Locally Optimum Adaptive Signal Processing Algorithms

George V. Moustakides

***Abstract*— We propose a new analytic method for comparing constant gain adaptive signal processing algorithms. Specifically, estimates of the convergence speed of the algorithms allow for the definition of a local measure of performance, called the efficacy, that can be theoretically evaluated. By definition, the efficacy is consistent with the fair comparison techniques currently used in signal processing applications. Using the efficacy as a performance measure, we prove that the LMS-Newton algorithm is optimum and is, thus, the fastest algorithm within a very rich algorithmic class. Furthermore, we prove that the regular LMS is better than any of its variants that apply the same nonlinear transformation on the elements of the regression vector (such as signed regressor, quantized regressor, etc.) for an important class of input signals. Simulations support all our theoretical conclusions.**

***Index Terms*— Adaptive estimation, adaptive filters, adaptive signal processing.**

I. Introduction

Adaptive signal processing algorithms (ASPA's) are widely used in many application areas such as filtering, control, communications, seismology, etc. Practice has shown their definite superiority as compared with classical techniques because of their ability to adapt to changing and unknown environments.

Due to their practical importance, there has been a great number of ASPA's developed over the years, each with its own merits and drawbacks. The existence of such a large variety of techniques naturally raises the question of performance evaluation and, consequently, the need for development of comparison methods that can rank the algorithms with respect to some desirable characteristics.

The most important characteristic of the ASPA can probably be considered to be the convergence speed during the initial transient phase. In fact, it is with respect to this characteristic that algorithms are usually compared in practice. As far as initial transient phase is concerned, performance evaluation and comparison methods rely either on simulations or on theoretical developments that most of the time consider special types of input signals (such as i.i.d. or Gaussian). Regarding the theoretical results, it must also be noted that even under the aforementioned special input signals, they make use of the well-known *independence assumption* (IA) in order to derive tractable expressions for the convergence rate. Although the IA is obviously erroneous, practice has shown that the corresponding conclusions are very plausible and in accordance with simulations. This is particularly the case when the step size in the adaptive algorithm is small (which is the most practically interesting case).

A common algorithmic model known for its simplicity and practical usefulness, which is often analyzed in the literature, is the generalized form of LMS

$$W_n = W_{n-1} + \mu g(\epsilon_n) f(X_n) \tag{1}$$

where

$$\epsilon_n = y_n - W_{n-1}^t X_n \tag{2}$$

and where $g(x), f(x)$ are scalar nonlinearities, and X_n is the input data vector with $f(X_n)$ denoting a vector obtained by applying the scalar nonlinearity $f(x)$ on every element of the vector X_n. Finally, $y(n)$ denotes the desired response, and $\mu > 0$ is a scalar positive quantity known as step size.

Existing results aiming in the optimization of the convergence speed of the algorithm in (1) by proper selection of the nonlinearities $g(x)$ and $f(x)$ are limited. In [9], we can find the analysis of the transient behavior of the algorithm defined in (1) for the case where $X_n = [x_n \cdots x_{n-N+1}]^t$ and the sequence $\{x_n\}$ being i.i.d. The analysis is based on the IA, and in the paper, it is strongly suggested (although not proved) that the regular LMS, i.e., the algorithm with $g(x) = f(x) = x$, is optimum. In [4], the special case $g(x) = x$ is considered; again, $X_n = [x_n \cdots x_{n-N+1}]^t$, but the processes $\{x_n\}$ is now assumed Gaussian. Using the IA, it is proved that when $\{x_n\}$ is also white, then the optimum scalar nonlinearity $f(x)$ has the form $f(x) = (x/c+\mu x^2)$ for a properly selected constant c. If, however, the sequence $\{x_n\}$ is Gaussian but not white, then it is shown that the nonlinearity $f(x) = x$ (the LMS algorithm) is locally optimum, i.e., optimum for the case $\mu \ll 1$.

In this paper, we also intend to consider the local case $\mu \ll 1$ and attempt to find the algorithm that has the fastest convergence rate for a considerably more general algorithmic class than the one studied in [4]. Furthermore, our optimality result will be shown to be valid for general input sequences without confining ourselves to Gaussian or i.i.d. data. Specifically, we are going to show that the LMS-Newton algorithm is optimum within a very rich algorithmic class and for a large variety of input signals. A second result consists of extending the local optimality property of LMS, which has

Manuscript received April 2, 1997; revised March 19, 1998. This work was supported by the Greek General Secretariat for Research and Technology under Grant ΠΕΝΕΔ 96:1584. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chi Chung Ko.

The author is with the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece.

Publisher Item Identifier S 1053-587X(98)08698-X.

been proved in [4] for Gaussian data, to a very interesting data class that contains the Gaussian as special case.

The rest of the paper is organized as follows. Section II has background material needed for our analysis. In Section III, we introduce our theoretical local performance measure. The optimum, with respect to this measure, algorithms are presented in Section IV. Section V contains simulations, and finally, Section VI has our conclusion.

II. Background Material

In this section, we are going to introduce the background material that is necessary for defining our performance measure. Let us first introduce our notation; with lowercase letters, we will denote scalars, and with uppercase vectors and with boldface uppercase letters, we will denote matrices. For X, a vector $X^{[i]}$ will denote its ith element.

Let us now introduce the data model we intend to use and the algorithms of interest. Suppose we are given sequentially two real processes $\{y_n\}, \{X_n\}$ related by

$$y_n = W_*^t X_n + w_n \tag{3}$$

where, as before

X_n input data vector;
y_n desired response;
w_n additive noise;
W_* deterministic vector.

Both vectors X_n and W_* are of length N. We are interested in adaptive algorithms that estimate W_* with the recursion

$$W_n = W_{n-1} + \mu \sum_{i=0}^{p-1} \epsilon_{n,i} F_i(X_n, \cdots, X_1) \tag{4}$$

where

$$\epsilon_{n,i} = y_{n-i} - W_{n-1}^t X_{n-i}, \qquad i = 0, \cdots, p-1 \tag{5}$$

with W_n being the estimate of W_* at time $n, \mu > 0$ is the scalar *step size,* and $F_i(X_n, \cdots, X_1), i = 0, \cdots, p-1$, are p real nonlinear vector transformations of the input data history $X_n, \cdots, X_1$. The vector transformations $F_i(X_n, \cdots, X_1)$ have the same length with W_n and are known as *regression vectors*. The algorithmic class defined by (4) and (5) is very rich, containing most known algorithms encountered in practice as regular, normalized, signed and quantized regressor LMS, constant forgetting factor RLS, sliding window RLS, underdetermined RLS, adaptive Newton, adaptive instrumental variables, etc.

We like to stress that the algorithm considered in [4], corresponding to (1) with $g(x) = x$, is significantly less general than the proposed algorithmic model defined by (4) and (5). Indeed, the algorithm in [4] can be obtained by selecting $p = 1$ and limiting $F_0(X_n, \cdots, X_1)$ to nonlinear transformations of the form $F_0(X_n)$ applied only on the current input vector X_n with the additional constraint that $F_0(X_n) = f(X_n)$, i.e., the same scalar nonlinearity $f(x)$ is applied to all elements of the vector X_n.

As in [4], our aim is to find the nonlinear transformations $F_i(X_n, \cdots, X_1), i = 0, \cdots, p-1$ that optimize the convergence speed of the corresponding algorithm during the transient phase. This will be made possible by defining a suitable local performance measure that is equivalent to the convergence speed. The rest of this section and the next section will be devoted to the definition of this performance measure.

Except the convergence speed of W_n toward W_*, which, as we said, is the primary characteristic in which we are interested, there is also another quantity that plays an important role in the analysis of adaptive algorithms. We refer to the amount of fluctuation of W_n around the ideal vector W_* under steady state. Regarding the steady-state performance, there exist very powerful results based on the theory of *stochastic approximation* with the important observation that their development does not require the employment of the IA. In fact, the algorithms covered by this theory are even more general than the ones defined by (4) and (5) [3], [22]. Unfortunately, corresponding results for the transient phase and, more specifically, for the convergence rate are not as widely available as for the steady state. Existing results either make use of the IA and obtain estimates of the convergence rate [4], [9], [10], [18] or do not rely on the IA and obtain only bounds for the desired rate [12]–[14]. Since, in order to define a reliable performance measure, we need efficient estimates of both the convergence rate and the steady-state performance, it seems rather imperative to base our analysis on the IA (as was the case in most similar publications in the past). However, it should be noted that regardless of the seemingly crude approximation induced by the use of the IA, the actual error seems in fact to be not so significant. We can, for example, easily verify that at steady state, the performance estimates obtained using the IA are correct up to a first-order approximation with respect to μ (as compared with the results obtained by the stochastic approximation theory, which does not rely on the IA). Furthermore, there are strong indications due to simulations [9], [18] and recent theoretical developments that are not based on the IA [21] that the same property is also true for estimates of the convergence rate. This means that the IA yields, for the practically important local case $\mu \ll 1$, very satisfactory estimates.

A. Assumptions

Before stating the two theorems that will provide the required estimates for the convergence speed and the steady-state performance, let us introduce our assumptions that are necessary for our analysis.

