On the Convergence Behavior of the LMS and the Normalized LMS Algorithms

Dirk T. M. Slock, *Member, IEEE*

***Abstract*—This paper has three parts. First, we indicate that the normalized least mean square (NLMS) algorithm is a potentially faster converging algorithm compared to the LMS algorithm, when the design of the adaptive filter is based on the usually quite limited knowledge of its input signal statistics. Second, we propose a very simple model for the input signal vectors that greatly simplifies analysis of the convergence behavior of the LMS and NLMS algorithms. Using this model, answers can be obtained to questions for which no answers are currently available using other (perhaps more realistic) models. The answers thus obtained can only acclaim a qualitative value, but we give examples to illustrate that even quantitatively, they can be good approximations. Finally, we want to emphasize that the convergence of the NLMS algorithm can be speeded up significantly by employing a time-varying step size. We are able to specify *a priori* the optimal step-size sequence for the case of a white input signal with arbitrary distribution.**

I. Introduction

The LMS algorithm [1] is undoubtedly the most popular algorithm for adapting the impulse response $W = [W^0 \cdots W^{N-1}]$ of an FIR filter so as to minimize the mean-square error (MSE) between its output signal WX_k and a desired-response signal d_k. It updates the filter coefficients according to[1]

$$\begin{aligned} \epsilon_k^p &= d_k - W_{k-1}X_k \\ W_k &= W_{k-1} + \mu\epsilon_k^p X_k^H \end{aligned} \tag{1}$$

where $X_k = [x_k^H \; x_{k-1}^H \cdots x_{k-N+1}^H]^H$ is the input signal vector and μ the adaptation gain or step size. The (LMS) algorithm is a nonvanishing step-size version of a stochastic gradient algorithm. The popularity of the LMS algorithm is to a large extent due to its computational simplicity. Furthermore, it is generally felt that its behavior is quite simple to understand [1], [2] and the algorithm appears to be fairly robust against implementation errors.

In this discussion, we want to concentrate on two important characteristics of an adaptive filter: its convergence behavior and the steady-state MSE, which remains after the algorithm has converged. Exact results for both of these items are very scarce, and actually only exist for the asymptotic case of small step-size μ. Perhaps the latest results on the steady-state MSE have been obtained by Solo. In [3], [4] results are given for the steady-state MSE that are exact up to first order in μ, with scenarios for the input and desired-response signals that are considerably more general than considered before in the literature. As far as convergence behavior is concerned, the only quality that is established in [3] is exponential convergence of the filter estimate (under appropriate conditions), in the form of an upper bound (lower than 1) on the eigenvalues involved in the average algorithm dynamics. This is the only qualification on the convergence behavior that is needed for the analysis of the steady-state MSE in [3], [4]. This type of worst case approach to the analysis of the convergence dynamics (resulting in not necessarily tight bounds) has been used quite often in the control literature over the last decade or so (see, e.g., [5]–[8]), often involving the concept of persistent excitation (or regularity of the covariance matrix of the input signal vector). Exact results are available for the eigenmodes of the adaptive filter dynamics, but also only for the asymptotic case of small step size. These results involve the so-called weak convergence theory of stochastic difference equations (see, e.g., the work of Kushner [9], [10]).

When one wants to maximize the convergence speed of the LMS algorithm, a big step size is needed, and especially when one wants to address the issue of the maximum step size for stable operation of the algorithm, one needs a theory that is valid beyond an infinitesimally small step-size range. At this time, no exact theory of that nature exists. All results for big step size currently available use the so-called independence assumption. (See [41] for some recent exact analysis, that turns out to be quite involved, though.) The independence assumption specifies that the sequence of input vectors $\{X_k\}$ is an i.i.d. sequence. This assumption, though clearly violated since in the typical time series applications, X_k and X_{k-1} have $N - 1$ elements in common, simplifies the analysis significantly. The independence assumption was used early on in [11], [12] and popularized in [2]. The discrepancy between theoretical results based on this assumption and the true algorithm behavior was investigated in [13] and found to be relatively small. Extensive results based on the assumption were obtained by Gardner in [14], where further references can be found.

Manuscript received May 25, 1990; revised October 6, 1992. The associate editor coordinating the review of this paper and approving it for publication was Prof. V. John Mathews.

The author is with Institut Eurecom, 06904 Sophia Antipolis Cedex, France.

IEEE Log Number 9210109.

[1] Superscript H denotes Hermitian (complex conjugate) transpose.

A specific form of the LMS algorithm, with a reparameterized step size, is the NLMS algorithm, viz.,

$$\epsilon_k^p = d_k - W_{k-1}X_k$$
$$W_k = W_{k-1} + \bar{\mu}\epsilon_k^p(X_k^H X_k)^{-1}X_k^H \tag{2}$$

which corresponds to the choice (if X_k has only one column) $\mu_k = \bar{\mu}/\|X_k\|^2$ in the LMS algorithm. According to Tsypkin [15], this algorithm corresponds to the algorithm of Kaczmarz [16]. In the Western literature, it was first proposed by Nagumo and Noda [17]. In the control literature, it is also known as the projection algorithm [18] for the following reason. Assume the minimum MSE (MMSE) is zero or, in other words, the desired response $d_k = W^o X_k$ is the output of a FIR filter W^o of order $\leq N$, fed by the same input signal as the adaptive filter. Then we get, with $\tilde{W}_k = W^o - W_k$ being the error in the estimated filter coefficients,

$$\epsilon_k^p = \tilde{W}_{k-1}X_k$$
$$\tilde{W}_k = \tilde{W}_{k-1}[I - \bar{\mu}X_k(X_k^H X_k)^{-1}X_k^H]. \tag{3}$$

The matrix multiplying the step size is the projection matrix onto X_k in $\mathbb{R}^N$ and, for $\bar{\mu} = 1$, the matrix within the square brackets is the projection matrix onto the orthogonal complement of X_k. In other words, with $\bar{\mu} = 1$, the error component in the filter estimate along X_k is projected out exactly at time k. If N consecutive input vectors X_k are orthogonal (e.g., the signal $\{x_k\}$ contains a certain nonzero sample and $N - 1$ zero samples before and after it), then the filter converges exactly in N samples. The matrix within the square brackets has $N - 1$ eigenvalues equal to 1 and one eigenvalue equal to $1 - \bar{\mu}$. This is very much a well-controlled situation compared to the equivalent situation for the LMS algorithm, where the matrix $I - \mu X_k X_k^H$ also has $N - 1$ eigenvalues equal to 1, but the last eigenvalue is equal to $1 - \mu\|X_k\|^2$, which is not so easy to control in general. We believe this deterministic projection interpretation is a key ingredient in understanding the possible advantages of the NLMS algorithm and also illuminates immediately the potential faster convergence characteristics of the NLMS algorithm. Interestingly, the update produced by the NLMS algorithm can be interpreted as the solution to the following least squares problem

$$\min_{W_k}\left\{\|d_k - W_k X_k\|^2 + \left(\frac{1}{\bar{\mu}} - 1\right)\|X_k\|^2\|W_k - W_{k-1}\|^2\right\} \tag{4}$$

providing an interpretation to the NLMS update for $\bar{\mu} \in [0, 1]$ as a compromise between the fit to the new data and the deviation from the prior estimate, with $\bar{\mu}$ determining the relative importance of the two terms. A similar interpretation can easily be worked out for the LMS algorithm too, but now the relative weighting factor $(1/\mu) - \|X_k\|^2$ can easily take on negative values for large step sizes.

It is clear that in the LMS algorithm, one should take the step size inversely proportional to the input signal variance σ_x^2 for stability reasons. Therefore, in many applications this variance is estimated on-line (e.g., using $\hat{\sigma}_x^2(k) = \rho\hat{\sigma}_x^2(k - 1) + (1 - \rho)\|x_k\|^2$) and the step size is reparameterized as $\mu_k = \hat{\mu}/\hat{\sigma}_x^2(k)$. One could regard the NLMS algorithm as a special case of this with $\hat{\sigma}_x^2(k) = \|X_k\|^2/N$. This is basically the viewpoint taken by Bershad in [19]. However, we wish to disagree with this view and we believe that the projection interpretation is a crucial characteristic of the NLMS algorithm, the nice properties of which do not follow from the use of an arbitrary $\hat{\sigma}_k^2(k)$. In [19], the NLMS algorithm is analyzed with, apart from the independence assumption, a Gaussian distribution for the input signal vectors. However, the results in [19] pertaining to the comparison of LMS versus NLMS are only valid for small step size. In this case, it is plausible that one finds that LMS and NLMS behave quite similarly. The potential advantages of the NLMS algorithm, however, only become apparent for big step-size values. In [20], the NLMS algorithm was analyzed using the independence and Gaussian assumptions also. Due to analytical difficulties, the analysis is limited to the two cases of white noise (just one distinct eigenvalue) and two distinct eigenvalues, one bigger value and the other eigenvalue with multiplicity $N - 1$, but with arbitrary step-size values. With the step sizes in both LMS and NLMS optimized for convergence speed, Tarrab and Feuer find that the NLMS algorithm converges faster than the LMS algorithm, but to a higher steady-state MSE.

It is useful to have simple approximation techniques which allow for a straightforward analytical exploration of the qualitative behavior of a system. Such a simple analysis for the LMS algorithm was provided in [2]. This analysis led to a widespread understanding of the modal behavior of the LMS algorithm. It was based on the independence assumption. However, in fact, only the mean behavior of the filter estimates was analyzed, leading to, e.g., estimates for the range of stable operation for the step size that are way off. Though a more careful analysis had been done already in the late 1960's [11], it took a while before second-order moments in a more proper analysis of the learning curve acquired widespread attention. By now, the proper analysis of the evolution of the second-order moments in the Gaussian case (with the independence assumption) has appeared in several papers, for both the LMS [14], [21]–[23] and the NLMS algorithms [19], [20]. This analysis is relatively tractable (especially for LMS) but leads to a system in which the dynamics of the modes are coupled. Obtaining further insight into these dynamics is hard in general and therefore further elaborations have been restricted to the cases of one (white noise) or two distinct eigenvalues for the input covariance matrix. For LMS, some bounds for the adaptive system's eigenvalues in the general case were offered in [22]. For NLMS, analytical expressions for those system eigenvalues have not been provided, even in the case of only two distinct input covariance eigenvalues. In this paper, we provide a new approximation which allows for a quick hit at the modal behavior of the second-order moments. We were motivated to carry out this analysis by some unexpected observations in simulations, showing much faster convergence for some ill-conditioned input covariance matrices than for the white noise case. These observations can be explained by the analysis presented here, showing the interplay between the time constant of a mode and its relative contribution to the total MSE.

