{Z})^{-1}\mathrm{vec}(\mathbf{I}))), \quad (37)\end{aligned}$$

and, the steady-state MSE is given by Eq. 33. From the general expression (Eq. 34), we will be able to predict the steady-state performance of LMS, ϵ-NLMS, AP, R-AP, BNDR-LMS, NLMS-OCF, BLMS, BNLMS, DR-LMS, NDR-LMS, and the subband adaptive filter algorithms in nonstationary environment.

IV. Simulation results

We justify the theoretical results presented in this paper by several computer simulations in a system identification setup. The unknown system has 8 taps and is selected at random. The input signal, $x(n)$ is a first order autoregressive (AR(1)) signal generated according to

$$x(n) = \rho x(n-1) + w(n), \tag{38}$$

where $w(n)$ can be either a zero mean white Gaussian signal or a zero mean uniformly distributed random sequence between -1 and 1. For the Gaussian case, ρ is set to 0.9. As a result, a highly colored Gaussian signal is generated. For the uniform case, ρ is set to 0.5. The measurement noise, $v(n)$, with $\sigma_v^2 = 10^{-3}$ was added to the noise free desired signal generated through $d(n) = \underline{h}_t(n)^T \underline{x}(n)$. The unknown channel changes according to Eq. 6. We assumed an independent and identically distributed sequence for $\underline{q}(n)$ with autocorrelation matrix $\mathbf{Q} = \sigma_q^2.\mathbf{I}$ where $\sigma_q^2 = 0.0025\sigma_v^2$. The adaptive filter and the unknown channel are assumed to have the same number of taps. For the TDAF algorithm, an 8-point Discrete Cosine Transform (DCT) was employed as the orthogonal transform. The filter bank used in the subband adaptive filters was the four subband Extended Lapped Transform (ELT) [30]. In all the simulations actually observed steady-state MSE are obtained by averaging over 500 steady-state samples from 500 independent realizations for each μ value for a given algorithm.

Figs. 2 and 3 show the steady-state MSE curves of NDR-LMS adaptive algorithm as a function of the step-size in nonstationary environment for both colored Gaussian and uniform input signals with $K = 2$. The theoretical results are calculated according to Eq. 33 and Eq. 34. As we can see there is a global minimum for the steady-state MSE in nonstationary environment. The theoretical results are in good agreement with simulation results. The agreement is better for the small value of the step-size for both colored and Gaussian input signals. For the large value of the step-size, some deviation between simulated and theoretical values is observed. But the results are still useful.

Figs. 4 and 5 show the the steady-state MSE curves of the TDAF algorithm as a function of the step-size in nonstationary environment. Fig. 4 shows the results for colored Gaussian input. The theoretical results have been obtained through Eq. 31 and Eq. 33. Again in the results, there is an optimal value for the step-size, that minimizes the MSE in the nonstationary environments. This fact can be seen in Fig. 5 for colored uniform input signal. Good agreement between simulated and theoretical values, especially for small step-size is again observed.

Figs. 6 and 7 show the steady-state MSE curves of the subband adaptive filter algorithm as a function of the step-size in nonstationary environment for both colored Gaussian and uniform input signals. In both simulations, the the number of subband is set to 4. The theoretical results are calculated according to Eq. 33 and Eq. 34. The results are in good agreement with simulation results and as before, there is an optimal value for the step-size, that minimizes the MSE in nonstationary environment. Compared with the other simulations, the theoretical values don't have as good agreement with the simulated values as before. But still, reasonable agreements, especially for the large value of the step-size for both colored Gaussian and uniform input signals are observed.

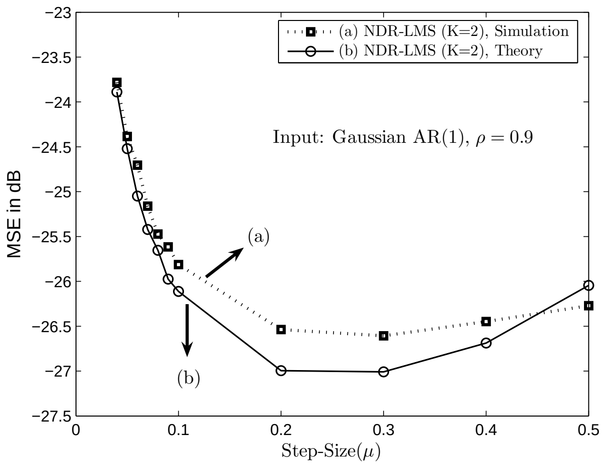

Fig. 2. Steady-state MSE of normalized data-reusing LMS (NDR-LMS) algorithm as a function of the step-size with $K = 2$ in nonstationary environment for colored Gaussian input (Input: Gaussian AR(1), $\rho = 0.9$).