1) The input vector process $\{X_n\}$ and the regression vector processes $\{F_i(X_n, \cdots, X_1)\}, i = 0, \cdots, p-1$ are all stationary and have up to fourth-order bounded moments; furthermore, we require the input covariance matrix $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$ to be positive definite, meaning that the input process is persistently exciting.
2) The additive noise process $\{w_n\}$ is stationary, zero mean, white, and independent of the processes $\{X_n\}$ with a finite variance equal to σ_w^2.

The assumptions we made are rather mild; furthermore, it is worth mentioning that we do not limit ourselves to data that are of a specific type as i.i.d. or Gaussian, as is the case in previous publications. Despite the mildness of our assumptions, there are, nevertheless, implications on the class of allowable algorithms. Thus, let us examine each assumption separately and elaborate on the effect it produces on the algorithmic class and the input data we like to consider.

For the problem we are addressing, the assumption that the input data process $\{X_n\}$ is stationary is not restrictive. Since we are interested in the convergence speed of the algorithms and not in their tracking capability, such an assumption is customary. The requirement that this process is persistently exciting is also very common since it guarantees convergence of the estimates W_n toward the ideal vector W_* (here, convergence in the mean).

The assumption that the regression vector processes $\{F_i(X_n, \cdots, X_1)\}$ are stationary is the most crucial one. Notice that the regression vectors relate to the whole history of the input data sequence and not just the current data vector X_n; therefore, the amount of information used for their computation increases with time, resulting in time-varying statistics. These statistics, however, as time progresses, tend, in most algorithms, to a steady state. Such is, for example, the case when exponential windowing of the data is used (i.e., forgetting factor RLS). Here, we clearly have violation of the stationarity assumption unless the regression vectors converge to their steady-state statistics significantly faster than the corresponding estimates W_n, or the algorithm is already in steady state, and there is a sudden change in the vector W_*. On the other hand, there exists an abundance of algorithms for which the regression vectors have finite memory being of the form $F_i(X_n, \cdots, X_{n-M})$. Such algorithms are, for example, the ones applying a sliding window on the data. Hence, once the whole window is covered with data, we clearly have stationarity for the regression vectors (i.e., sliding window RLS).

The requirement that the fourth-order moments of X_n and $F_i(X_n, \cdots, X_1)$ exist is technical and ensures existence of all expectations appearing in our analysis. Finally, Assumption A2 introduces the simplest and most commonly used noise model.

B. Excess Mean Square Error

In most signal processing applications, the quality of the estimate W_n is not measured through the estimation error vector $\Delta_n = W_n - W_*$ but rather through the scalar error

$$\begin{aligned}\epsilon_n &= y_n - W_{n-1}^t X_n = w_n - (W_{n-1} - W_*)^t X_n \\ &= w_n - \Delta_{n-1} X_n \end{aligned} \tag{6}$$

and, more precisely, through the *mean square error* $\mathbb{E}\{\epsilon_n^2\} = \sigma_w^2 + \mathbb{E}\{e_n^2\}$, where $e_n = \Delta_{n-1}^t X_n$, and $\mathbb{E}\{\cdot\}$ denotes expectation. The quantity e_n is known as *excess error*, its power $\mathbb{E}\{e_n^2\}$ as *excess mean square error*, and the ratio $\mathbb{E}\{e_n^2\}/\sigma_w^2$ as *misadjustment* [15].

Since σ_w^2 is part of the mean square error of every algorithm in our class, the excess mean square error becomes a natural candidate for the subject of study. Another possibility could, of course be the power of the estimation error vector, namely, $\mathbb{E}\{\|\Delta_n\|^2\}$. Clearly, the latter would be more appropriate for a pure estimation problem, but in this work, we concentrate on the former.

Let us from now on, for notational simplicity, denote the regression vectors as

$$Z_{n,i} = F_i(X_n, \cdots, X_1). \tag{7}$$

Because $\{w_n\}$ is independent of the processes $\{X_n\}$ and since the regression vectors $Z_{n,i}$ are functions only of input history, we conclude that $\{w_n\}$ is also independent of the processes $\{Z_{n,i}\}$. This suggests the following partitioning of the estimation error vector Δ_n into two parts, namely

$$\Delta_n = \Gamma_n + \Pi_n \tag{8}$$

satisfying the recursions

$$\begin{aligned}\Gamma_n &= \left(\boldsymbol{I}_N - \mu \sum_{i=0}^{p-1} Z_{n,i} X_{n-i}^t\right)\Gamma_{n-1}, \Gamma_0 = \Delta_0 \\ \Pi_n &= \left(\boldsymbol{I}_N - \mu \sum_{i=0}^{p-1} Z_{n,i} X_{n-i}^t\right)\Pi_{n-1} \\ &\quad + \mu \sum_{i=0}^{p-1} w_{n-i} Z_{n,i}, \Pi_0 = 0 \end{aligned} \tag{9}$$

where $\boldsymbol{I}_m$ denotes the identity matrix of dimension m, and $\Delta_0 = W_0 - W_*$ is some constant but arbitrary vector. The above partition in turn suggests a corresponding partition of the excess mean square error in the form

$$\begin{aligned}\mathbb{E}\{e_n^2\} &= \gamma_n + \pi_n \\ \gamma_n &= \mathbb{E}\{(\Gamma_{n-1}^t X_n)^2\} \\ \pi_n &= \mathbb{E}\{(\Pi_{n-1}^t X_n)^2\}. \end{aligned} \tag{10}$$

Note that the crossterm $\mathbb{E}\{(\Gamma_{n-1}^t X_n)(\Pi_{n-1}^t X_n)\}$ is equal to zero due to the independence between $\{w_n\}$ from one side and $\{X_n\}$ and $\{Z_{n,i}\}$ from the other. Part γ_n is due to the initial conditions and the fact that our initial estimate is away from the true value W_*. We can see that it starts from a $\Theta(1)$[1] value, and we are going to show below that for stable algorithms, it tends exponentially fast to zero. Part π_n, on the other hand, is due to the additive noise. It starts from an $\Theta(\mu^2)$ value and tends to a value $\Theta(\mu)$ at steady state. Since, in our study, we are concerned with the local case $\mu \ll 1$, we conclude that part γ_n is responsible for the transient phase of the algorithm, whereas part π_n is responsible for the steady-state behavior of the excess mean square error. The next subsection introduces estimates of the convergence rate and the steady-state excess mean square error which, as we said, are necessary for defining our performance measure.

C. Exponential Convergence and Steady-State Excess Mean Square Error

From this point on, we are going to rely on the IA to derive our results. Let us first obtain estimates for the convergence rate of γ_n. The exponential rate of γ_n is defined as $\lim_{n\to\infty} (\log(\gamma_n^{-1})/n)$, and the following theorem estimates it.

[1] By the expression $\Theta(x)$, we mean that $c_1|x| \leq |\Theta(x)| \leq c_2|x|$ for constants c_1, c_2. In addition, by $o(x)$, we mean that $\lim_{x\to 0}\ o(x)/x = 0$.

Theorem 1: Let $\boldsymbol{A} = \Sigma_{i=0}^{p-1} \mathbb{E}\{Z_{n,i}X_{n-i}^t\} = \Sigma_{i=0}^{p-1} \mathbb{E}\{Z_{n+i,i}X_n^t\} = \mathbb{E}\{\overline{Z}_n X_n^t\}$, where $\overline{Z}_n = \Sigma_{i=0}^{p-1} Z_{n+i,i}$. Define $\lambda_{\min}(\boldsymbol{A}) = \min_i\{\mathrm{Re}(\lambda_i)\}$, where λ_i are the eigenvalues of the matrix $\boldsymbol{A}$ and $\mathrm{Re}(\cdot)$ denoting the real part. If $\lambda_{\min}(\boldsymbol{A}) \neq 0$, then we have

$$\lim_{\mu\to 0} \frac{1}{\mu}\left(\lim_{n\to\infty} \frac{\log(\gamma_n^{-1})}{n}\right) = 2\lambda_{\min}(\boldsymbol{A}). \tag{11}$$

Proof: A proof based on the IA can be found in the Appendix. For a proof that does not use the IA (but requires more stringent assumptions), see [21]. ■

As a result of Theorem 1, we have the first corollary addressing the stability properties of γ_n.

Corollary 1: If $\lambda_{\min}(\boldsymbol{A}) > 0$, then γ_n for small enough μ converges exponentially fast to zero at a rate that is approximately equal to $2\mu\lambda_{\min}(\boldsymbol{A})$. If $\lambda_{\min}(\boldsymbol{A}) < 0$, then γ_n for small enough μ tends exponentially fast to infinity, meaning that the algorithm is unstable. Finally, if $\lambda_{\min}(\boldsymbol{A}) = 0$, no conclusion can be derived since higher order approximations in μ are required.

Comment: In A1, we assumed that the input process $\{X_n\}$ is persistently exciting by requiring the covariance matrix $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$ to be nonsingular. It should be noted that this assumption is rather necessary for our analysis to be applicable. This is because, as explained in Corollary 1, the estimate of the convergence rate of a stable algorithm is valid if $\lambda_{\min}(A) > 0$. However, if $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$ is singular, then we can easily show that this will also be the case for the matrix $\boldsymbol{A} = \mathbb{E}\{\overline{Z}_n X_n^t\}$, and therefore, we will have $\lambda_{\min}(\boldsymbol{A}) = 0$.

Under the result of Theorem 1, we can now proceed with the estimation of the steady-state behavior of π_n.