In the next section, we introduce and motivate a specific i.i.d. distribution model for the input signal vectors, and continue to analyze the convergence of the NLMS algorithm with the assumptions thus introduced. The assumptions introduced lead to such a simple analysis that results are obtained for any eigenvalue scenario. This allows us to obtain new insights into the eigenvalue-distribution-dependent convergence behavior of the NLMS algorithm. The parallel analysis for the LMS algorithm is then presented. Next, learning curves thus obtained analytically are compared with simulations. In Section III, we address the issue of using a time-varying step size in order to overcome the compromise between fast convergence speed and low steady-state MSE. The optimal step-size sequence is derived for a white but otherwise arbitrary input signal. The resulting step-size sequence agrees very well with intuition. We also consider the case of slowly drifting parameters and show how the NLMS algorithm can outperform the RLS algorithm, depending on the covariance matrix of the parameter increments. In Section IV, we summarize our findings and offer some concluding remarks. This paper is an extended version of [24], in which a simpler input signal model was used which only applied to the NLMS algorithm.

II. Convergence Analysis and Modal Behavior

We start out by introducing a simple model for the input signal vectors $\{X_k\}$.

A. A Model for the Purpose of Analysis

First, we take the independence assumption (A1) described in the previous section, assuming the vectors X_k to be independent and identically distributed (i.i.d.). Then we take a distribution for the vector X_k that we require to be as simple as possible, but consistent with the actual first- and second-order statistics of the input signal. So, consider the eigendecomposition of the covariance matrix

$$R = EX_kX_k^H = V\Sigma V^H = \sum_{i=1}^{N} \lambda_i V_i V_i^H \qquad (5)$$

where the V_i are orthonormal. We shall assume that the random vector X_k is the product of three independent variables that are i.i.d., viz.,

$$\text{(A2) } X_k = sr\mathcal{V}$$

$$\text{where} \begin{cases} \Pr\{s = \pm 1\} = \frac{1}{2} \\ r \sim \|X_k\| \\ \Pr\{\mathcal{V} = V_i\} = p_i = \dfrac{\lambda_i}{\operatorname{tr} R}, \quad i = 1, \cdots, N \end{cases} \qquad (6)$$

where tr denotes trace and $r \sim \|X_k\|$ means that r has the same distribution as the norm of the regression vector filled with samples from the original input signal. Note that

$$\sum_{i=1}^{N} \lambda_i = \operatorname{tr} R = N\sigma_x^2 \quad \text{and} \quad \sum_{i=1}^{N} p_i = 1.$$

Note also that $EX_k = 0$, $EX_kX_k^H = R$. The model (A2) decouples two aspects of the distribution of X_k. The distribution of the projection matrix $X_k(X_k^HX_k)^{-1}X_k^H$ depends solely on the "angular" distribution of X_k (the distribution of sV), not on its "radial" distribution (the distribution of r). The angular distribution is defined on a hypersphere and is obtained by integrating the distribution along radial directions. So the model (6) can be seen to be a discretization of the angular distribution into N directions. N is the minimum number of directions in order to have a nonsingular covariance matrix and hence to represent the second-order moments correctly. The description of random variables with a discrete distribution is closer in some sense to a deterministic description than a description with a continuous distribution. Considering the simplicity of the convergence analysis of the steepest-descent algorithm (the deterministic counterpart to the LMS algorithm), we may expect some benefits to accrue from the model (A2) in the convergence analysis of the (N)LMS algorithms.

We can identify the following desirable ergodic implication of the models (A2) (such implication only depends on the consistent second-order description). If the eigenvalues of the covariance matrix are different, then there are certain directions in $\mathbb{R}^N$ along which the input signal vector lies more often than along others. Consider the angle θ_{ik} with $\cos\theta_{ik} = V_i^H X_k / \|X_k\|$, then $E(\cos\theta_{ik})^2 = p_i$. In an extreme case, this means that if R is singular, then there is a subspace of $\mathbb{R}^N$ in which X_k never appears.

Though the radial distribution of X_k is irrelevant for the dynamics of the convergence of the NLMS algorithm, it plays a crucial role, however, in the convergence behavior of the LMS algorithm.

B. The NLMS Learning Curve Associated with the Model

To analyze the learning curve, we shall now assume that the MMSE is not zero (or $d_k = W^o X_k$) as in the introduction. However, we shall need the following additional assumption (typical of the system identification experiment):

$$\text{(A3)} \qquad d_k = W^o X_k + \epsilon_k^o \qquad (7)$$

where $\{\epsilon_k^o\}$ is i.i.d. with zero mean and variance ξ^o, and independent of $\{X_k\}$. We can rewrite the *a priori* (predicted) filtering error ϵ_k^p in (1) as

$$\epsilon_k^p = \epsilon_k^o + \tilde{W}_{k-1} X_k$$

$$\text{with } \tilde{W}_k = W^o - W_k. \qquad (8)$$

Hence ϵ_k^p has two independent components and we can write the learning curve (MSE) as

$$\begin{aligned}\xi_k &= E\|\epsilon_k^p\|^2 = \xi^o + E\|\tilde{W}_{k-1}X_k\|^2\\ &= \xi^o + \operatorname{tr}(R\,\mathrm{Cov}_{k-1})\end{aligned} \tag{9}$$

where $\mathrm{Cov}_k = E\tilde{W}_k^H\tilde{W}_k$. Note that Cov_k is the second moment of $\tilde{W}_k$, and hence is the sum of its variance and its mean squared. So if Cov_k converges, or equivalently (under a condition of persistent excitation (R nonsingular)) ξ_k converges, then the propagation of both the mean and the variance of $\tilde{W}_k$ must be asymptotically stable. Hence it is convenient to concentrate on the convergence of the learning curve ξ_k, since the convergence of, e.g., the mean of $\tilde{W}_k$ will automatically be subsumed in that. From the NLMS equations (2), and (8), we get

$$\tilde{W}_k = \tilde{W}_{k-1}[I - \bar{\mu}X_k(X_k^HX_k)^{-1}X_k^H] - \bar{\mu}\epsilon_k^o(X_k^HX_k)^{-1}X_k^H \tag{10}$$

which leads to $\lim_{k\to\infty}E\tilde{W}_k = 0$, at least if the learning curve converges (a sufficient condition).[2] Hence the parameter estimates are asymptotically unbiased. Taking the expectation of the outer product of (10) with itself, we get

$$\begin{aligned}\mathrm{Cov}_k &= E\left(\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\tilde{W}_{k-1}^H\tilde{W}_{k-1}\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\right)\\ &\quad + \bar{\mu}^2E\left(\|\epsilon_k^o\|^2\frac{X_kX_k^H}{\|X_k\|^4}\right)\\ &= E\left(\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\mathrm{Cov}_{k-1}\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\right)\\ &\quad + \bar{\mu}^2\xi^o\left(E\frac{1}{r^2}\right)(E\mathcal{V}\mathcal{V}^H)\\ &= E\left(\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\mathrm{Cov}_{k-1}\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\right)\\ &\quad + \frac{\bar{\mu}^2\xi^o}{\operatorname{tr}R}R\left(E\frac{1}{r^2}\right)\end{aligned} \tag{11}$$

where the cross-products of the two terms in (10) disappear because of the independent and zero mean character of ϵ^o (or of s). We introduce the diagonal elements

$$\tilde{\lambda}_i(k) = (V^H\,\mathrm{Cov}_k\,V)_{ii}, \qquad i = 1, \cdots, N. \tag{12}$$

The learning curve may now be rewritten as

$$\xi_k = \xi^o + \sum_{i=1}^{N}\lambda_i\tilde{\lambda}_i(k-1). \tag{13}$$

By premultiplying and postmultiplying (11) with V_i, we get

$$\begin{aligned}&\tilde{\lambda}_i(k)\\ &= E\left(V_i^H\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]\mathrm{Cov}_{k-1}\left[I - \bar{\mu}\frac{X_kX_k^H}{X_k^HX_k}\right]V_i\right)\\ &\quad + \frac{\bar{\mu}^2\xi^o}{\operatorname{tr}R}\lambda_i\left(E\frac{1}{r^2}\right)\\ &= \sum_{j=1}^{N}\frac{\lambda_j}{\operatorname{tr}R}\left(V_i^H[I - \bar{\mu}V_jV_j^H]\,\mathrm{Cov}_{k-1}[I - \bar{\mu}V_jV_j^H]V_i\right)\\ &\quad + \frac{\bar{\mu}^2\xi^o}{\operatorname{tr}R}\lambda_i\left(E\frac{1}{r^2}\right)\\ &= \frac{\lambda_i}{\operatorname{tr}R}(1-\bar{\mu})^2\tilde{\lambda}_i(k-1) + \tilde{\lambda}_i(k-1)\sum_{j\neq i}\frac{\lambda_j}{\operatorname{tr}R}\\ &\quad + \frac{\bar{\mu}^2\xi^o}{\operatorname{tr}R}\lambda_i\left(E\frac{1}{r^2}\right)\\ &= \left[1 - \bar{\mu}(2-\bar{\mu})\frac{\lambda_i}{\operatorname{tr}R}\right]\tilde{\lambda}_i(k-1) + \frac{\bar{\mu}^2\xi^o}{\operatorname{tr}R}\lambda_i\left(E\frac{1}{r^2}\right).\end{aligned} \tag{14}$$

We see that convergence occurs if and only if

$$\bar{\mu} \in (0, 2) \tag{15}$$

which is a condition on the step size that is independent of the eigenvalue distribution of R. The fastest convergence occurs for

$$\bar{\mu} = 1 \tag{16}$$

which corresponds to the projection interpretation discussed in the introduction. The steady-state MSE is often expressed in terms of the misadjustment, viz.,

$$M = \frac{\xi_\infty - \xi^o}{\xi^o}, \quad M_{\mathrm{NLMS}} = \frac{\bar{\mu}}{2-\bar{\mu}}\operatorname{tr}\left(RE\frac{1}{r^2}\right). \tag{17}$$

For the case $\lambda_1 = \cdots = \lambda_L = \lambda$, $\lambda_{L+1} = \cdots = \lambda_N = 0$, we find from (13), (14)

$$\begin{aligned}\xi_k - \xi^o &= \left[1 - \bar{\mu}(2-\bar{\mu})\frac{1}{L}\right](\xi_{k-1} - \xi^o)\\ &\quad + \bar{\mu}^2\xi^o\frac{\operatorname{tr}R}{L}E\frac{1}{r^2}.\end{aligned} \tag{18}$$

This result coincides with the result that one can obtain using the Gaussian assumption [19], [20]. The result of (18) also coincides, for $L = N$ (the white noise case), with the result one can obtain using only the assumptions (A1), (A2) and that the distribution of x_k is symmetric with zero mean. This coincidence is determined by the fact that for a white symmetric distribution, the projection matrix $X_kX_k^H/X_k^HX_k$ averages out to the identity matrix, and the fact that the driving term only depends on the radial distribution of X_k (which is modeled correctly in (A2)). When

[2] An alternative approach would be to consider the unknown W^o as having a prior distribution with mean W_{-1} and some finite covariance matrix Cov_{-1}, typically a multiple of the identity matrix.