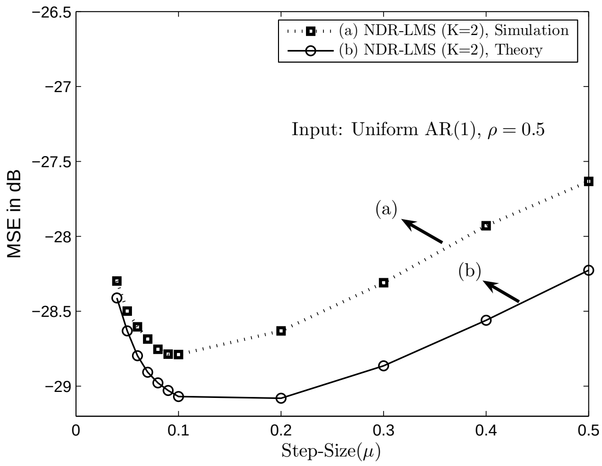

Fig. 3. Steady-state MSE of normalized data-reusing LMS (NDR-LMS) algorithm as a function of the step-size with $K = 2$ in nonstationary environment for colored Uniform input (Input: Uniform AR(1), $\rho = 0.5$).

V. Summary and conclusion

In this paper we have presented a general framework for the mean square performance analysis of adaptive filter algorithms based on generic adaptive filter update equations presented in [19] in the nonstationary environments. Through the general expressions and selection of the parameters according to Table I, the steady-state EMSE of the LMS, NLMS, ϵ-NLMS, family of AP (R-APA, BNDR-LMS, NLMS-OCF), the data-reusing (DR-LMS, NDR-LMS), RLS, the transform domain,

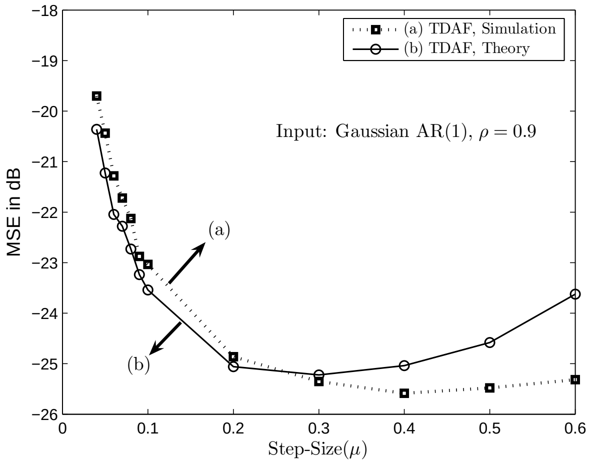

Fig. 4. Steady-state MSE of TDAF algorithm as a function of the step-size in nonstationary environment for colored Gaussian input (Input: Gaussian AR(1), $\rho = 0.9$).

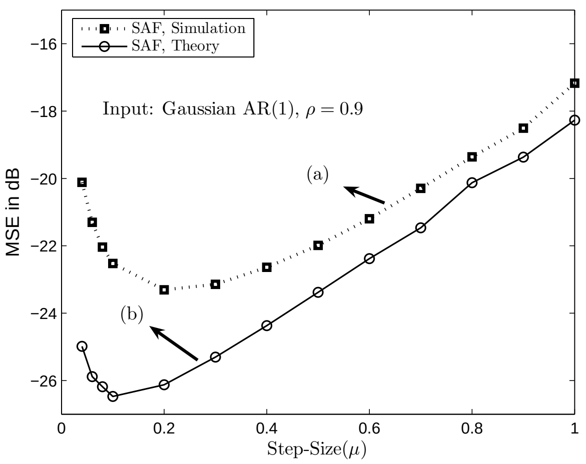

Fig. 6. Steady-state MSE of SAF algorithm as a function of the step-size in nonstationary environment for colored Gaussian input (Input: Gaussian AR(1), $\rho = 0.9$).