Theorem 2: Let the matrix $\boldsymbol{A} = \Sigma_{i=0}^{p-1} \mathbb{E}\{Z_{n,i}X_{n-i}^t\} = \sum_{i=0}^{p-1} \mathbb{E}\{Z_{n+i,i}X_n^t\} = \mathbb{E}\{\overline{Z}_n X_n^t\}$, with $\overline{Z}_n = \Sigma_{i=0}^{p-1} Z_{n+i,i}$, have eigenvalues with strictly positive real parts; then

$$\lim_{\mu\to 0} \frac{1}{\mu}\left(\lim_{n\to\infty} \pi_n\right) = \sigma_w^2 \,\mathrm{trace}\{\boldsymbol{QP}\} \tag{12}$$

where $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$, and $\boldsymbol{P}$ satisfies the Lyapunov equation

$$\boldsymbol{AP} + \boldsymbol{PA}^t = \boldsymbol{R} \tag{13}$$

with $\boldsymbol{R} = \mathbb{E}\{\overline{Z}_n \overline{Z}_n^t\}$.

Proof: A proof based on the IA is presented in the Appendix. A more rigorous proof that does not rely on the IA can be found, for example, in [3, p. 107]. ■

We have now available all necessary background results to define our performance measure and specify the optimum algorithms.

III. Local Performance Measure

Since speed of convergence is the characteristic that is of interest to us, the most natural candidate for measuring performance is clearly the exponential rate of convergence of γ_n. This rate, as we can see from Theorem 1, depends on the step size μ and, for small μ, can be approximated by $2\mu\lambda_{\min}(\boldsymbol{A})$. It is this dependence on μ on which we want to elaborate.

When comparing algorithms from our class, there is clearly no reason to use the same step-size μ in each one of them. Since convergence rates depend directly on μ, this raises the question of what is a proper selection of the step sizes that can guarantee a "fair" comparison without favoring any algorithm at the expense of another. There are two possible directions we can follow to solve this problem. The first, which was proposed in [6] and [9], consists of selecting the step sizes so that the corresponding rates are equal and then consider as optimum the algorithm with the smallest steady-state excess mean square error. The second, which was proposed in [5] and also used in most signal processing applications, consists of selecting the step sizes so that the steady state excess mean square errors are equal and consider as optimum the algorithm with the highest convergence rate. Here, we are going to follow the latter direction because it will also allow for the definition of the notion of relative performance of two algorithms. However, for the local case, we can show that both methods can lead to the same final performance measure.

Let us first introduce a relative performance measure between two adaptive algorithms from the class defined by (4) and (5). Denote by $\boldsymbol{A}_i, \boldsymbol{Q}_i, \boldsymbol{R}_i, \boldsymbol{P}_i, i = 1, 2$ the corresponding matrices defined by Theorems 1 and 2 for the two algorithms. Furthermore, let μ_1, μ_2 be the two step sizes selected so that the steady-state excess mean square error of both algorithms is equal to a common value $\pi \ll 1$. From Theorem 2, we immediately conclude that

$$\mu_i = \frac{\pi}{\sigma_w^2} \frac{1}{\mathrm{trace}\{\boldsymbol{Q}_i \boldsymbol{P}_i\}} + o(\pi), i = 1, 2. \tag{14}$$

Under the constraint imposed by (14) that the two step sizes are selected to yield the same steady-state excess mean square error, we can now consider the transient parts of the two algorithms. As a relative performance measure, we can clearly define the relative number of iterations required by the corresponding transient parts to converge to zero. Since, theoretically, each algorithm requires an infinite number of steps to converge to zero, in order to find this ratio, we proceed as follows. We first compute the number of iterations required by each transient part γ_n to reach a common value $\gamma > 0$; then, take the limit of the corresponding ratio of iterations as γ tends to zero.

The number of iterations $n_i, i = 1, 2$ required by the corresponding γ_n to reach a common value γ, from the definition of the exponential convergence rate and Theorem 1, is equal to

$$n_i = \frac{\log\gamma^{-1} + o(\log\gamma^{-1})}{2\mu_i\lambda_{\min}(\boldsymbol{A}_i) + o(\mu_i)} \tag{15}$$

which using (14) takes the form

$$n_i = \frac{\log\gamma^{-1} + o(\log\gamma^{-1})}{\dfrac{\pi}{\sigma_w^2}\dfrac{2\lambda_{\min}(\boldsymbol{A}_i)}{\mathrm{trace}\{\boldsymbol{Q}_i\boldsymbol{P}_i\}} + o(\pi)}. \tag{16}$$

The relative performance, as we said, is the limit of the ratio of n_1/n_2 as γ tends to zero. We thus have

$$\lim_{\gamma\to 0}\frac{n_1}{n_2}=\frac{\dfrac{\lambda_{\min}(\boldsymbol{A}_2)}{\mathrm{trace}\{\boldsymbol{Q}_2\boldsymbol{P}_2\}}+\dfrac{o(\pi)}{\pi}}{\dfrac{\lambda_{\min}(\boldsymbol{A}_1)}{\mathrm{trace}\{\boldsymbol{Q}_1\boldsymbol{P}_1\}}+\dfrac{o(\pi)}{\pi}}. \tag{17}$$

Since we consider the local case $\pi \ll 1$, the terms $o(\pi)/\pi$ in the previous expression are negligible as compared with the remaining terms. We can thus define as the *local relative measure* of Algorithm 1 with respect to Algorithm 2 $(\mathrm{LRM}_{1,2})$ the expression

$$\lim_{\gamma\to 0}\frac{n_1}{n_2}\approx \mathrm{LRM}_{1,2}=\frac{\dfrac{\lambda_{\min}(\boldsymbol{A}_2)}{\mathrm{trace}\{\boldsymbol{Q}_2\boldsymbol{P}_2\}}}{\dfrac{\lambda_{\min}(\boldsymbol{A}_1)}{\mathrm{trace}\{\boldsymbol{Q}_1\boldsymbol{P}_1\}}}. \tag{18}$$

Since an algorithm is better (converges faster) when it requires fewer iterations to converge, this means that algorithm 1 is better than algorithm 2 if $\mathrm{LRM}_{1,2} \le 1$.

Let us now define a quantity that will constitute our final performance measure, which will refer to a single algorithm. We define as *efficacy* of an algorithm the expression

$$\mathrm{EFF}=\frac{\lambda_{\min}(\boldsymbol{A})}{2\mathrm{trace}\{\boldsymbol{Q}\boldsymbol{P}\}} \tag{19}$$

where we recall that $\boldsymbol{A}=\mathbb{E}\{\overline{Z}_n X_n^t\}, \boldsymbol{Q}=\mathbb{E}\{X_n X_n^t\}, \boldsymbol{P}$ is the solution to the Lyapunov equation (13) with $\boldsymbol{R}=\mathbb{E}\{\overline{Z}_n\overline{Z}_n^t\}$, and $\overline{Z}_n=\Sigma_{i=0}^{p-1} Z_{n+i,i}$. From (18), we can now see that the LRM can be computed using the efficacies of the two algorithms as

$$\mathrm{LRM}_{1,2}=\frac{\mathrm{EFF}_2}{\mathrm{EFF}_1} \tag{20}$$

and clearly, Algorithm 1 is better than Algorithm 2 if $\mathrm{EFF}_1 \ge \mathrm{EFF}_2$. Consequently, if we are interested in the optimum (fastest converging) algorithm in our class, this amounts to *finding the algorithm with the maximum efficacy*. This will be the subject of our next section.

A last observation regarding the efficacy is the fact that it can be directly related to the exponential convergence rate of the algorithm. Specifically, from Theorem 1 and (14), to a first-order approximation, we have that

$$\lim_{n\to\infty}\frac{\log\gamma_n^{-1}}{n}=4\,\frac{\pi}{\sigma_w^2}\,\mathrm{EFF} \tag{21}$$

where we recall that π is the steady-state excess mean square error, and consequently, π/σ_w^2 is the misadjustment. In other words, for fixed misadjustment, the efficacy is proportional to the exponential convergence rate.

IV. Locally Optimum Algorithms

In this section, we are going to maximize the efficacy over two different algorithmic classes. The first will be the general algorithmic model defined by (4) and (5) and Assumptions $\mathcal{A}_1, \mathcal{A}_2$. The second will refer to an LMS-like class where some additional constraint will be imposed on the input data, aiming in generalizing the second optimality result of [4].

A. Optimum Algorithm for the General Model

With the following theorem, we are going to show that the algorithm that maximizes the efficacy is the LMS-Newton [8], [11], [23], which therefore is the locally optimum algorithm in our class.

Theorem 3: The maximum value of the efficacy is equal to $1/N$, and it is attained if and only if $\overline{Z}_n=\alpha\boldsymbol{Q}^{-1}X_n$, where α is any positive real scalar.