$L = N$, one may use the series expansion

$$
\begin{aligned}
E\frac{1}{r^2} &= \frac{1}{Er^2}E\frac{1}{1-\left(1-\dfrac{r^2}{Er^2}\right)}\\
&= \frac{1}{N\sigma_x^2}\sum_{i=0}^{\infty}E\left(1-\frac{r^2}{Er^2}\right)^i\\
&= \frac{1}{N\sigma_x^2}\left(1+0+\frac{\nu_x-1}{N}+O\left(\frac{1}{N^2}\right)\right)\\
&= \frac{1+\dfrac{\nu_x-1}{N}}{N\sigma_x^2}+O\left(\frac{1}{N^3}\right)\\
&= \frac{1}{N\sigma_x^2}\frac{1}{1-\dfrac{\nu_x-1}{N}}+O\left(\frac{1}{N^3}\right)\\
&= \frac{1}{\sigma_x^2(N+1-\nu_x)}+O\left(\frac{1}{N^3}\right)\\
&\approx \frac{1}{\sigma_x^2(N+1-\nu_x)}
\end{aligned}
\tag{19}
$$

where $\nu_x = Ex_k^4/\sigma_x^4$ measures the kurtosis of the input signal. The approximations in (19) hold if $\nu_x << N$. ν_x varies from 1 for a binary distribution, to 1.8 for a uniform distribution, to 3 for a Gaussian distribution, to ∞ for a Cauchy distribution. In the Gaussian case, the approximate expression in (19) holds exactly, even for $2 < L < N$, with L replacing N.

C. Modal Behavior of the NLMS Learning Curve

We shall now analyze the noiseless case ($\xi^o = 0$) in more detail. To do so, we need to make some assumptions about initial conditions. In absence of any *a priori* knowledge of the optimal filter W^o, one normally takes the initial value $W_{-1} = 0$. Also, the representative situation is the one in which all $\tilde{\lambda}_k(-1)$ are equal (the unknown W^o has components of equal magnitude along all eigenvectors of R, maximum entropy assumption). Taking also into account that, with $W_{-1} = 0$, $E\|\epsilon_0^p\|^2 = \sigma_d^2 = Ed_k^2$, a possible choice for W^o to meet the constraints just mentioned is

$$
W^o = \sqrt{\frac{\sigma_d^2 - \xi^o}{\mathrm{tr}\,R}}\;1_N V^H \tag{20}
$$

where $1_N = [1 \cdots 1]$. We shall take this choice in simulations. We can reexpress the learning curve in the noiseless case as

$$
\xi_k = f_k\sigma_d^2 \tag{21}
$$

with

$$
f_k = \sum_{i=1}^{N}(1-\alpha p_i)^k p_i \tag{22}
$$

and $\alpha = \bar{\mu}(2-\bar{\mu}) \in (0, 1)$, $p_i = \lambda_i/\mathrm{tr}\,R$. The learning curve is determined by f_k and to study its convergence as a function of the eigenvalue distribution, it is better to express it explicitly in terms of independent degrees of freedom (remember: $\Sigma_{i=1}^N p_i = 1$), viz.,

$$
\begin{aligned}
f_k = f_k(p_1, \cdots, p_{N-1}) &= \sum_{i=1}^{N-1}(1-\alpha p_i)^k p_i\\
&+ \left(1-\alpha\left(1-\sum_{i=1}^{N-1}p_i\right)\right)^k\left(1-\sum_{i=1}^{N-1}p_i\right).
\end{aligned}
\tag{23}
$$

The behavior of the learning curve is illustrated in Fig. 1.

One can see for example in Fig. 1 that initially the learning curve is concave as a function of eigenvalue distribution. This is true in general, as is readily deduced from (23) for $k = 0, 1$. Because of symmetry, the MSE attains its maximum for the uniform eigenvalue distribution. As time progresses however, the white noise case will not remain the worst. We shall find the last moment k_0 for which white noise gives the largest MSE by finding the last moment for which the Hessian of f_k remains negative definite. Therefore consider the first derivatives

$$
\begin{aligned}
\frac{\partial f_k}{\partial p_i} &= (1-\alpha p_i)^{k-1}(1-\alpha(k+1)p_i)\\
&- \left(1-\alpha\left(1-\sum_{l=1}^{N-1}p_l\right)\right)^{k-1}\\
&\cdot\left(1-\alpha(k+1)\left(1-\sum_{l=1}^{N-1}p_l\right)\right),\\
&i = 1, \cdots, N-1, \quad k \geq 1
\end{aligned}
\tag{24}
$$

and the second derivatives

$$
\begin{aligned}
\frac{\partial^2 f_k}{\partial p_i \partial p_j} &= -\alpha(1-\alpha p_i)^{k-2}k(2-\alpha(k+1)p_i)\,\delta_{ij}\\
&- \alpha\left(1-\alpha\left(1-\sum_{l=1}^{N-1}p_l\right)\right)^{k-2}k\\
&\cdot\left(2-\alpha(k+1)\left(1-\sum_{l=1}^{N-1}p_l\right)\right),\\
&i, j = 1, \cdots, N-1, \quad k \geq 2
\end{aligned}
$$

So we get for the Hessian, evaluated at the uniform eigenvalue distribution,

$$
\begin{aligned}
H_k &= \left[\frac{\partial^2 f_k}{\partial p_i \partial p_j}\right]_{i,j=1}^{N-1}\Bigg|_{p_i=1/N}\\
&= -\alpha\left(1-\frac{\alpha}{N}\right)^{k-2}k\left(2-\frac{\alpha}{N}(k+1)\right)\\
&\cdot\left(I_{N-1} + 1_{N-1}^H 1_{N-1}\right).
\end{aligned}
\tag{26}
$$

So, H_k is negative definite for

$$
k \leq k_0(\alpha, N) = \left\lfloor\frac{2N}{\alpha} - 1\right\rfloor. \tag{27}
$$

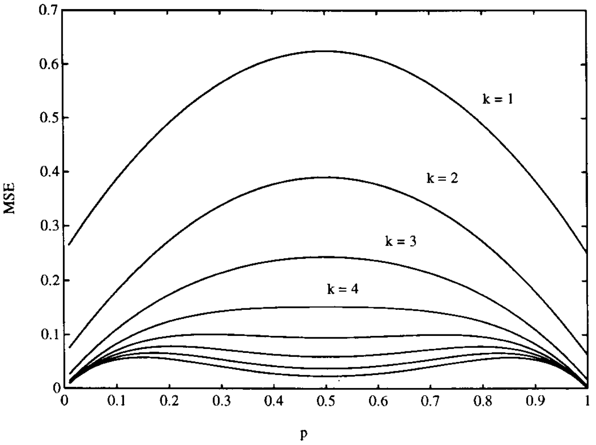


Fig. 1. Behavior of the learning curve for $N = 2$, $\bar{\mu} = 0.5$, $\text{tr}\, R = 1$, as a function of $p = p_1$, at consecutive time instances $k = 1, \cdots, 8$.

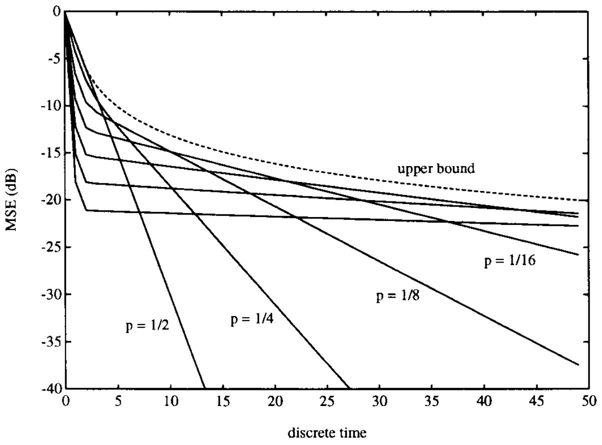


Fig. 2. Behavior of the learning curve for $N = 2$, $\bar{\mu} = 0.5$, $\text{tr}\, R = 1$, as a function of time k, for $p = p_1 = 2^{-i}$, $i = 1, \cdots, 7$.

For the example in Fig. 1, we get $k_0 = 4$, which agrees with the figure. For the reduction in MSE at k_0, we get

$$f_{k_0} = \left(1 - \frac{\alpha}{N}\right)^{\lfloor 2N/\alpha - 1 \rfloor} \xrightarrow{N/\alpha \to \infty} e^{-2} \tag{28}$$

which also agrees with Fig. 1. So the reduction at $k = k_0$ is only about 10 dB. However, this is the worst case. The learning curve for the example of Fig. 1 is displayed in Fig. 2 for different eigenvalue distributions. One can see that for any time instant k, it is possible to find an eigenvalue distribution for which the MSE at that time is lower than for the white noise case. Such extreme (increasingly ill conditioned) eigenvalue distributions give a rapid initial decrease of the MSE, but an exceedingly slow asymptotic convergence after a certain "knee" has been reached. Whether the MSE reduction achieved before the knee is useful depends upon the application. The main point we want to draw attention to is the interplay between the convergence speed of a mode and its contribution to the total MSE, as revealed in (22): in an average situation, the slowest modes have the smallest contribution to the MSE.

One notes in Fig. 1 that for $k > k_0$, we get two peaks in the MSE, whose abscissas move as a function of time. One may wonder what the MSE surface looks like in general. Intuitively, due to the special form of f_k, it is clear that for $k > k_0$, we get N peaks positioned symmetrically in the hyperplane $\Sigma_{i=1}^{N} p_i = 1$, with $N - 1$ p_i's equal to some value p, and the Nth p_i equal to $1 - (N - 1)p$. To obtain information about this value p, consider the fact that the first derivatives have to vanish at such a distribution, viz.,

$$\left.\frac{\partial f_k}{\partial p_i}\right|_{p_i = p} = 0 = (1 - \alpha p)^{k-1}(1 - \alpha(k+1)p) - (1 - \alpha(1 - (N-1)p))^{k-1} \cdot (1 - \alpha(k+1)(1 - (N-1)p)). \tag{29}$$

On the other hand, the Hessian has to be negative definite, which implies from (25) the following bound:

$$p \leq \frac{2}{\alpha(k+1)}. \tag{30}$$

This means that for large k, the second terms in the expressions for f_k, $\partial f_k / \partial p_i$, and $\partial^2 f_k / \partial p_i \partial p_j$ behave as the exponential decay $(1 - \alpha)^k$ and hence become negligible compared to the first terms. Considering the derivative again in (29), this means that p has to make the first term zero or hence

$$p = \frac{1}{\alpha(k+1)} \tag{31}$$

for large values of k. Plugging this value into (23), we get that any learning curve is asymptotically upper bounded by the following curve:

$$f_k \leq \frac{(N-1)e^{-1}}{\alpha(k+1)} \quad \text{for large } k \tag{32}$$

which is also shown in Fig. 2 for that particular example. Note that this inverse proportionality with time has nothing to do with a similar law for the step size in the case of stochastic approximations (we consider the noiseless case here).