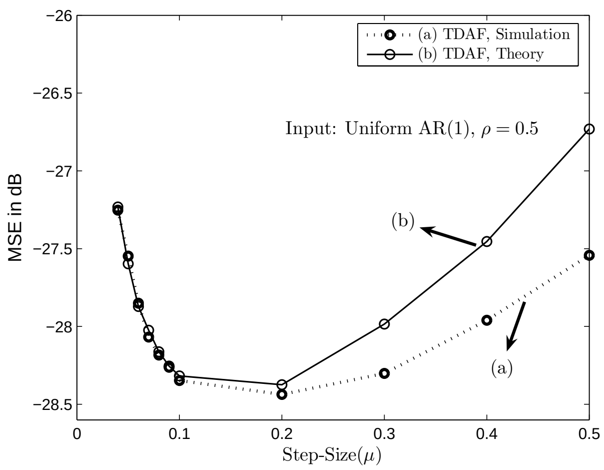

Fig. 5. Steady-state MSE of TDAF algorithm as a function of the step-size in nonstationary environment for colored Uniform input (Input: Uniform AR(1), $\rho = 0.5$).

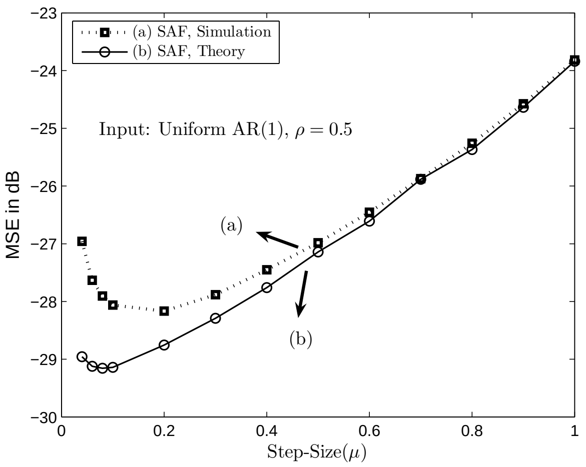

Fig. 7. Steady-state MSE of SAF algorithm as a function of the step-size in nonstationary environment for colored Uniform input (Input: Uniform AR(1), $\rho = 0.5$).

the block adaptive filters (BLMS, BNLMS), and the subband, adaptive filter algorithms were predicted in the nonstationary environment. We demonstrated the usefulness of the general performance results for NDR-LMS, the transform domain, and the subband adaptive filter algorithms.

References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[2] S. Haykin, *Adaptive Filter Theory*. NJ: Prentice-Hall, 4th edition, 2002.
[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Wiley, 2003.
[4] B. Widrow, J. M. McCool, M. Larimore, and C. R. Johnson, “Stationary and nonstationorny learning characteristics of the LMS adaptive filter,” in *Proc. IEEE*, 1976, pp. 1151–1162.
[5] N. J. Bershad, P. Feintuch, A. Reed, and B. Fisher, “Tracking charcteristics of the LMS adaptive line enhancer: Response to a linear chrip signal in noise,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 504–516, 1980.
[6] S. Marcos and O. Macchi, “Tracking capability of the least mean square algorithm: Application to an asynchronous echo canceller,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 1570–1578, 1987.
[7] E. Eweda, “Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data,” *IEEE Trans. Circuits, Syst.*, vol. 37, pp. 1367–1374, Nov. 1990.
[8] ——, “Optimum step size of the sign algorithm for nonstationary adaptive filtering,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1897–1901, 1990.
[9] ——, “Comparsion of RLS and LMS, and sign algorithms for tracking randomly time-varying channels,” *IEEE Trans. Signal Processing*, vol. 42, pp. 2937–2944, 1994.
[10] N. R. Yousef and A. H. Sayed, “Steady-state and tracking analyses of the sign algorithm without the explicit use of the independence assumption,” *IEEE Signal Processing Letters*, vol. 7, pp. 307–309, 2000.
[11] ——, “A unified approach to the steady-state and tracking analyses of adaptive filters,” *IEEE Trans. Signal Processing*, vol. 49, pp. 314–324, 2001.
[12] A. H. Sayed and M. Rupp, “A time-domain feedback analysis of adaptive algorithms via the small gain theorem,” in *Proc. SPIE*, vol. 2563, 1995, pp. 458–469.
[13] M. Rupp and A. H. Sayed, “A time-domain feedback analysis of filtered-

error adaptive gradient algorithms," *IEEE Trans. Signal Processing*, vol. 44, pp. 1428–1439, 1996.