Proof: Let us first introduce a very useful property concerning the trace of products of matrices. If $\boldsymbol{D},\boldsymbol{E}$ are matrices with the same dimensions, we then have

$$\mathrm{trace}\{\boldsymbol{D}^t\boldsymbol{E}\}=\mathrm{trace}\{\boldsymbol{E}\boldsymbol{D}^t\}=\mathrm{trace}\{\boldsymbol{D}\boldsymbol{E}^t\}=\mathrm{trace}\{\boldsymbol{E}^t\boldsymbol{D}\}. \tag{22}$$

We can now proceed with our proof. From the non-negative definiteness of the covariance matrix of $[\overline{Z}_n^t X_n^t]^t$, we can conclude that[2] $\boldsymbol{R}\ge\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^t$, where we recall that $\boldsymbol{A}=\mathbb{E}\{\overline{Z}_n X_n^t\}, \boldsymbol{Q}=\mathbb{E}\{X_n X_n^t\}$, and $\boldsymbol{R}=\mathbb{E}\{\overline{Z}_n\overline{Z}_n^t\}$. We have equality if and only if

$$\overline{Z}_n=\boldsymbol{A}\boldsymbol{Q}^{-1}X_n \tag{23}$$

holds in the mean square sense. Notice now that the solution $\boldsymbol{P}$ to the Lyapunov equation (13) is given by [7, p. 428] $\boldsymbol{P}=\int_0^\infty e^{-\boldsymbol{A}\tau}\boldsymbol{R}e^{-\boldsymbol{A}^t\tau}\,d\tau$, meaning that $\boldsymbol{P}$ is increasing (see footnote) in $\boldsymbol{R}$. In other words, if we replace $\boldsymbol{R}$ by $\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^t$ and denote with $\boldsymbol{P}_x$ the solution to

$$\boldsymbol{A}\boldsymbol{P}_x+\boldsymbol{P}_x\boldsymbol{A}^t=\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^t \tag{24}$$

we have $\boldsymbol{P}_x\le\boldsymbol{P}$. Multiplying from right and left with $\boldsymbol{Q}^{1/2}$ results in $\boldsymbol{Q}^{1/2}\boldsymbol{P}_x\boldsymbol{Q}^{1/2}\le\boldsymbol{Q}^{1/2}\boldsymbol{P}\boldsymbol{Q}^{1/2}$, which with the help of (22), yields

$$\mathrm{trace}\{\boldsymbol{Q}\boldsymbol{P}_x\}\le\mathrm{trace}\{\boldsymbol{Q}\boldsymbol{P}\}. \tag{25}$$

If in (24) we multiply from the right by $\boldsymbol{P}_x^{-1}$, take traces, and use (22), we obtain

$$2\mathrm{trace}\{\boldsymbol{A}\}=\mathrm{trace}\{\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^t\boldsymbol{P}_x^{-1}\}. \tag{26}$$

We know that matrices of given dimensions form a linear vector space. For this space, we can define an inner product of the form $\langle\boldsymbol{D},\boldsymbol{E}\rangle=\mathrm{trace}\{\boldsymbol{D}^t\boldsymbol{E}\}$. That $\langle\cdot,\cdot\rangle$ is indeed an inner product is easy to verify using the definition. We thus have validity of the Schwarz inequality, which for this case takes the form

$$(\mathrm{trace}\{\boldsymbol{D}^t\boldsymbol{E}\})^2\le\mathrm{trace}\{\boldsymbol{D}^t\boldsymbol{D}\}\mathrm{trace}\{\boldsymbol{E}^t\boldsymbol{E}\} \tag{27}$$

with equality if and only if $\boldsymbol{D}=\alpha\boldsymbol{E}$ for some scalar α. The Schwarz inequality combined with (22) yields

$$\begin{aligned}(\mathrm{trace}\{\boldsymbol{A}\})^2&=(\mathrm{trace}\{(\boldsymbol{P}_x^{-1/2}\boldsymbol{A}\boldsymbol{Q}^{-1/2})(\boldsymbol{Q}^{1/2}\boldsymbol{P}_x^{1/2})\})^2\\&\le\mathrm{trace}\{\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^t\boldsymbol{P}_x^{-1}\}\mathrm{trace}\{\boldsymbol{P}_x\boldsymbol{Q}\}\end{aligned} \tag{28}$$

with equality if and only if

$$\boldsymbol{A}=\alpha\boldsymbol{P}_x\boldsymbol{Q} \tag{29}$$

[2] If $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are symmetric matrices, we say that $\boldsymbol{P}_1\ge\boldsymbol{P}_2$ if the difference $\boldsymbol{P}_1-\boldsymbol{P}_2$ is non-negative definite.

for some scalar α. Since we consider stable algorithms, we have from Theorem 1 that all eigenvalues of $\boldsymbol{A}$ must have positive real part. This means that $\text{trace}\{\boldsymbol{A}\} > 0$. Now substituting (26) in (28) yields

$$\text{trace}\{\boldsymbol{A}\} \leq 2\text{trace}\{\boldsymbol{P}_x\boldsymbol{Q}\}. \tag{30}$$

Recalling that $\boldsymbol{A}$ is real (thus, complex eigenvalues appear in conjugate pairs), we have

$$N\lambda_{\min}(\boldsymbol{A}) \leq \text{trace}\{\boldsymbol{A}\}. \tag{31}$$

combining this with (25) and (30), we conclude that the efficacy is bounded from above by $1/N$.

To attain the upper limit, notice that we must have validity of (23) and (29) and equality in (31) simultaneously. Since $\boldsymbol{P}_x, \boldsymbol{Q}$ are symmetric positive definite matrices, so is the matrix $\boldsymbol{P}_x^{1/2}\boldsymbol{Q}\boldsymbol{P}_x^{1/2}$. Therefore, we can make the following diagonalization: $\boldsymbol{P}_x^{1/2}\boldsymbol{Q}\boldsymbol{P}_x^{1/2} = \boldsymbol{T}\boldsymbol{D}\boldsymbol{T}^{-1}$ with $\boldsymbol{D}$ diagonal and having real and positive diagonal elements. Multiplying from left and right by $\boldsymbol{P}_x^{-1/2}$ and $\boldsymbol{P}_x^{1/2}$, respectively, we conclude that the product $\boldsymbol{P}_x\boldsymbol{Q}$ is also diagonalizable and has real and positive eigenvalues. Thus, in order for $\boldsymbol{A} = \alpha\boldsymbol{P}_x\boldsymbol{Q}$ to correspond to a stable algorithm, we must select $\alpha > 0$. In order now for $\boldsymbol{A}$ to satisfy (31) with equality, since $\boldsymbol{A}$ has real eigenvalues, all eigenvalues of $\boldsymbol{A}$ need to be equal. However, $\boldsymbol{A} = \alpha\boldsymbol{P}_x\boldsymbol{Q}$ is diagonalizable, and thus, it can have a single eigenvalue if and only if $\boldsymbol{A} = \alpha\boldsymbol{I}_N$. Applying this relation to (23), we prove the necessity of the condition $\overline{Z}_n = \alpha\boldsymbol{Q}^{-1}X_n$. Sufficiency can be easily established by direct substitution. This concludes the proof. ■

Of course, the LMS-Newton algorithm has only theoretical interest since its application requires knowledge of the input data covariance matrix $\boldsymbol{Q}$, which is usually not available in practice. On the other hand, it is expected that by estimating this matrix using the input data, we will be able to achieve performance close to the optimum. Indeed, as we are going to see in the next section, there are several such possibilities.

There is one special case where the matrix $\boldsymbol{Q}$ is known beforehand, and the corresponding optimum algorithm is readily realizable in practice. This case is presented in the following corollary.

Corollary 2: If $\boldsymbol{Q} = \sigma_x^2\boldsymbol{I}_N$, then the optimum algorithm in the class is the LMS.

In other words, if the elements of the input data vector X_n are uncorrelated and have equal variances, no other algorithm has better performance than LMS. This corollary, in a sense, generalizes considerably the first optimality result of [4]. Specifically, in [4], it is shown that if the input data are Gaussian and white, then the optimum nonlinearity $f(x)$ in (1), when $g(x)$ is fixed to $g(x) = x$, is $f(x) = (x/c + \mu x^2)$ for some constant c. In the local case $\mu \ll 1$, this translates into the optimum nonlinearity having the form $f(x) = x$, i.e., the LMS algorithm. With our corollary, we have, in fact, shown that when the input data sequence is white, then LMS is locally optimum for a significantly larger algorithmic class and for data sequences having any marginal distribution (not necessarily Gaussian).

Comment: A noticeable characteristic of the optimum algorithm comes from the fact that its efficacy is independent of the dependency structure of the input data vector. Using (21), this suggests that the optimum convergence rate also has the same property.

B. Optimum Algorithm for an LMS Like Family

In this subsection, we will focus on LMS and some of its variants. We pay special attention to this algorithm since, because of its simplicity, low complexity, and robustness, it is a very popular candidate for real-time applications. Notice that the efficacy of LMS is $\text{EFF}_{\text{LMS}} = (\lambda_{\min}(\boldsymbol{Q})/\text{trace}\{\boldsymbol{Q}\})$. We can see that the larger the eigenvalue spread of the matrix $\boldsymbol{Q}$, the poorer LMS performs as compared with the optimum algorithm of the previous subsection.