D. The Learning Curve of the LMS Algorithm

Paralleling the steps for the NLMS algorithm, we find the following string of expressions:

$$\tilde{W}_k = \tilde{W}_{k-1}[I - \mu X_k X_k^H] - \mu \epsilon_k^o X_k^H \tag{33}$$

$$\text{cov}_k = E([I - \mu X_k X_k^H]\, \text{Cov}_{k-1}\, [I - \mu X_k X_k^H]) + \mu^2 \xi^o R \tag{34}$$

$$\tilde{\lambda}_i(k) = \left[1 - \mu\left(2 - \mu \frac{Er^4}{\text{tr}\, R}\right)\lambda_i\right]\tilde{\lambda}_i(k-1) + \mu^2 \xi^o \lambda_i. \tag{35}$$

From (35), one determines that the LMS algorithm converges if and only if

$$\mu \in \left(0,\ 2\frac{\operatorname{tr} R}{Er^4}\right) \tag{36}$$

and the fastest convergence is obtained for

$$\mu = \frac{\operatorname{tr} R}{Er^4}. \tag{37}$$

In general, we have $\operatorname{tr} R/Er^4 < 1/\operatorname{tr} R$ and for the Gaussian case in particular, we get

$$\frac{Er^4}{\operatorname{tr} R} = \operatorname{tr} R + 2\frac{\sum_{i=1}^{N} \lambda_i^2}{\operatorname{tr} R}. \tag{38}$$

The misadjustment is given by

$$M_{\text{LMS}} = \frac{\mu \operatorname{tr} R}{2 - \mu \dfrac{Er^4}{\operatorname{tr} R}}. \tag{39}$$

We may note that the quantities in (36)–(39) are quite dependent on the distribution of X_k.

For the case $\lambda_1 = \cdots = \lambda_L = \lambda$, $\lambda_{L+1} = \cdots = \lambda_N = 0$, we find

$$\xi_k - \xi^o = \left[1 - \mu\left(2 - \mu\frac{Er^4}{\operatorname{tr} R}\right)\frac{\operatorname{tr} R}{L}\right](\xi_{k-1} - \xi^o) + \mu^2\xi^o\frac{(\operatorname{tr} R)^2}{L} \tag{40}$$

which can be rewritten in the Gaussian case (for r), or if $L = N$ (white noise case), as

$$\xi_k - \xi^o = \left[1 - \mu\left(2 - \mu \operatorname{tr} R\frac{L + \nu_x - 1}{L}\right)\frac{\operatorname{tr} R}{L}\right] \cdot (\xi_{k-1} - \xi^o) + \mu^2\xi^o\frac{(\operatorname{tr} R)^2}{L}. \tag{41}$$

These results coincide with those of [25], [14] for the white noise case (and a general zero-mean distribution for the input signal), and coincide also with the results that can be easily derived from the discussion of the Gaussian case in [25], [14]. In the white noise case, the coincidence is due to the fact that $X_kX_k^H$ averages out correctly to R in our model, and the fact that $E\|X_k\|^2X_kX_k^H$ is a multiple of the identity matrix, with the multiple depending solely on the radial distribution of X_k (which is modeled correctly in A2)). In the Gaussian case, (41) leads to the following range of stable operation:

$$\mu \in \left(0,\ \frac{2}{\operatorname{tr} R}\frac{L}{L+2}\right). \tag{42}$$

From (35), it is clear that the discussion of the modal behavior of the NLMS algorithm also applies to the LMS algorithm, with just a different definition of the gain factor α.

We have modeled the distribution of X_k as the product of two marginal distributions, the "radial" distribution and the "angular" distribution, and we have discretized the angular distribution. For the NLMS algorithm, this factorization has no repercussions for the exactness of the resulting analysis (only the discretization introduces an approximation). For the LMS algorithm, the radial distribution is of paramount importance, not only because of possible unboundedness issues, but also because of a possible destabilizing interaction due to the dependence of $\|X_k\|$ on the "angle" of X_k (see [26] for some analysis of this aspect). As far as the former (unboundedness) aspect is concerned, one may note from (35) or (41) that the LMS algorithm cannot work for signals with unbounded fourth moments. However, since signals are bounded in practice, this should not be taken too literally. For the NLMS algorithm on the other hand, a basic result of our analysis is that (15) is a necessary and sufficient condition for exponential covariance of the MSE (assuming a persistent excitation). In spite of the assumptions (A1)–(A3) we made in arriving at this result, this conclusion can be shown (see, e.g., [8]) to hold for general signal distributions.

E. Verification of the Results and Discussions

Due to our assumptions, especially (A1) and (A2), one may wonder about the relevance of these results to the true algorithm behavior. One cannot expect our model to be quantitatively accurate for say, the case of Gaussian input signals. Nevertheless, in Fig. 3, we compare simulations of the NLMS and LMS algorithms, and the learning curves predicted by our theory for NLMS. We consider two cases. The input signal used in the simulations is a Gaussian first-order autoregressive (AR(1)) process with pole $a = 0$ and $a = 0.9$ in the respective cases. The filter order used is $N = 20$ and the eigendecomposition for the covariance matrix of size 20 for the AR(1) process was computed in order to obtain the theoretical learning curves. The step size used was $\bar{\mu} = 1$ for the NLMS algorithm, which seemed to be optimal in all cases. For the LMS algorithm, some experimentation with the step size yielded the (approximately) optimal values $\mu = 1.0/\operatorname{tr}(R)$ for the case $a = 0$, and $\mu = 0.3/\operatorname{tr}(R)$ for the case $a = 0.9$. Note the capability of the theoretical learning curves to closely predict the behavior of the actual curves. Actually, the (simulation) curves for the NLMS algorithm (with fixed $\bar{\mu} = 1$) and the LMS algorithm (with μ optimized for the particular input signal) do not differ very much (over this short time span), though the NLMS algorithm converges consistently faster than the LMS algorithm. This was one of the main points of [20]. One may also note that the theoretical curves for the NLMS algorithm seem to correspond more closely to the simulated learning curves for the LMS algorithm (with optimized μ for each particular AR pole a, thus not $\mu = \bar{\mu}/\operatorname{tr} R$) than those for the NLMS algorithm. A similar remark holds when one compares the largest eigenvalue in the propagation of $\text{Cov}_{N,T}$ as given by our model (see (14)),

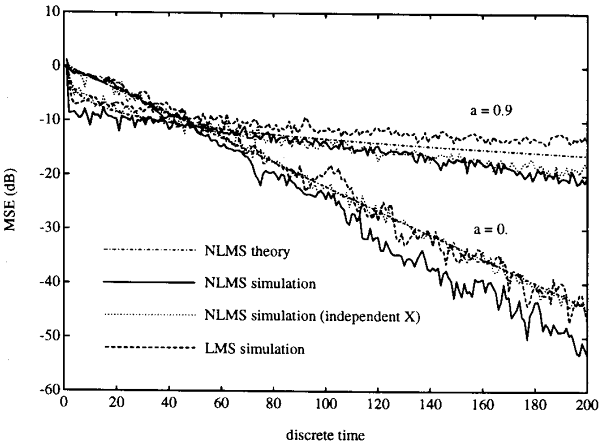


Fig. 3. Monte Carlo simulations (100 runs) of the NLMS and LMS algorithms, compared with a simulation of the NLMS algorithm with independent regressor vectors, and the NLMS learning curve predicted from (21), (22) for a Gaussian AR(1) input process with pole $a = 0$ and $a = 0.9$, respectively, and $N = 20$, $\xi^o = 0$. The (constant) step sizes are set for fastest convergence in all algorithms.

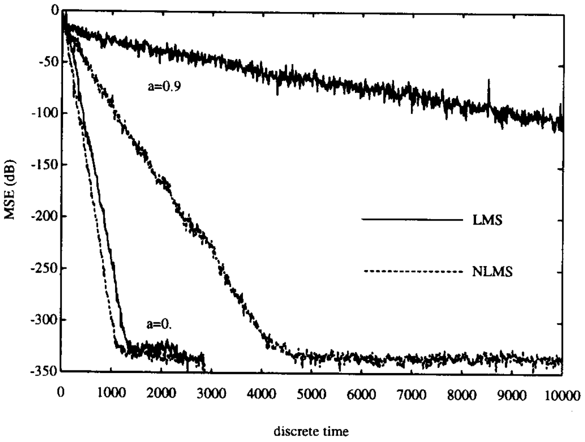


Fig. 4. Monte Carlo simulations (single run, ten consecutive samples averaged into one) of the NLMS and LMS algorithms with the step sizes set for fastest convergence. The conditions are the same as in Fig. 3, but the simulations are run over a longer time interval.

with the eigenvalues given in [20, figs. 4 and 5] for the cases considered there.

Actually, also shown in Fig. 3 are the simulation results of the NLMS algorithm with independent regression vectors X_k. These simulations correspond precisely to the exact analytical results for the Gaussian case, but still with the independence assumption, as can be found in [19], [20] [simulations are used because the analytical results have not been worked out in (and are not easily obtainable from) [19], [20] for the input signal considered here]. So the issue is how much closer these learning curves are to the learning curves of the actual NLMS algorithm (no independent X_k), than the curves resulting from our theory. For the white noise case ($a = 0$), the learning curve for the NLMS algorithm with independent X_k coincides with the curve predicted from (21), (22) (and also with the LMS algorithm with optimal step size). However, all these curves lie slightly above the curve for the actual NLMS algorithm. In the colored case ($a = 0.9$), the NLMS algorithm with independent X_k describes the knee behavior during the important transient period in which all modes are active, in a way that is quite similar to the curve resulting from our theory. The knee behavior of the actual NLMS algorithm turns out to be a bit more pronounced. Asymptotically, when the slowest eigenmode dominates, the curve for the NLMS algorithm with independent X_k lies in between the curve for the actual NLMS algorithm and the curve predicted by (21), (22), indicating that the independence assumption leads to a value for the slowest eigenmode that is in between the actual value and the value provided by our theory.

To bring out the differences between the slowest eigenmodes of the NLMS and LMS algorithms more clearly, we have performed the simulations shown in Fig. 4. For white noise ($a = 0$), we find the small convergence speed advantage of the NLMS algorithm over the LMS algorithm (with optimized step size though!) that has been discussed in [25], [20]. In the noiseless case considered here, the learning curve converges to -335 dB, the floor due to roundoff errors in the double-precision floating-point representation. Apparently, some time after convergence, the filter coefficients lock onto their correct value. However, the colored input signal example ($a = 0.9$) reveals the big potential advantage in convergence speed of the NLMS algorithm over the LMS algorithm (a factor 10!); this with the LMS stepsize being optimized, leads in fact to an unrealistic comparison.