[14] M. S. E. Abadi and A. M. Far, "A unified approach to steady-state performance analysis of adaptive filters without using the independence assumptions," *Signal Processing*, vol. 87, pp. 1642–1654, 2007.

[15] H.-C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Processing*, vol. 52, pp. 90–102, Jan. 2004.

[16] H.-C. Shin, W. J. Song, and A. H. Sayed, "Mean-square performance of data-reusing adaptive algorithms," *IEEE Signal Processing Letters*, vol. 12, pp. 851–854, Dec. 2005.

[17] J. H. Husøy and M. S. E. Abadi, "A common framework for transient analysis of adaptive filters," in *Proc. 12th IEEE Mediterranean Electrotechnical Conference*, Dubrovnik, Croatia, May 2004, pp. 265–268.

[18] ——, "Transient analysis of adaptive filters using a general framework," *Automatika, Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 45, pp. 121–127, 2004.

[19] J. H. Husøy, "A streamlined approach to adaptive filters," in *Proc. EUSIPCO*, Firenze, Italy, Sept. 2006, published online by EURASIP at http://www.arehna.di.uoa.gr/Eusipco2006/papers/1568981236.pdf.

[20] P. S. R. Diniz, *Adaptive Filtering: Algorithms and practical implementation*, 2nd ed. Kluwer, 2002.

[21] S. S. Pradhan and V. E. Reddy, "A new approach to subband adaptive filtering," *IEEE Trans. Signal Processing*, vol. 47, pp. 655–664, 1999.

[22] M. de Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," *IEEE Trans. Signal Processing*, vol. 46, pp. 2359–2371, 1998.

[23] K. A. Lee and W. S. Gan, "Improving convergence of the NLMS algorithm using constrained subband updates," *IEEE Signal Processing Letters*, vol. 11, pp. 736–739, 2004.

[24] J. Apolinario, M. L. Campos, and P. S. R. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," *IEEE Trans. Signal Processing*, vol. 48, pp. 3235–3242, Nov. 2000.

[25] S. G. Sankaran and A. A. L. Beex, "Normalized LMS algorithm with orthogonal correction factors," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, 1997, pp. 1670–1673.

[26] B. A. Schnaufer and W. K. Jenkins, "New data-reusing LMS algorithms for improved convergence," in *Proc. Asimolar Conf.*,, Pacific Groves, CA, May 1993, pp. 1584–1588.

[27] R. A. Soni, W. K. Jenkins, and K. A. Gallivan, "Acceleration of normalized adaptive filtering data-reusing methods using the Tchebyshev and conjugate gradient methods," in *Proc. Int. Symp. Circuits Systems*, 1998, pp. 309–312.

[28] S. L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*. Boston, MA: Kluwer, 2000.

[29] T. K. Moon and W. C. Sterling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River: Prentice Hall, 2000.

[30] H. Malvar, *Signal Processing with Lapped Transforms*. Artech House, 1992.

Mohammad Shams Esfand Abadi was born in Tehran, Iran, on September 18, 1978. He received the B.S. degree in electrical engineering from Mazandaran University, Mazandaran, Iran and the M.S. degree in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 2000 and 2002, respectively and the Ph.D. degree in biomedical engineering from Tarbiat Modares University, Tehran, Iran in 2007. Since 2004 he has been with the department of electrical engineering, Shahid Rajaee University, Tehran, Iran. During the fall of 2003, spring 2005, and again in the spring of 2007, he has a visiting scholar with the Signal Processing Group at the University of Stavanger, Norway. His research interests include digital filter theory and adaptive signal processing algorithms.

John Håkon Husøy received the M.Sc. and Ph.D. degrees in electrical engineering from the Norwegian University of Science and Technology. In his early career he was involved in hardware and software development in various positions in several companies in Canada and Norway. Since 1992 he has been a Professor with the Department of Electrical and Computer Engineering, University of Stavanger, Norway. His research interests include adaptive algorithms, digital filtering, signal representations, image compression, bioelectrical signal processing, and image analysis.