Let us now consider the case where the elements of the regression vectors $Z_{n,i}$ are given nonlinear transformations of the corresponding elements of $X_{n-i}, i = 0, \cdots, p-1$, that is

$$\begin{aligned} \epsilon_{n,i} &= y_{n-i} - W_{n-1}^t X_{n-i} \\ W_n &= W_{n-1} + \mu \sum_{i=0}^{p-1} \epsilon_{n,i} f_i(X_{n-i}) \end{aligned} \tag{32}$$

where $f_i(x), i = 0, \cdots, p-1$, are p scalar nonlinearities, and $f_i(X_n)$ denotes the vector obtained by applying the scalar nonlinearity $f_i(x)$ to each element of the vector X_n, i.e., $f_i(X_n) = [f_i(X_n^{[1]})f_i(X_n^{[2]}) \cdots f_i(X_n^{[N]})]^t$. Examples are the signed regressor LMS using $p = 1$ and $f_0(x) = \text{sign}(x)$, the quantized regressor LMS using some quantized version of the elements, a special case of the normalized LMS with $f_0(x) = (x/c + x^2)$, etc. We now come to our last theorem that identifies the regular LMS as the optimum among all such variants for an interesting class of input signals.

Theorem 4: Let the matrix $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$ be nonsingular and the elements $X_n^{[j]}, j = 1, \cdots, N$ of the input data vector X_n have identical marginal distributions satisfying the conditional linearity constraint (CLC) as in

$$\mathbb{E}\{X_n^{[k]}|X_n^{[l]}\} = c_{k,l}X_n^{[l]} \tag{33}$$

for $k, l = 1, \cdots, N$, and $c_{k,l}$ constants. Then, the efficacy of the algorithm defined in (32) is given by

$$\text{EFF}_f = \frac{(\mathbb{E}\{X_n^{[1]}\overline{f}(X_n^{[1]})\})^2}{\mathbb{E}\{[X_n^{[1]}]^2\}\mathbb{E}\{[\overline{f}(X_n^{[1]})]^2\}}\ \text{EFF}_{\text{LMS}} \tag{34}$$

where $\overline{f}(x) = \Sigma_{i=0}^{p-1}\ f_i(x)$. The efficacy is maximized when $\overline{f}(x) = \alpha x$ with α any positive constant.

Proof: The CLC in (33) states that the conditional expectation of the kth element of X_n given its lth element must be a linear function of the lth element (consequently, $c_{k,l}$ is the correlation coefficient of $X_n^{[k]}$ and $X_n^{[l]}$; thus, $c_{k,l} = c_{l,k}$).

Let us first compute the matrix $\boldsymbol{A}$, which here takes the form $\boldsymbol{A} = \mathbb{E}\{[\Sigma_{i=0}^{p-1}\ f_i(X_n)]X_n^t\}$. We have that the k,lth element of this matrix, using conditional expectation, can be written as $\mathbb{E}\{[\Sigma_{i=0}^{p-1}\ f_i(X_n^{[k]})]X_n^{[l]}\} = \Sigma_{i=0}^{p-1}\ \mathbb{E}\{\mathbb{E}\{X_n^{[l]}|X_n^{[k]}\}f_i(X_n^{[k]})\} = c_{k,l}\mathbb{E}\{X_n^{[1]}\overline{f}(X_n^{[1]})\}$, where the last equality comes from the CLC (33) and the fact that we have the same marginal distributions. In the same way, we can show that $\mathbb{E}\{X_n^{[k]}X_n^{[l]}\} = c_{k,l}\mathbb{E}\{[X_n^{[1]}]^2\}$. From this, we conclude that

$$\begin{aligned} \boldsymbol{A} &= r\boldsymbol{Q} \\ r &= \frac{\mathbb{E}\{\overline{f}(X_n^{[1]})X_n^{[1]}\}}{\mathbb{E}\{[X_n^{[1]}]^2\}}. \end{aligned} \tag{35}$$

Substituting these relations in (13) and taking traces, we have that

$$\text{trace}\{\boldsymbol{QP}\} = \frac{N\mathbb{E}\{[\overline{f}(X_n^{[1]})]^2\}}{2r}. \tag{36}$$

Using (35) and (36) in the definition of the efficacy, we can easily show (34). To maximize the efficacy, we only need to maximize the scalar term in front of EFF_{LMS} in (34). By a simple application of the Schwarz inequality, we can see that this term is no larger that unity, and it becomes one if and only if $\overline{f}(x) = \alpha x$. Finally, we must select $\alpha > 0$ to produce a stable algorithm. This concludes the proof. ■

Let us now elaborate on the CLC defined in (33). Several well-known classes of input data processes satisfy this condition. The first is the case where X_n is composed of i.i.d. elements where $c_{k,l} = 0$ for $k \neq l$ (here, however, we have a stronger optimality of LMS because of Corollary 2). The second is the more interesting case, where the distribution of X_n is zero mean Gaussian; this is the case analyzed in [4]. The third is when the distribution of X_n is a convex combination of Gaussian distributions. Finally, if we regard the elements of the vector X_n as consecutive points of a scalar process $\{x_n\}$, then we can show that the CLC is equivalent to the Bussgang condition (which is also known from blind deconvolution techniques) [1]. Several processes, especially of the Markov type, are shown in [1] to satisfy the Bussgang condition and, consequently, (33). In the next section, we are going to present an example of such a process having uniform marginals.

Comment: Notice from (34) that by dividing any two efficacies corresponding to two different algorithms from the class, the resulting LRM depends only on the common marginal distribution of the elements of X_n and not on its actual multivariate distribution. In other words, the relative performance of any two such algorithms is independent of the dependency structure of the elements of X_n.

As an example, let us compute the LRM of the signed regressor LMS (SRLMS) with respect to the regular LMS. Using (34), we obtain

$$\text{LRM}_{\text{SRLMS,LMS}} = \frac{\mathbb{E}\{[X_n^{[1]}]^2\}}{[\mathbb{E}\{|X_n^{[1]}|\}]^2}. \tag{37}$$

If the data have Gaussian marginals, then the ratio of the two efficacies becomes $\pi/2$, whereas in the case of uniformly distributed marginals, it is equal to 4/3. This means that in the first case, LMS requires 57% fewer iterations than the SRLMS to converge for any process satisfying the CLC and having Gaussian marginals, whereas in the second case, for uniform marginals, this percentage drops to 33%. Detailed analysis of SRLMS and comparisons with LMS under Gaussian data can be found in [10]. The two algorithms were also compared in [5] for uniform i.i.d. input data.

V. Simulations

In this section, we are going to present simulation in order to compare with our theoretical conclusions obtained in the previous section. We are basically going to present two sets of experiments corresponding to the two optimality results introduced in Theorems 3 and 4.

In the first set, we simulate the following algorithms: LMS-Newton (LMS-N), RLS, and sliding window RLS (SWRLS). Specifically, we use the following adaptation formulas for each algorithm.

LMS-Newton:

$$\begin{aligned} \epsilon_n &= y_n - W_{n-1}^t X_n \\ W_n &= W_{n-1} + \mu\epsilon_n \boldsymbol{Q}^{-1} X_n \end{aligned} \tag{38}$$

with $\boldsymbol{Q} = \mathbb{E}\{X_n X_n^t\}$ being the exact covariance matrix of X_n, which is assumed to be known exactly.

RLS:

$$\begin{aligned} \epsilon_n &= y_n - W_{n-1}^t X_n \\ \boldsymbol{Q}_n &= (1-\mu)\boldsymbol{Q}_{n-1} + \mu X_n X_n^t, \boldsymbol{Q}_0 = \mu\delta\boldsymbol{I}_N \\ W_n &= W_{n-1} + \mu\epsilon_n \boldsymbol{Q}_n^{-1} X_n \end{aligned} \tag{39}$$

with the parameter δ selected to have a small value $\delta = 0.001$.

SWRLS:

$$\begin{aligned} e_{n,i} &= y_{n-i} - W_{n-1}^t X_{n-i}, \qquad i = 0, \cdots, p-1 \\ \boldsymbol{Q}_n &= \sum_{i=0}^{p-1} X_{n-i} X_{n-i}^t \\ W_n &= W_{n-1} + \mu\boldsymbol{Q}_n^{-1} \sum_{i=0}^{p-1} \epsilon_{n,i} X_{n-i}. \end{aligned} \tag{40}$$

Regarding the last algorithm, it should be noted that it is, in fact, a modification of the classical SWRLS whose estimates are given by $W_n = \boldsymbol{Q}_n^{-1} \sum_{i=0}^{p-1} y_{n-i}X_{n-i}$. It is easy to see that the algorithm in (40) reduces to the classical SWRLS when $\mu = 1$. The modification in (40) of the original algorithm was considered necessary in order to achieve control, for fixed window size p, over the excess mean square error (which is something that is not possible with the classical version). The above modification is proposed in [2] and [19], where it is also stated that this algorithm has the highest convergence rate of all algorithms belonging to the underdetermined RLS class. As we are going to see, this performance can, at best, match the performance of the LMS-Newton algorithm, which is also significantly inferior for cases where the window size is not adequately large. For our experiments, we are going to use two window values, namely, $p = 30$ (SWRLS-30) and $p = 100$ (SWRLS-100).