The condition (15) for convergence of the NLMS algorithm is independent of the input signal statistics and corresponds to the actual necessary and sufficient condition for convergence. However, the accuracy of the theoretical predictions for the range of stable step-size values is quite different for the LMS algorithm. The analysis of Gardner [14] for the Gaussian case is exact, except for the independence assumption (A1). So for the Gaussian AR(1) examples considered above, one can determine the optimal μ which will minimize the largest system eigenvalue for a given AR pole a. For $a = 0$, we find $\mu^{\text{opt}} = 0.9/\text{tr } R$ (eigenvalue of slowest mode $\lambda_{\max} = 0.955$), while $\mu = 1/\text{tr } R$ is quite close to optimal. For $a = 0.9$, we find approximately $\mu^{\text{opt}} = 1/\text{tr } R$ with a sharp minimum at $\lambda_{\max} = 0.995$. However, when simulating the LMS algorithm with a Gaussian AR(1) input signal ($N = 20$, $a = 0.9$), $\mu = 1/\text{tr } R$ is not at all stable, $\mu = 1/3 \text{ tr } R$ is barely stable, while $\mu = 0.3/\text{tr } R$ is about optimal (the optimal μ and the stability bound for μ appear to be quite closer than the factor 2 suggested by (36), (37)). On the other hand, the stability bound for μ is computed in [14], [22] to be the smallest value that satisfies

$$\mu = \frac{2}{\sum_{i=1}^{N} \lambda_i (1 - \mu\lambda_i)^{-1}} \tag{43}$$

which gives the maximum value $\mu = 2/(1 + 2/N)\mathrm{tr}\, R$ for white noise, and the minimum value $\mu = 2/3\ \mathrm{tr}\, R$ for the limiting case of R of rank one. For the AR(1) example with $a = 0.9$, it gives the bound $\mu < 1.06/\mathrm{tr}\, R$. The approximation [22, eq. (32)] (considered tight in [22]) gives the bound $1.09/\mathrm{tr}\, R$, while the bound resulting from our analysis (36), (38) is $1.15/\mathrm{tr}\, R$. Hence the bound provided by our analysis corresponds closely to the bound based on the exact analysis of the Gaussian case (but with the independence assumption). However, both bounds are quite optimistic w.r.t. the actual bound which is about $0.3/\mathrm{tr}\, R$.

Another example is $x_k = (-1)^k + n_k$, where n_k is white noise with a negligible variance. Note that R is approximately of rank one and that the signal x_k is approximately periodic with period 2. Our analysis (36) leads to the bound $\mu < 2/\mathrm{tr}\, R$. An exact analysis with or without the independence assumption leads to the same bound.

The conclusion we can draw from the considerations above is that the overall approximation error due to the independence assumption (A1) appears to dominate additional inaccuracies due to the distribution assumption (A2).

As a last remark in this context, we want to come back to the issue of viewing the NLMS algorithm as an "adaptive" LMS algorithm with $\mu_k = \bar{\mu}/N\hat{\sigma}_x^2(k)$ and $\hat{\sigma}_x^2(k) = \|X_k\|^2/N$ in particular. This view is not very appropriate since, if $\hat{\sigma}_x^2(k)$ would be any other estimator of σ_x^2 and in particular have a long time constant (compared to N), then the resulting algorithm would have properties which are quite different from those of the NLMS algorithm (in particular, diverge for $\bar{\mu} = 1$ on the AR(1) process considered above), since the algorithm would resemble the LMS algorithm with $\mu = \bar{\mu}/\mathrm{tr}\, R$.

Finally, some comments on the misadjustment M. For the NLMS algorithm, the expression in (17) appears to be little dependent on the eigenvalue distribution of R, especially for high filter orders or signal distributions with low kurtosis. We have not done extensive tests to check this weak dependence (see Fig. 6 below for one example though). Gardner [14] finds the following corresponding expression for the LMS algorithm (using Gaussian signals, see also [22]):

$$M = \frac{\eta}{2 - \eta}; \qquad \eta = \mu \sum_{i=1}^{N} \frac{\lambda_i}{1 - \mu\lambda_i} \tag{44}$$

which gives $\eta = \mu\ \mathrm{tr}\, R$ for small μ. The expression for M in (44) corresponds closely to (and may even be more accurate than) a recently proposed refined expression for M in [27]. The authors of [27], who do not seem to be aware of Gardner's work, however, use a not so rigorous argument in their derivation. We have checked our result (39) for the example treated in [27, fig. 2] in which case the input signal is a sinusoid plus white noise. Though the frequency of the sinusoid is not mentioned in [27], with $\omega = \pi/10$ and, e.g., $\mu = 0.015$, we find $M = 2.15$ which is exactly the experimental value determined in [27, fig. 2]. We may note that the expression (39) gives in any case a higher value for the misadjustment than the Nehorai estimate mentioned in [27], which corresponds to $\eta = \mu\ \mathrm{tr}\, R$ in (44).

III. Step-Size Optimization

A. Motivation of the Problem and the Line of Attack

Considering the MSE, there are two conflicting requirements on the step size. It should be large to have fast dynamics and hence fast forgetting of the initial parameter settings (or of parameter changes, see further). On the other hand, a large step size means fast dynamics combined with a significant amplification of the driving term, which results in a large steady-state MSE. Most existing work on step-size optimization has considered a constant step size. Two fairly easy approaches to optimize such a constant step size are either to minimize the maximum of the absolute value of the eigenvalues in (14) (maximum convergence speed) or to minimize the steady-state MSE. The latter approach is used most often in practice since after the initial convergence, which is a temporary phenomenon, one has to live with the steady-state MSE. This consideration leads to a small step size and hence slow convergence. Bershad [28] has considered choosing the step size to minimize the MSE at the end of a given interval. This is a quite meaningful criterion in applications such as channel equalization for data communications. Here, the actual data transmission only begins after a start-up period at the end of which one would like to have a predictable performance, with the start-up period being as short as possible. For this criterion, the resulting optimal step size is a complicated function of various parameters.

Our main point here is, that whichever optimization point of view one takes, one can do significantly better with a time-varying step size. In practical implementations, this is what algorithm designers do, usually in the form of a piecewise constant step size with, e.g., two values, one for rapid convergence and one that will determine the steady-state MSE. The critical point then is the choice of the transition time. We propose here a specific time-varying step-size sequence that is optimal for certain scenarios, and depends on only one parameter. We shall indicate that the choice of this parameter is not too critical.

It is clear that the optimal step-size sequence depends on the eigenvalue distribution of R. For a general eigenvalue distribution, the optimal step size is complicated to determine and depends critically on various parameters such as the eigenvalues, the MMSE, the optimal filter coefficients. The analysis simplifies considerably for the white noise case. Also, the eigenvalues of R are usually unknown in practice and one may consider white noise, by lack of any further information. Furthermore, as we saw in the previous section, the white noise case is not the uniformly best case and hence may in some (admittedly a bit skewed) sense be viewed as an average case.

B. Derivation and Analysis of an Optimal Step-Size Sequence

With the step-size $\overline{\mu}$ being replaced by a time-varying $\overline{\mu}_k$ in the NLMS algorithm (2), the learning curve for the white noise case is determined as (see (18))

$$\xi_k = \xi^o + \tilde{\lambda}(k-1)\operatorname{tr} R$$

$$\tilde{\lambda}(k) = \left(1 + \frac{-2\overline{\mu}_k + \overline{\mu}_k^2}{N}\right)\tilde{\lambda}(k-1) + \frac{\overline{\mu}_k^2 \xi^o}{N \operatorname{tr} R} \tag{45}$$

where we used the approximation $E(1/r^2) \approx 1/\operatorname{tr} R$ (see (19)) since its consequences are of little importance here. The expressions in (45) are valid, even if assumption (A2) is dropped. So the distribution of the white input signal x_k can be arbitrary as long as it is zero mean and symmetric for positive and negative values. Optimizing recursively, assume we have the optimal step-size sequence up to time

$$k-1: \{\overline{\mu}_0^*, \cdots, \overline{\mu}_{k-1}^*\}.$$

Then we can replace $\tilde{\lambda}(k-1)$ in (45) by $\tilde{\lambda}^*(k-1)$ and optimize $\tilde{\lambda}(k)$ w.r.t. $\overline{\mu}_k$. We find

$$\overline{\mu}_k^* = \frac{\tilde{\lambda}^*(k-1)}{\tilde{\lambda}^*(k-1) + \dfrac{\xi^o}{\operatorname{tr} R}} = 1 - \frac{\xi^o}{\xi_k^*} \tag{46}$$

which leads to

$$\tilde{\lambda}^*(k) = \left(1 - \frac{1}{N}\right)\tilde{\lambda}^*(k-1) + \frac{1}{N}\left((\tilde{\lambda}^*(k-1))^{-1} + \left(\frac{\xi^o}{\operatorname{tr} R}\right)^{-1}\right)^{-1}. \tag{47}$$

So $\tilde{\lambda}^*(k)$ is given as a convex combination of two functions of $\tilde{\lambda}^*(k-1)$: the first function alone would lead to constant $\tilde{\lambda}^*(k)$, while the second function would lead to $\tilde{\lambda}^*(k)$ being inversely proportional to time. One can also find a recursion for the $\overline{\mu}_k^*$ by computing $\tilde{\lambda}^*(k-1)$ in terms of $\overline{\mu}_k^*$ from (46), and substituting in (47). One finds upon a rearrangement of terms

$$\overline{\mu}_k^* = \overline{\mu}_{k-1}^* \frac{1 - \dfrac{\overline{\mu}_{k-1}^*}{N}}{1 - \dfrac{(\overline{\mu}_{k-1}^*)^2}{N}}. \tag{48}$$

For the initialization we assume the common practice in absence of *a priori* knowledge: $W_{-1} = 0$ (otherwise redefine $d_k - W_{-1}X_k$ to be the desired-response signal). Then we have $\xi_0 = \xi^o + \tilde{\lambda}(-1)\operatorname{tr} R = \sigma_d^2$, which leads to the initialization

$$\overline{\mu}_0^* = 1 - \frac{\xi^o}{\sigma_d^2}. \tag{49}$$

We get $\overline{\mu}_0^* = 1$ for $\xi^o = 0$, which leads to $\overline{\mu}_k^* \equiv 1$ (from (48)). Indeed, one should use maximum convergence speed in the noiseless case. On the other hand, $\xi^o = \sigma_d^2$ leads to $\overline{\mu}_0^* = 0$ and hence $\overline{\mu}_k^* \equiv 0$ which also agrees with intuition: equality of the MMSE to the variance of the desired response means that no improvement is possible by adapting the filter. The sequence $\{\overline{\mu}_k^*\}$ is plotted in Fig. 5 for various initial values $\overline{\mu}_0^*$.

Initially, we have $1 - \overline{\mu}_k^* << 1$ and one may derive from (48) that $\overline{\mu}_k^*$ behaves initially as

$$\frac{\overline{\mu}_k^*}{1 - \overline{\mu}_k^*} \approx \left(1 - \frac{1}{N}\right)^k \frac{\overline{\mu}_0^*}{1 - \overline{\mu}_0^*} \tag{50}$$

thus, we have an exponential decay of $\overline{\mu}_k^*/(1 - \overline{\mu}_k^*)$. On the other hand, one can consider the series expansion

$$\frac{1}{\overline{\mu}_k^*} = \frac{1}{\overline{\mu}_{k-1}^*} + \frac{1}{N} + O(\overline{\mu}_{k-1}^*). \tag{51}$$

From (48) we see that $\{\overline{\mu}_k^*\}$ is a strictly decreasing sequence for $\overline{\mu}_0^* \in (0, 1)$ and converges to zero. So (51) leads to the asymptotic approximation

$$\overline{\mu}_k^* \approx g_k = \frac{1}{k/N - b}, \qquad k >> N \tag{52}$$

for some constant b. One can verify that the convergence to this approximation is according to

$$\frac{\overline{\mu}_k^* - g_k}{\overline{\mu}_k^*} \sim \frac{\ln k}{k} \to 0. \tag{53}$$

The constant b is asymptotically irrelevant, but plays a role in the finite-time behavior of $\{g_k\}$. A suitable constant b can be determined by requiring, e.g., $g_{k_0} = \overline{\mu}_{k_0}^*$ for some time k_0. Different initial values $\overline{\mu}_0^*$ will lead to different values for b and hence a perturbation of the initial value $\overline{\mu}_0^*$ will merely correspond to a shift in time of the convergence process.

C. Approximation and Performance Evaluation

The optimal step-size $\overline{\mu}_k^*$ we found, makes sense intuitively. Indeed, initially Cov_k and hence ξ_k are dominated by the mean of $\tilde{W}_k$ and hence we have a large step size to reduce this mean as fast as possible. As the algorithm approaches convergence, the step size should be reduced to reduce the variance of $\tilde{W}_k$. It is a well-known result from stochastic gradient algorithm theory that a necessary and sufficient condition for the convergence of Cov_k to zero is [29], [30]

$$\sum_{k=0}^{\infty} \mu_k = \infty, \qquad \sum_{k=0}^{\infty} \mu_k^2 < \infty \tag{54}$$

and, with our model for $\{X_k\}$, the same conditions hold for $\{\overline{\mu}_k\}$. The typical example for such a sequence is $\overline{\mu}_k = 1/k$. One can see that $\overline{\mu}_k^*$ is asymptotically of this form and satisfies (54). However, for fast convergence it is important to deviate from the $1/k$ behavior initially. In a practical implementation, one may wish to replace $\overline{\mu}_k^*$ computed from (48) by an approximation. The following approximation captures the main features of $\overline{\mu}_k^*$ (see Fig.

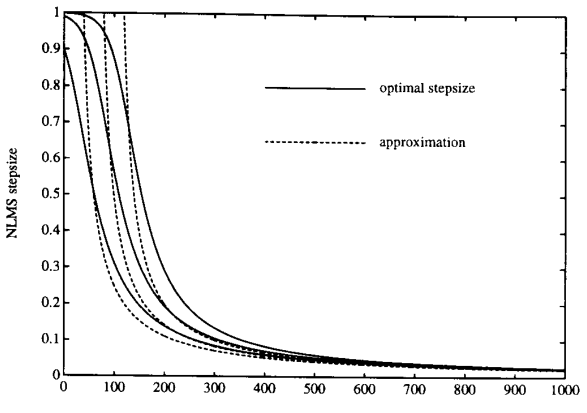


Fig. 5. Graphs of the optimal step-size sequence $\{\bar{\mu}_k^*\}$ for $N = 20$ and $\bar{\mu}_0^* = 0.9, 0.99, 0.999$, and of the approximation (55) with $c = 2, 4, 6$.

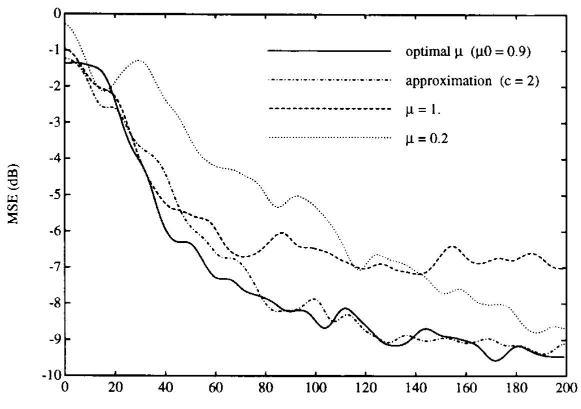


Fig. 6. Simulated learning curves (100 runs, smoothed) of the NLMS algorithm for Gaussian white noise with $N = 20$, $\sigma_d^2 = 1.0$, $\xi^o = 0.1$ for the following cases: optimal step-size $\bar{\mu}_k^*$ from (48) with $\bar{\mu}_0^* = 0.9$, approximation (55) with $c = 2$, $\bar{\mu}_{\min} = 0$, and fixed step-sizes $\bar{\mu} = 1$ and $\bar{\mu} = 0.2$, respectively.

5, where $\bar{\mu}_{\min} = 0$)

$$\bar{\mu}_k = \begin{cases} 1, & 0 \le k \le cN \\ \max\left\{\bar{\mu}_{\min}, \dfrac{1}{1 - c + k/N}\right\}, & k > cN \end{cases}. \tag{55}$$

The degree of freedom c is related to the degree of freedom $\bar{\mu}_0^*$ in the optimal step-size sequence. A related practical step-size sequence was proposed in [30] for the LMS algorithm. Note that we have introduced a certain lower bound $\bar{\mu}_{\min}$ to keep the adaptive algorithm alive so that it can track possible changes in the optimal parameters.

Some simulation results are presented in Fig. 6. The learning curves for the optimal step-size sequence and the approximation (55) are compared and seen to correspond quite closely. The choice of a constant step-size $\bar{\mu}_k \equiv 1$ for maximum speed of convergence performs equally well initially, but leads to a steady-state MSE of -7 dB $=$ -10 dB $+$ 3 dB, which corresponds closely to the misadjustment $M = 1.1$ predicted by (17), (19). The choice of a constant step-size $\bar{\mu}_k \equiv 0.2$ to reach a steady-state level at about $k = 200$ (the optimal $\bar{\mu}$ of [28], namely, the optimal constant $\bar{\mu}$ to minimize ξ_{200}), leads to a value for ξ_k that is not optimal for $k = 200$ and even less so for $k < 200$ or $k > 200$.

D. Comparison with the LMS Algorithm

With the (constant) step size being optimized for convergence speed in both the NLMS and LMS algorithms, Tarrab and Feuer [20] conclude that while the NLMS algorithm converges faster than the LMS algorithm, it has a higher MSE in steady-state (higher by a factor $1 + O((\nu_x - 1)/N)$). This last conclusion is also drawn in [19], where it is attributed to the fact that the NLMS algorithm uses less *a priori* knowledge (e.g., about tr R) than the LMS algorithm. Tarrab and Feuer find in particular (when the step size is set for maximum convergence speed) that the steady-state MSE for the NLMS algorithm increases significantly with ill-conditioning, while this dependence appears to be very weak for the LMS algorithm (see [31] though for a critique on this way of comparing things).

In Fig. 7, we provide a comparison of the NLMS and LMS algorithms with a time-varying step size. The filter order we have used, $N = 3$, is very low so that the misadjustment noise amplification due to $E(1/r^2)$ can be expected to be substantial. The particular step size we have used for the NLMS algorithm is of the form (55). Since the input signal is far from being white (a Gaussian AR(1) signal with pole $a = 0.9$), a step size of the form (55) is not optimal, but is still helpful. Judging from (14), it appears to be a good idea to use $c = O(\sigma_x^2/\lambda_{\min})$ ($\lambda_{\min}$ being the smallest eigenvalue of R) in (55) to give the slowest mode a chance to progress substantially before the step size starts to decrease. On the other hand, the influence of the slowest mode may be of little significance w.r.t. the near-convergence contribution of the fastest modes. We have, quite arbitrarily, chosen $c = 10$. For the LMS algorithm, we have also taken the step size to be of the form indicated in (55), but multiplied with some $\mu_{\max}$ which leads to the fastest convergence in the noiseless case (considering that $\bar{\mu} = 1$ leads to the corresponding situation for the NLMS algorithm). As can be seen in Fig. 7, taking the same c for LMS as for NLMS leads to a huge discrepancy between the convergence behavior of the two algorithms. Therefore, we have also tried to favor the LMS algorithm by giving it a much larger $c = 100$. Still, the faster convergence speed of the NLMS algorithm predicted from Fig. 4 leads to an overall better performance for the NLMS algorithm. Exploiting this faster convergence speed and using a time-varying step size, one can "turn down" the step size in the NLMS algorithm long before the LMS algorithm nears convergence, so that the disadvantage of a higher steady-state MSE which would result from using a constant step size, can be overcome.

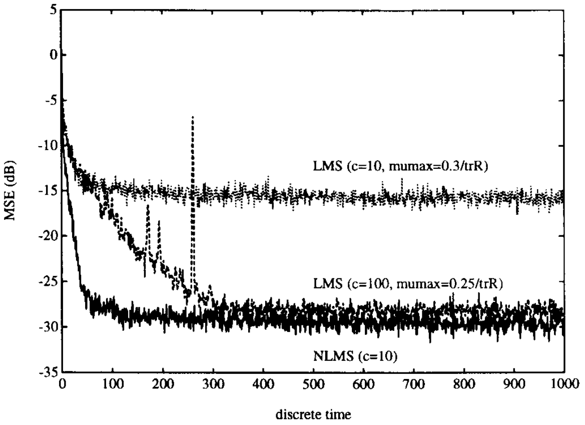


Fig. 7. Simulated learning curves (100 runs) of the NLMS and LMS algorithms with a Gaussian AR(1) input signal with pole $a = 0.9$, and $N = 3$, $\sigma_d^2 = 0$ dB, $\xi^o = -30$ dB. In all cases, the step size is time varying and of the form (55) with $\bar{\mu}_{\min} = 0$. For the NLMS algorithm, $c = 10$. For the LMS algorithm, we have used $c = 10$ and $c = 100$, and $\mu_k = \mu_{\max}\bar{\mu}_k$ with the value of $\mu_{\max}$ as indicated in the figure.