The input data vector X_n has the form $X_n = [x_n, x_{n-1}, \cdots, x_{n-19}]^t$, where the scalar process $\{x_n\}$ is an AR Gaussian satisfying $x_n = ax_{n-1} + v_n$ with $a = 0.9$ and $\{v_n\}$ Gaussian i.i.d. The random variable x_n is normalized to unit variance. The input sequence $\{x_n\}$ is passed through

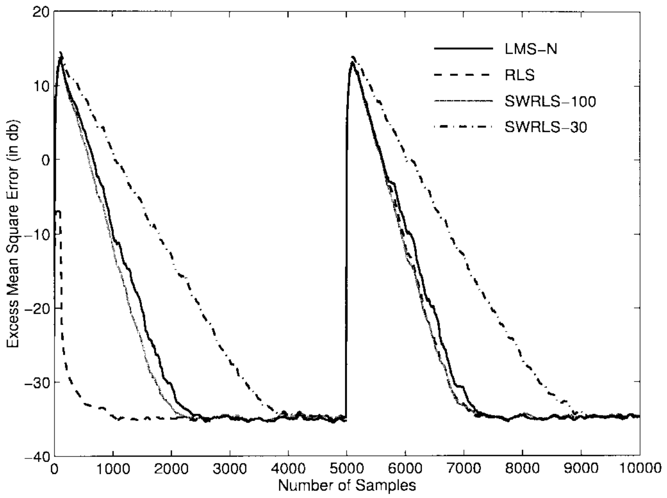


Fig. 1. Performance of adaptive algorithms during the initial transient phase and after an abrupt change of the true system for Gaussian AR data.

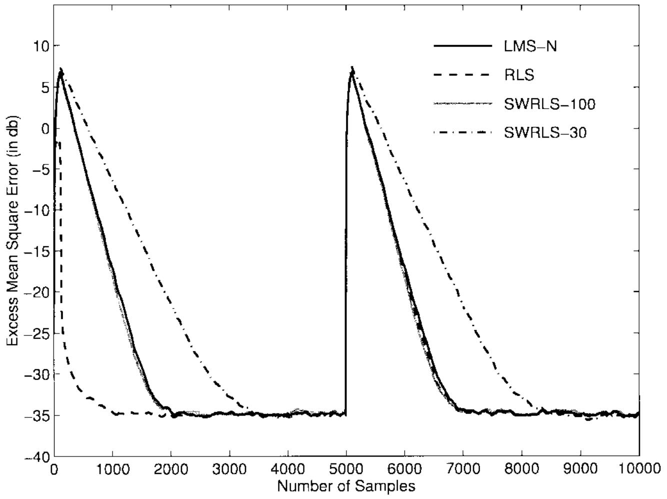


Fig. 2. Performance of adaptive algorithms during the initial transient phase and after an abrupt change of the true system for uniform i.i.d. data.

an FIR system (the vector W_*) of length $N = 20$ having elements randomly distributed in $[-1\ 1]$. Finally, to the output of the FIR system, we add white Gaussian noise $\{w_n\}$ of variance 0.01 to generate the desired response $\{y_n\}$. The step sizes are selected so that the steady-state excess mean square error becomes equal to 35 dB (misadjustment of 15 dB). This results in using, for LMS-N, RLS, and SWRLS-100, the value $\mu = 0.0032$ and, for SWRLS-30, the value $\mu = 0.0017$. We apply the four algorithms we mentioned before for a number of points equal to 10 000 with the characteristic that at time 5000, we make a change of the true model from W_* to $-W_*$. The experiment is repeated 50 times and as excess mean squared error, for every time instant n, we use the arithmetic mean of the corresponding 50 realizations of the excess squared error. The results are presented in Fig. 1.

Fig. 2 depicts the same algorithms for a similar experiment as the previous one, only here, the input sequence $\{x_n\}$ is uniform i.i.d. with variance equal to unity, and the additive noise $\{w_n\}$ is also uniform with variance 0.01. The step sizes are the same as in the previous experiment.

Before examining the two figures, let us briefly discuss what is to be expected according to our analysis. Notice first that RLS does not satisfy our assumptions during the whole test period. Specifically, the part of Assumption A1 referring to the stationarity of the regression vector $\boldsymbol{Q}_n^{-1} X_n$ is not satisfied because of the nonstationarity of the matrix $\boldsymbol{Q}_n$. Therefore, we do not expect that, during the initial transient phase, RLS will necessarily follow our conclusions. On the other hand, we expect that this will be the case when the algorithm has converged, and there is a sudden change in the model W_*. This is true because the regression vector depends only on the input data and not on the true model W_*. Let us now examine whether RLS can match the optimum performance whenever it satisfies our assumptions. Notice that if the step size μ is small, then the matrix $\boldsymbol{Q}_n$, at steady state is a good approximation to $\boldsymbol{Q}$. This means that the regression vector $Z_n = \boldsymbol{Q}_n^{-1} X_n$ is also a good approximation to $\boldsymbol{Q}^{-1} X_n$ and, according to Theorem 3, we expect that RLS will match in performance the LMS-N.

As far as the SWRLS algorithm is concerned, we have validity of our assumptions once the sliding window is covered with data; hence, this algorithm can, at best, match the performance of LMS-N. Let us examine whether this is indeed possible. From Theorem 3, we have that the necessary and sufficient condition for optimality is $\overline{Z}_n = \alpha \boldsymbol{Q}^{-1} X_n$. Since $\overline{Z}_n = \Sigma_{i=0}^{p-1} Z_{n,i}$ and, from (40), we have $Z_{n,i} = \boldsymbol{Q}_n^{-1} X_{n-i}$, this yields

$$\overline{Z}_n = \left[\sum_{i=0}^{p-1} \boldsymbol{Q}_{n+i}^{-1} \right] X_n. \tag{41}$$

The question clearly is whether the sum $\Sigma_{i=0}^{p-1} \boldsymbol{Q}_{n+i}^{-1}$ constitutes a satisfactory approximation to $\alpha \boldsymbol{Q}^{-1}$ (for some scalar α), where we recall that $\boldsymbol{Q}_n = \Sigma_{i=0}^{p-1} X_{n-i} X_{n-i}^t$. For a large enough window size p, using the law of large numbers, the matrix $\boldsymbol{Q}_n/p$ approximates well $\boldsymbol{Q}$, and thus, the sum in question is also expected to approximate well $\alpha \boldsymbol{Q}^{-1}$. For small window size p, however, $\boldsymbol{Q}_n/p$ is not a good approximation to $\boldsymbol{Q}$, and therefore, the corresponding sum does not approximate $\alpha \boldsymbol{Q}^{-1}$ well. For example, for a problem size $N = 20$, a window of size $p = 30$ might be considered small, whereas $p = 100$ might be considered adequate. In other words, SWRLS-100 is expected to match the performance of LMS-N, whereas SWRLS-30 is expected to be inferior.

Observing Figs. 1 and 2, we have that, indeed, SWRLS-100 is close to the performance of LMS-N and so is RLS after the change of the model (where it satisfies our assumptions). On the other hand, we can see that SWRLS-30 is significantly inferior to LMS-N.

In both figures, we can see that RLS, during the initial transient phase, is significantly faster than all other algorithms. In fact, it is even faster than the optimum LMS-N. As we explained above, during this period, RLS does not satisfy our assumptions; therefore, it does not necessarily follow our conclusions. The extraordinary fast convergence speed of this popular algorithm comes from the fact that it has the unique property of exactly estimating W_* in a finite number of steps when there is no additive noise present. LMS and most other adaptive algorithms do not enjoy this characteristic.

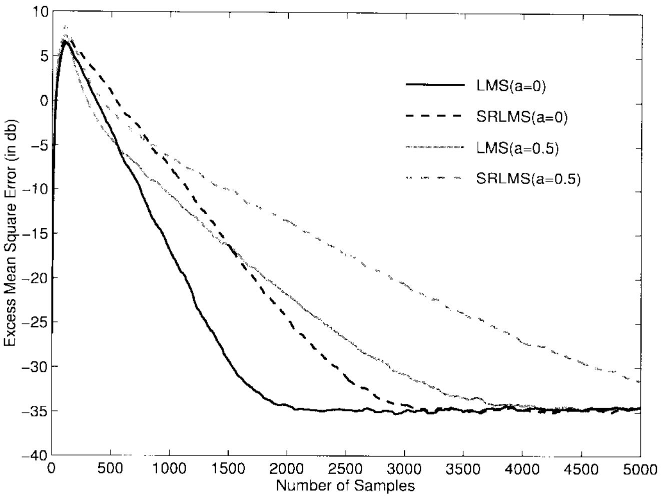


Fig. 3. Performance of LMS and signed regressor LMS during the initial transient phase for Gaussian i.i.d. and AR data.

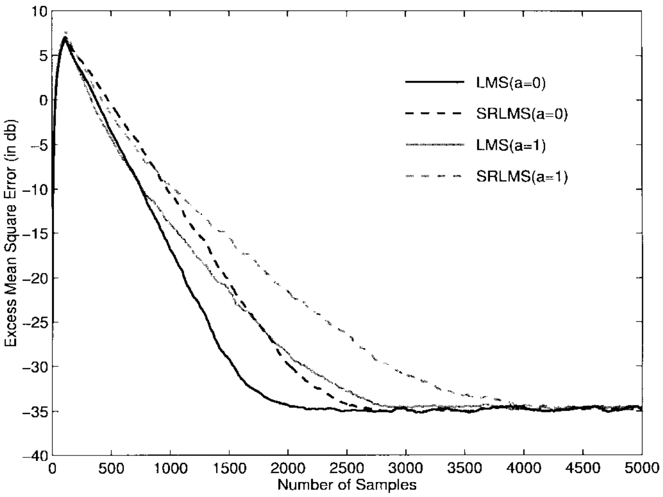


Fig. 4. Performance of LMS and signed regressor LMS during the initial transient phase for uniform i.i.d. and Markov data.

This convergence property is preserved, as was proved in [20], when the additive noise is low, and the parameter δ in (39) is small (which is the case in our experiments).

Finally, we can see that the convergence rate of the algorithms that match the LMS-N is indeed independent of the distribution and the dependency structure of the data, as was pointed out in our comment at the end of Section IV-A. This is demonstrated by the different data distributions and dependency structures used to produce the results appearing in Fig. 1 (Gaussian and highly correlated) and Fig. 2 (uniform and i.i.d.).

The second set of experiments we intend to present refers to Theorem 4. Here, we test the two most well-known algorithms in the class, namely, LMS and signed regressor LMS (SRLMS). The corresponding recursions are the following.

LMS:

$$\begin{aligned}\epsilon_n &= y_n - W_{n-1}^t X_n\\ W_n &= W_{n-1} + \mu\epsilon_n X_n.\end{aligned} \tag{42}$$

SRLMS:

$$\begin{aligned}\epsilon_n &= y_n - W_{n-1}^t X_n\\ W_n &= W_{n-1} + \mu\epsilon_n \mathrm{sign}(X_n).\end{aligned} \tag{43}$$

Fig. 3 depicts the performance of the two algorithms for AR data of the form $x_n = ax_{n-1} + v_n$, where $\{v_n\}$ is i.i.d. Gaussian and x_n is normalized to unit variance. The FIR system W_* is the same as in the previous examples, and the additive noise $\{w_n\}$ is also Gaussian with variance 0.01. Two values for the parameter a are used, namely, $a = 0$ and 0.5. The step sizes for the two algorithms, for both values of a, are $\mu = 0.0032$ for LMS and $\mu = 0.0025$ for SRLMS, producing an excess mean square error of 35 dB. The algorithms are applied to 5000 points, except here, we do not impose any change in the true model W_*. Again, the experiment is repeated 50 times in order to compute estimates of the excess mean square error.

Since the process $\{x_n\}$ is Gaussian, we have validity of the CLC (33); consequently, as was stated in Section IV-B, LMS is 56% faster than SRLMS, regardless of the dependency structure of the data. We can see in Fig. 3 that this is indeed the case. The relative performance of the two algorithms is the same for both values of the parameter a and approximates the theoretical value well.

Finally, Fig. 4 has a similar experiment as the previous one but with a uniformly distributed data process $\{x_n\}$ of unit variance. To generate a non-i.i.d. process with uniform marginals, we considered $\{x_n\}$ to be a Markov process with transition probability density $s(x_n|x_{n-1})$ equal to

$$s(x_n|x_{n-1}) = u(x_n)\Big[1 + \frac{a}{3}\, x_n x_{n-1}\Big] \tag{44}$$

where $u(x)$ denotes the uniform density with support $[-\sqrt{3}, \sqrt{3}]$ (in order for x_n to have unit variance). When $|a| \le 1$, then $s(x_n|x_{n-1})$ is a legitimate transition density for $\{x_n\}$ since it is non-negative and integrates to unity for any $x_{n-1} \in [-\sqrt{3}, \sqrt{3}]$. It is easy to show that the marginal (steady-state) density of this Markov process is indeed uniform by verifying that $u(x_n) = \int s(x_n|x_{n-1})u(x_{n-1})\, dx_{n-1}$. Furthermore, we can show that the bivariate density of x_n and x_{n-j} is of the form $s(x_n, x_{n-j}) = u(x_n)u(x_{n-j})[1 + (\frac{a}{3})^j x_n x_{n-j}]$; therefore, we can easily verify that the CLC in (33) is also valid.

For uniform marginal densities, we have seen in Section IV-B that LMS is 33% faster than SRLMS, regardless of the dependency structure of the data. Fig. 4 depicts exactly this point for two values of the parameter a in (44), namely, $a = 0$ and 1. In this experiment, the step size for LMS is $\mu = 0.0032$ and for SRLMS $\mu = 0.0027$. We can see again that the relative performance does not depend on a and that it is close to the theoretical value.

VI. Conclusion

We have presented a new analytic method for comparing constant gain adaptive signal processing algorithms. By making an asymptotic analysis of the second-order statistics of the excess error during the transient as well as the steady-state phase of the algorithm, we were able to obtain estimates of the convergence rate and the steady-state excess mean square error. This, in turn, led to the definition of a theoretically computable local performance measure (the efficacy), which is consistent with the comparison methods used in most signal processing applications. We showed that the algorithm that optimizes our measure is the Newton-LMS. This algorithm has only theoretical interest since, for its practical realization, it requires the *a priori* knowledge of the second-order statistics of the input data. Practically realizable algorithms were also presented that approximate the optimum performance very closely. Finally, limiting ourselves to an LMS-like family of algorithms, we showed that for an important class of input signals, LMS is better than any of its variants that apply the same nonlinear transformation on the elements of the regression vector.

Appendix

Before going to the proofs of Theorems 1 and 2, let us briefly introduce the notion of the Kronecker product for two matrices along with some basic properties that will be necessary for our proofs.

Let $\boldsymbol{C}, \boldsymbol{D}$ be two matrices with elements $c_{i,j}, d_{i,j}$ and of dimensions $m \times l$ and $k \times q$. The Kronecker product of $\boldsymbol{C}$ and $\boldsymbol{D}$, which are denoted as $\boldsymbol{C} \otimes \boldsymbol{D}$, is a matrix of dimensions $(mk) \times (lq)$ defined in a block form as

$$\boldsymbol{C} \otimes \boldsymbol{D} = \begin{bmatrix} c_{1,1}\boldsymbol{D} & \cdots & c_{1,l}\boldsymbol{D} \\ \vdots & \vdots & \vdots \\ c_{m,1}\boldsymbol{D} & \cdots & c_{m,l}\boldsymbol{D} \end{bmatrix}. \tag{45}$$

In addition, let vec$\boldsymbol{C}$ denote a vector of length ml that results if we place the columns of the matrix $\boldsymbol{C}$ one after the other.

Provided that the dimensions of the matrices involved are such that the operations below are valid, we have the following properties of the Kronecker product:

1) $(\boldsymbol{C} + \boldsymbol{D}) \otimes \boldsymbol{E} = \boldsymbol{C} \otimes \boldsymbol{E} + \boldsymbol{D} \otimes \boldsymbol{E}$;
2) $\boldsymbol{E} \otimes (\boldsymbol{C} + \boldsymbol{D}) = \boldsymbol{E} \otimes \boldsymbol{C} + \boldsymbol{E} \otimes \boldsymbol{D}$;
3) $(\boldsymbol{CD}) \otimes (\boldsymbol{EF}) = (\boldsymbol{C} \otimes \boldsymbol{E})(\boldsymbol{D} \otimes \boldsymbol{F})$;
4) $\mathrm{vec}\{\boldsymbol{CD}\} = (\boldsymbol{I} \otimes \boldsymbol{C})\mathrm{vec}\{\boldsymbol{D}\} = (\boldsymbol{D}^t \otimes \boldsymbol{I})\mathrm{vec}\{\boldsymbol{C}\}$.

For a proof of the above and other interesting properties of the Kronecker product, see [17, ch. 12].

We have now the following lemma concerning the eigenvalues of an expression involving Kronecker products.

Lemma 1: Let $\boldsymbol{A}$ be a square matrix of dimensions $N \times N$, where $\lambda_i, i = 1, \cdots, N$ are the corresponding eigenvalues; then, the eigenvalues of the matrix $(\boldsymbol{A} \otimes \boldsymbol{I}_N + \boldsymbol{I}_N \otimes \boldsymbol{A})$ are the N^2 numbers $\lambda_i + \lambda_j, i, j = 1, \cdots, N$.

Proof: See [17, p. 412].

We can now proceed with the proofs of our two theorems.

Proof of Theorem 1: The application of the IA will consist of assumptions that recursive estimates of any kind are independent of $\{X_n\}$ and that $\{Z_{n,i}\}, i = 0, \cdots, p-1$. Thus, to find γ_n, we have

$$\begin{aligned} \gamma_n &= \mathbb{E}\{\Gamma_{n-1}^t X_n X_n^t \Gamma_{n-1}\} = \mathbb{E}\{\Gamma_{n-1}^t \mathbb{E}\{X_n X_n^t\} \Gamma_{n-1}\} \\ &= \mathrm{trace}\{\boldsymbol{Q}\mathbb{E}\{\Gamma_{n-1}\Gamma_{n-1}^t\}\}. \end{aligned} \tag{46}$$