We may note that though the LMS algorithm may be stable (converging) on the average over a long period of time, it will in general be unstable (diverging) some portion of the time. The duration of such an instability is a random variable with some distribution and has some nonzero probability to take on large values, in which case the algorithm may diverge in practice (overflow). Such phenomena become more pronounced as the stepsize gets closer to the stability boundary. The step-size values used in Fig. 7 are really pushed for fastest convergence and hence are quite close to this boundary, which explains the glitches appearing in the learning curve for the second LMS simulation (even though 100 runs were averaged).

E. Comparison to RLS, and Influence of Eigenvalue Distribution

In comparison, the recursive least squares (RLS) algorithm has the following learning curve (asymptotically, for $k >> N$) [32]:

$$\xi_k^{\text{RLS}} = \xi^o \left(1 + \frac{N}{k+1}\right) \tag{56}$$

independently of the eigenvalue distribution of R. For the NLMS algorithm in the case of white input noise, the optimal stepsize sequence leads to the following asymptotic behavior of the learning curve (via (46), (52)):

$$\xi_k^* = \xi^o \left(1 + \frac{N}{k - N(b+1)}\right). \tag{57}$$

So basically, the learning curve for the optimal NLMS algorithm in the white noise case is asymptotically just a delayed version of the RLS learning curve. Of course, for time $k = O(N)$, there may be significant differences. Another aspect is the asymptotic behavior of the NLMS learning curve in case the stepsize is optimized for a white input signal, but the actual input signal is colored. Then the stepsize is suboptimal. It is difficult to asses the resulting performance degradation in general. One special case is $\lambda_1 = \cdots = \lambda_L > 0, \lambda_{L+1} = \cdots \lambda_N = 0$. This leads to identical behavior (for ξ_k, not for Cov_k) to the case of L parameters with uniform eigenvalue distribution. One can verify (using the ODE approximation of difference equations with vanishing variation) that the learning curve is asymptotically given by

$$\xi_k = \xi^o \left(1 + \frac{N}{k}\frac{1}{2 - L/N}\right). \tag{58}$$

So the misadjustment is asymptotically suboptimal by a factor $N/(L(2 - L/N))$, but also decays inversely proportionally with time. In general the learning curve is asymptotically dominated by the mode associated with the smallest nonzero eigenvalue λ_m of R. One can verify the following asymptotic behavior of the learning curve (using the results from the modal analysis above):

$$\begin{aligned} \frac{1}{2}\sigma_x^2 < \lambda_m < \sigma_x^2: &\quad \xi_k = \xi^o \left(1 + \frac{1}{k}\frac{(\lambda_m/\sigma_x^2)^2}{2\lambda_m/\sigma_x^2 - 1}\right) \\ \lambda_m = \frac{1}{2}\sigma_x^2: &\quad \xi_k = \xi^o \left(1 + \frac{\ln k}{4k}\right) \\ \lambda_m < \frac{1}{2}\sigma_x^2: &\quad \xi_k - \xi^o \sim \left(\frac{1}{k}\right)^{2\lambda_m/\sigma_x^2}. \end{aligned} \tag{59}$$

Hence, the convergence slows down with increasing eigenvalue disparity.

F. An Alternative Normalization

The NLMS algorithm is obtained from the LMS algorithm by reparameterizing the step size as $\mu_k = \bar{\mu}_k/\|X_k\|^2$. This leads to an algorithm for which the stability condition is $\bar{\mu}_k \in (0, 2)$, a condition that is independent of the signal characteristics. Another possible reparameterization with similar properties is

$$\mu_k = \frac{1}{\bar{\bar{\mu}} + \|X_k\|^2} \tag{60}$$

where $\bar{\bar{\mu}}_k \geq 0$ is a sufficient (though not necessary) condition for stability. This step-size choice can be motivated as follows. Whereas the NLMS algorithm is designed based on the deterministic mechanism of projections, the LMS algorithm with stepsize as in (60) is designed based on stochastic considerations. Consider the RLS algorithm

$$W_k = W_{k-1} + \frac{\epsilon_k^p}{1 + X_k^H R_{k-1}^{-1} X_k} X_k^H R_{k-1}^{-1} \tag{61}$$

where

$$R_k = \mu I_N + \sum_{i=0}^{k} X_i X_i^H = R_{k-1} + X_k X_k^H$$

is the sample covariance matrix. Now approximate R_{k-1} in (61) by $R_{k-1} = \bar{\bar{\mu}}_k I_N$, then we get

$$W_k = W_{k-1} + \frac{\epsilon_k^p}{\bar{\bar{\mu}}_k + \|X_k\|^2} X_k^H \tag{62}$$

which is exactly the LMS algorithm with step size μ_k of the form (60). For instance, one could choose

$$\bar{\bar{\mu}}_k = \mu + \sum_{i=0}^{k-N} \|x_i\|^2$$

which is the (N, N) entry of R_{k-1}. Then the step size in (60) becomes

$$\mu_k = \frac{1}{\mu + \sum_{i=0}^{k} \|x_i\|^2} \approx \frac{1}{\mu + k\sigma_x^2} = \frac{1}{\mu' + k/N} \frac{1}{\text{tr } R} \tag{63}$$

which is related to (55) (with $\mu' = 1$, $c = 0$) through the transformation $\mu_k = \bar{\mu}_k/\text{tr } R$. One could obtain a LMS algorithm with nonvanishing gain by starting from the exponentially weighted RLS algorithm. The *a posteriori* error $\epsilon_k = d_k - W_k X_k$ for the algorithm with step size as in (60) can be written as

$$\epsilon_k = \frac{\bar{\bar{\mu}}_k}{\bar{\bar{\mu}}_k + \|X_k\|^2} \epsilon_k^p. \tag{64}$$

From the connection with the RLS algorithm (and hence the Kalman filter), we know that $\bar{\bar{\mu}}_k$ reflects the fact that ϵ_k^p does not give a noise-free measurement of $\tilde{W}_{k-1}$ (assuming $\xi^o > 0$, see (8)). Despite these interesting interpretations for the step size in (60), we cannot really recommend the resulting LMS algorithm, since its proper design requires *a priori* knowledge about R. For instance, to determine the necessary condition on $\bar{\bar{\mu}}_k$ for stability, or to choose a $\bar{\bar{\mu}}$ to achieve a given steady-state MSE, requires (partial) knowledge of R. A similar conclusion is drawn in [33].

G. *Tracking Drifting Parameters*

Finally, we investigate the case of time-varying parameters. To do so, we make the following assumptions about the desired response and optimal parameters:

$$\begin{aligned} d_k &= W_{k-1}^o X_k + \epsilon_k^o \\ W_k^o &= W_{k-1}^o + U_k \end{aligned} \tag{65}$$

where $\{\epsilon_k^o\}$ and $\{U_k\}$ are independent of each other and of $\{x_k\}$, and are independently distributed with zero mean and variances ξ_k^o and Q_k, respectively. This leads to

$$\begin{aligned} \tilde{W}_k &= W_k^o - W_k \\ \epsilon_k^p &= \tilde{W}_{k-1} X_k + \epsilon_k^o \end{aligned} \tag{66}$$

and using our model, we find

$$\tilde{W}_k = \tilde{W}_{k-1}\left(I - \bar{\mu}_k \frac{X_k X_k^H}{X_k^H X_k}\right) - \frac{\bar{\mu}_k}{\|X_k\|^2} \epsilon_k^o X_k^H + U_k$$

$$\tilde{\lambda}_i(k) = \left(1 - \bar{\mu}_k(2 - \bar{\mu}_k)\frac{\lambda_i}{\text{tr } R}\right)\tilde{\lambda}_i(k-1) + \left(\frac{\bar{\mu}_k}{\text{tr } R}\right)^2 \xi_k^o \lambda_i + \bar{\lambda}_i(k)$$

$$\xi_k = \xi_k^o + \sum_{i=1}^{N} \lambda_i \tilde{\lambda}_i(k-1) \tag{67}$$

where $\bar{\lambda}_i(k) = [V_H Q_k V]_{ii}$. With time-invariant $\bar{\mu}$, ξ^o, and Q, we find for the steady-state MSE

$$\begin{aligned} \xi_{\text{NLMS}} &= \xi^o + \frac{\bar{\mu}}{2 - \bar{\mu}} \xi^o + \frac{\text{tr } R}{\bar{\mu}(2 - \bar{\mu})} \text{tr } Q \\ &\approx \xi^o + \frac{\bar{\mu}}{2} \xi^o + \frac{\text{tr } R}{2\bar{\mu}} \text{tr } Q \end{aligned} \tag{68}$$

where the approximation holds for small $\bar{\mu}$. The three terms in (68) are the MMSE, the estimation noise, and the lag noise, respectively. One could agree (as in [3], [4]) that in the approximation of the lag noise term, one should keep the terms of $O(1)$ and $O(\mu)$ since the estimation noise term is $O(\mu)$. However, the inclusion of those terms only leads to higher order (in the misadjustment) perturbation terms in the expressions in (69) resulting from the optimization problem considered now. Namely, optimization of ξ_{NLMS} in (68) w.r.t. $\bar{\mu}$ leads to

$$\begin{aligned} \bar{\mu}^* &= \sqrt{\frac{\text{tr } R \text{ tr } Q}{\xi^o}} \\ \xi_{\text{NLMS}}^* &= \xi^o + \sqrt{\xi^o \text{ tr } R \text{ tr } Q}. \end{aligned} \tag{69}$$

A parallel analysis of the RLS algorithm with exponential weighting (weighting factor λ) leads to the steady-state MSE [34]

$$\xi_{\text{RLS}} = \xi^o + \frac{1 - \lambda}{2} N\xi^o + \frac{1}{2}\frac{1}{1 - \lambda} \text{tr } RQ \tag{70}$$

which holds for small enough $1 - \lambda$. Optimization w.r.t. λ leads to

$$\begin{aligned} (1 - \lambda^*)N &= \sqrt{\frac{N \text{ tr } RQ}{\xi^o}} \\ \xi_{\text{RLS}}^* &= \xi^o + \sqrt{\xi^o N \text{ tr } RQ}. \end{aligned} \tag{71}$$

The ratio of the minimum steady-state misadjustments for both algorithms is

$$\frac{M_{\text{NLMS}}^*}{M_{\text{RLS}}^*} = \sqrt{\frac{\text{tr } R \text{ tr } Q}{N \text{ tr } RQ}}. \tag{72}$$

So which algorithm is better depends on R and Q. Here are two examples:

$$\begin{aligned} Q = R&: \quad \text{NLMS is better} \\ Q = R^{-1}&: \quad \text{RLS is better.} \end{aligned} \tag{73}$$

We may note that for a given model for the time-varying parameters as shown in (65), the Kalman filter is the optimal adaptive algorithm (in the Gaussian case) and the RLS and NLMS algorithms are just two different approximations of it (see also [35], [36]).