Since $\boldsymbol{Q}$ is constant, to study the convergence properties of γ_n, it suffices to study the convergence properties of the matrix $\mathbb{E}\{\Gamma_n \Gamma_n^t\}$ or, equivalently, its vector form $\mathrm{vec}\{\mathbb{E}\{\Gamma_n \Gamma_n^t\}\}$. Since

$$\mathrm{vec}\{\mathbb{E}\{\Gamma_n \Gamma_n^t\}\} = \mathbb{E}\{\mathrm{vec}\{\Gamma_n \Gamma_n^t\}\} = \mathbb{E}\{\Gamma_n \otimes \Gamma_n\} \tag{47}$$

we can equivalently study the vector $\mathbb{E}\{\Gamma_n \otimes \Gamma_n\}$. For this vector, we have the following recursion using (9) and stationarity of the processes $\{X_n\}, \{Z_{n,i}\}, i = 0, \cdots, p-1$:

$$\begin{aligned} \mathbb{E}\{\Gamma_n \otimes \Gamma_n\} &= \boldsymbol{F}\mathbb{E}\{\Gamma_{n-1} \otimes \Gamma_{n-1}\} \\ \boldsymbol{F} &= \boldsymbol{I}_{N^2} - \mu \boldsymbol{A} \otimes \boldsymbol{I}_N - \mu \boldsymbol{I}_N \otimes \boldsymbol{A} + \mu^2 \boldsymbol{D} \end{aligned} \tag{48}$$

where $\boldsymbol{D} = \mathbb{E}\{(\overline{Z}_n \otimes \overline{Z}_n)(X_n \otimes X_n)^t\}$ and where, again, we used the IA to separate the expectations. We now conclude that $\mathbb{E}\{\Gamma_n \otimes \Gamma_n\} = \boldsymbol{F}^n(\Delta_0 \otimes \Delta_0)$; consequently, the behavior of $\mathbb{E}\{\Gamma_n \otimes \Gamma_n\}$ is exponential and governed by the eigenvalue of $\boldsymbol{F}$ with the maximum amplitude. More precisely, we have

$$\lim_{n \to \infty} \sqrt[n]{\|\boldsymbol{F}^n\|} = \max_i |f_i| \tag{49}$$

where $\|\cdot\|$ is any matrix norm, and f_i are the eigenvalues of $\boldsymbol{F}$ [16, pp. 36–38]. Since we assumed that μ is small, we can see from (48) that the matrix $\boldsymbol{F}$ is, in fact, a perturbation of the identity matrix $\boldsymbol{I}_{N^2}$. Thus, its eigenvalues f_i satisfy $f_i = 1 - \mu\rho_i + o(\mu)$, where ρ_i are the eigenvalues of the matrix $\boldsymbol{I}_N \otimes \boldsymbol{A} + \boldsymbol{A} \otimes \boldsymbol{I}_N$ [16, pp. 74–83]. Taking logarithms in (49), using the relation $\log(1+x) = x + o(x)$, which is true for small x, using Lemma 1, then dividing by μ and taking the limit as $\mu \to 0$ proves the required relation. ∎

Proof of Theorem 2: To prove this theorem, we are going to proceed as in Theorem 1. Using the IA, we first note that $\pi_n = \mathrm{trace}\{\boldsymbol{Q}\mathbb{E}\{\Pi_{n-1}\Pi_{n-1}^t\}\}$. Using the vector form for matrices, we can easily verify that for two matrices $\boldsymbol{D}, \boldsymbol{E}$, we have $\mathrm{trace}\{\boldsymbol{D}^t\boldsymbol{E}\} = \mathrm{vec}\{\boldsymbol{D}\}^t\mathrm{vec}\{\boldsymbol{E}\}$. Using this property, we can write

$$\begin{aligned} \pi_n &= \mathrm{trace}\{\boldsymbol{Q}\mathbb{E}\{\Pi_{n-1}\Pi_{n-1}^t\}\} \\ &= \mathrm{vec}\boldsymbol{Q}^t\mathrm{vec}\mathbb{E}\{\Pi_{n-1}\Pi_{n-1}^t\} \\ &= \mathrm{vec}\boldsymbol{Q}^t\mathbb{E}\{\Pi_{n-1} \otimes \Pi_{n-1}\}. \end{aligned} \tag{50}$$

From the definition of Π_n in (9) and using again the IA and stationarity of the processes $\{Z_{n,i}\}, \{X_n\}$, we can find the recursion

$$\mathbb{E}\{\Pi_n \otimes \Pi_n\} = \boldsymbol{F}\mathbb{E}\{\Pi_{n-1} \otimes \Pi_{n-1}\} + \mu^2\sigma_w^2\mathrm{vec}\{\boldsymbol{R}\} \tag{51}$$

where we recall that $\boldsymbol{R} = \mathbb{E}\{\overline{Z}_n\overline{Z}_n^t\}$, and thus, $\mathrm{vec}\{\boldsymbol{R}\} = \mathbb{E}\{\overline{Z}_n \otimes \overline{Z}_n^t\}$, where $\boldsymbol{F}$ was defined in (48). For stable algorithms, that is, algorithms for which $\lambda_{\min}\{\boldsymbol{A}\} > 0$ and for small enough μ, we have that $\boldsymbol{F}$ has all its eigenvalues inside the unit circle. Thus, the above recursion is stable and converges to the vector

$$\lim_{n \to \infty} \mathbb{E}\{\Pi_n \otimes \Pi_n\} = \mu^2\sigma_w^2(\boldsymbol{I}_{N^2} - \boldsymbol{F})^{-1}\,\mathrm{vec}\{\boldsymbol{R}\}. \tag{52}$$

Substituting $\boldsymbol{F}$ from (48), dividing by μ, and taking the limit as $\mu \to 0$, we conclude that we can write

$$\lim_{\mu \to 0} \frac{1}{\mu}\Big(\lim_{n \to \infty} \mathbb{E}\{\Pi_n \otimes \Pi_n\}\Big) = \sigma_w^2\,\mathrm{vec}\{\boldsymbol{P}\} \tag{53}$$

where

$$\mathrm{vec}\{\boldsymbol{P}\} = (\boldsymbol{I}_N \otimes \boldsymbol{A} + \boldsymbol{A} \otimes \boldsymbol{I}_N)^{-1}\ \mathrm{vec}\{\boldsymbol{R}\}. \tag{54}$$

It turns out that the last relation is the solution to the Lyapunov equation (13) written in a vector form [7, p. 428]. Combining this with (50) yields the desired result. ■

References

[1] J. F. Barett and D. G. Lambart, "An expansion for some second order probability distributions and its application to noise problems," *IRE Trans. Inform. Theory*, vol. IT-1, pp. 10–15, 1955.

[2] B. Baykal and A. G. Konstantinides, "Underdetermined-order recursive least-square adaptive filtering: The concept and algorithms," *IEEE Trans. Signal Processing*, vol. 45, pp. 346–362, Feb. 1997.

[3] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.

[4] N. J. Bershad, "On the optimum data nonlinearity in LMS adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 69–76, Feb. 1986.

[5] J. A. Bucklew, T. G. Kurtz, and W. A. Sethares, "Weak convergence and local stability properties of fixed step size recursive algorithms," *IEEE Trans. Inform. Theory*, vol. 39, pp. 966–978, May 1993.

[6] T. A. C. M. Claasen and W. F. G. Mecklenbräuker, "Comparison of the convergence of two algorithms for adaptive FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 670–678, June 1981.

[7] R. A. De Carlo, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[8] P. S. R. Diniz, M. L. R. de Campos, and A. Antoniou, "Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor," *IEEE Trans. Signal Processing*, vol. 43, pp. 617–627, Mar. 1995.

[9] D. L. Duttweiler, "Adaptive filter performance with nonlinearities in the correlation multiplier," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 578–586, Aug. 1982.

[10] E. Eweda, "Analysis and design of a signed regressor LMS algorithms for stationary and nonstationary adaptive filtering with correlated Gaussian data," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1367–1374, Nov. 1990.

[11] B. Farhang-Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1987–2000, Aug. 1997.

[12] L. Guo and L. Ljung, "Exponential stability of general tracking algorithms," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1376–1387, Aug. 1995.

[13] ——, "Performance analysis of general tracking algorithms," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1388–1402, Aug. 1995.

[14] L. Guo, L. Ljung, and G. L. Wang, "Necessary and sufficient conditions for stability of LMS," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 761–770 June 1997.

[15] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[16] T. Kato, *Perturbation Theory for Linear Operators*. New York: Springer-Verlag, 1966.

[17] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. New York: Academic, 1985.

[18] V. J. Mathews and S. H. Cho, "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 450–454, Apr. 1987.

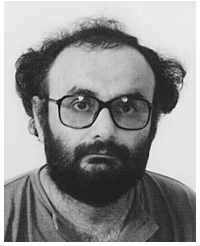
[19] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. Signal Processing*, vol. 43, pp. 444–453, Feb. 1995.

[20] G. V. Moustakides, "Study of the transient phase of the forgetting factor RLS," *IEEE Trans. Signal Processing*, vol. 45, pp. 2468–2476, Oct. 1997.

[21] ——, "Exponential convergence of products of random matrices, application to adaptive algorithms," *Int. J. Adapt. Contr. Signal Process*, to be published.

[22] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms, Stability and Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[23] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.



George V. Moustakides was born in Drama, Greece, in 1955. He received the diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1979, the M.Sc. degree in systems engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, in 1980, and the Ph.D. degree in electrical engineering from Princeton University, Princeton NJ, in 1983.

From 1983 to 1986, he held a research position at the Institut de Resherche en Informatique et Systemes Aleatoires (IRISA-INRIA), Rennes, France, and from 1987 to 1990, he held a research position at the Computer Technology Institute (CTI) of Patras, Patras, Greece. From 1991 to 1996, he was an Associate Professor with the Department of Computer Engineering and Informatics, University of Patras, and since 1996, he has been a Professor with the same department. His interests include adaptive estimation algorithms, design of classical filters, theory of optimal stopping times, and biomedical signal processing.