A similar comparison was made between the LMS and RLS algorithms in [37], and a similar conclusion was drawn. In [37], it was objected though that the optimal μ^* thus obtained from the analysis would often be bigger than the stability bound. However, this means that the μ^* thus obtained is too big for the analysis to hold, because the true optimal step size corresponds to a stable algorithm of course. On the other hand, such an argument does not make sense for the NLMS algorithm and in any case, for small enough Q of a certain structure as discussed above, also the LMS algorithm can outperform the RLS algorithm.

IV. Concluding Remarks

We have emphasized the deterministic interpretation of the NLMS algorithm as a (relaxed) projection algorithm. This property implies that the NLMS algorithm is not just stable on the average, but also deterministically at every sample instant (in contrast to the LMS algorithm). For analysis purposes, we have introduced a simple stochastic model for the input signal vectors, which despite its simplicity has allowed us to derive some useful results. One of these is a better insight in the qualitative convergence behavior of the algorithm as a function of the eigenvalue distribution of R.

Another point of the paper was the emphasis on the use of a time-varying step size to speed up the convergence. Though the original stochastic gradient algorithms were equipped with such step sizes, the usefulness of time-varying step sizes seems to have been forgotten in the emphasis of recent years on adaptation. General guidelines exist in the literature on stochastic gradient algorithms, on how to choose a step-size sequence. However, to the author's knowledge, an optimal step-size sequence as obtained in (48) appears to be a first result of this nature. Another possible improvement may come from the variation in time of the order N of the FIR filter. Such techniques have been used for years in the modem industry and have recently been treated in the literature [38]. Substantial gains in convergence speed can be expected if the lower order filter used initially captures most of the possible MSE reduction.

The choice of the NLMS algorithm over the LMS algorithm is not only a way out of the problem of missing *a priori* knowledge of R (or just tr R). We have seen that the convergence of the LMS algorithm is a nontrivial issue for input signals with an ill-conditioned covariance matrix or with distributions with unbounded support. In the first case, no existing analysis offers accurate predictions of the range of stable operation for the step size (often necessitating a very conservative design in practice). Though the second case is perhaps mostly of theoretical interest only, a practical circumstance that is quite closely related is a situation with wildly varying (as a function of time) input variance σ_x^2. Though this poses no problem for the NLMS algorithm, an adaptive (to circumvent the lack of *a priori* knowledge) LMS algorithm employing an estimate $\hat{\sigma}_k$ with a large time constant (compared to N) may show unacceptably large growths in the variance of the parameter estimates and the filter output. In summary, the tracking dynamics of the NLMS algorithm appear to be (significantly) less sensitive to a variety of input signal distribution aspects than holds for the LMS algorithm.

Though many algorithm designers are aware of advantages of the NLMS algorithm w.r.t the LMS algorithm, the NLMS algorithm does not seem to be widely used in practice. The reason for this is that the computational complexity of the division $\bar{\mu}\epsilon_k^p/\|X_k\|^2$ would be incompatible with present generation DSP's, which usually have a hardware multiplier, but not a divider. However, though we wish not to go into detail here, there are many ways of obtaining this quotient within an acceptable relative precision of say 1% with few computations, by using series expansions, table lookups, etc. Also, the squared norm $\|X_k\|^2$ can of course be computed recursively with just one or two multiplications per update.

We have seem that for tracking steadily drifting parameters, either one of the NLMS and RLS algorithms can be the better one, depending on the covariance matrix of the parameter increments, and the covariance matrix of the input signal (see [39], [40] for some analysis in the case of deterministic parameter variations, leading to similar conclusions). However, it is clear that the RLS algorithm always wins in the initial convergence, and a similar conclusion would be true for the problem of tracking occasionally jumping parameters [34], if proper mechanisms could be designed for detecting such jumps, the shape of the window in the LS criterion could be properly adjusted after the detection of a jump, and such changing window shapes could be accommodated in a recursive LS algorithm.

References

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

[2] B. Widrow *et al.*, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151-1162, Aug. 1976.

[3] V. Solo, "The limiting behavior of LMS," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1909-1922, Dec. 1989.

[4] V. Solo, "The error variance of LMS with time-varying weights," *IEEE Trans. Signal Processing*, vol. 40, pp. 803-813, Apr. 1992.

[5] R. R. Bitmead and B. D. O. Anderson, "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Trans. Automat. Contr.*, vol. AC-25, no. 4, pp. 788-793, Aug. 1980.

[6] R. R. Bitmead, "Convergence properties of LMS adaptive estimators with unbounded dependent inputs," *IEEE Trans. Automat. Contr.*, vol. AC-29, no. 5, pp. 477-479, May 1984.

[7] O. Macchi, "Optimization of adaptive identification for time-varying filters," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 3, pp. 283-287, Mar. 1986.

[8] P. Voltz and F. Kozin, "Almost-sure convergence of adaptive algorithms by projections," *IEEE Trans. Automat. Contr.*, vol. AC-34, no. 3, pp. 325-327, Mar. 1989.

[9] H. Kushner, *Approximation and Weak Convergence Methods for Random Processes, with Applications to Stochastic Theory*. Cambridge, MA: M.I.T. Press, 1984.

[10] H. J. Kushner and H. Huang, "Averaging methods for the asymptotic analysis of learning and adaptive systems with small adjustment rate," *SIAM J. Contr. Optimiz.*, vol. 19, pp. 635-650, 1980.

[11] K. D. Senne, "Adaptive linear discrete-time estimation," Tech. Rep. 6778-5, SU-SEL-68-090, Stanford Univ. Center Syst. Res., Stanford, CA, June 1968.

[12] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.*, pp. 546-555, Nov. 1972.

[13] J. E. Mazo, "On the independence theory of equalizer convergence," *Bell Syst. Tech. J.*, vol. 58, pp. 963-993, May-June 1979.

[14] W. A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique," *Signal Processing*, vol. 6, pp. 113-133, 1984.

[15] Ya. Z. Tsypkin, *Foundations of the Theory of Learning Systems*. New York: Academic, 1973.

[16] S. Kaczmarz, "Angenäherte Auflösung von Systemen Linearen Gleichungen," *Bull. Int. Acad. Polon. Sci. Lett. Cl. Sci. Math. Nat. A*, 1937.

[17] J. I. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, pp. 282-287, June 1967.

[18] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[19] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 793-806, Aug. 1986.

[20] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Inform. Theory*, vol. IT-34, no. 4, pp. 680-691, July 1988.

[21] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 722-735, June 1981.

[22] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 1, pp. 222-230, Feb. 1985.

[23] J. B. Foley and F. M. Boland, "A note on the convergence analysis of LMS adaptive filters with Gaussian data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, no. 7, pp. 1087-1089, July 1988.

[24] D. T. M. Slock, "On the convergence behavior of the LMS and NLMS algorithms," in *Proc. EUSIPCO 90, V European Signal Processing Conf.*, Barcelona, Spain, Sept. 18-21, 1990, pp. 197-200.

[25] T. C. Hsia, "Convergence analysis of LMS and NLMS adaptive algorithms," in *ICASSP-83*, vol. 2, Boston, MA, Apr. 1983, pp. 667-670.

[26] R. R. Bitmead and R. K. Boel, "On stochastic convergence of infinite products of random matrices and its role in adaptive estimation theory," in *Proc. 7th IFAC Symp. Syst. Identif. Param. Estim.*, York, July 1985, pp. 1223-1228.

[27] S. Jaggi and A. B. Martinez, "Upper and lower bounds of the misadjustment in the LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 164-166, Jan. 1990.

[28] N. J. Bershad, "On the optimum gain parameter in LMS adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1065-1068, July 1987.

[29] A. E. Albert and L. A. Gardner, *Stochastic Approximation and Nonlinear Regression*. Cambridge, MA: M.I.T. Press, 1967.

[30] J. M. Mendel, *Discrete Techniques of Parameter Estimation: The Equation Error Formulation*. New York: Marcel Dekker, 1973.

[31] D. R. Morgan, "Comments on convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, p. 1299, Nov. 1989.

[32] J. M. Cioffi and T. Kailath, "Fast, recursive least squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 2, pp. 304-337, Apr. 1984.

[33] N. J. Bershad, "Behavior of the ϵ-normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 636-644, May 1987.

[34] D. T. M. Slock and T. Kailath, "Fast transversal filters with data sequence weighting," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 3, pp. 346-359, Mar. 1989.

[35] A. Benveniste, "Design of adaptive algorithms for the tracking of time-varying systems," *Int. J. Adapt. Contr. Signal Processing*, vol. 1, pp. 3-29, 1987.

[36] A. Benveniste and G. Ruget, "A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 639-649, 1982.

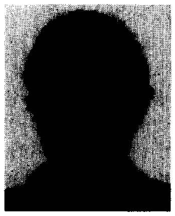
[37] E. Eleftheriou and D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 5, pp. 1097-1110, Oct. 1986.

[38] Z. Pritzker and A. Feuer, "Variable length stochastic gradient algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 997-1001, Apr. 1991.

[39] O. M. Macchi and N. J. Bershad, "Adaptive recovery of a chirped sinusoid in noise, part 1: Performance of the RLS algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 583-594, Mar. 1991.

[40] N. J. Bershad and O. H. Macchi, "Adaptive recovery of a chirped sinusoid in noise, part 2: Performance of the LMS algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 595-602, Mar. 1991.

[41] S. C. Douglas, "Exact expectation analysis of the LMS adaptive filter for correlated Gaussian input data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, vol. III, 1993,pp. 519-522.



Dirk T. M. Slock (S'85-M'88) was born in Ghent, Belgium, on October 22, 1959. He received the degree of Engineer in Electrical Engineering from the State University of Ghent in 1982, the M.S. degrees in electrical engineering and statistics and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1986, 1989, and 1989, respectively.

From 1982 to 1984, he was a Research and Teaching Assistant at the State University of Ghent on a fellowship from the National Science Foundation, Belgium. In 1984, he received a Fulbright grant and went to Stanford University, where he held appointments as a Research and Teaching Assistant. From 1989 until 1991 he was a Member of the Scientific Staff of the Philips Research Laboratory, Belgium, where he worked on the application of fast and efficient adaptive filtering algorithms to telecommunications problems. He currently is an Assistant Professor at the Eurecom Institute, Sophia Antipolis, France, where he is involved with the mobile and multimedia communications departments. His present areas of interest include fast recursive least squares algorithms and their implementation and application to electrical and acoustic echo cancellation; system identification and adaptive control; blind equalization techniques and mobile radio receiver design; wavelets and their application to image coding and adaptive filtering; and the propagation of roundoff errors in recursive algorithms.