A Mean-Square Stability Analysis of the Least Mean Fourth (LMF) Adaptive Algorithm

Pedro Inácio Hübscher, José Carlos M. Bermudez, *Senior Member, IEEE*,

and Vítor H. Nascimento, *Member, IEEE*

Abstract

This paper presents a new convergence analysis of the Least Mean Fourth (LMF) adaptive algorithm, in the mean square sense. The analysis improves previous results, in that it is valid for non-Gaussian noise distributions and explicitly shows the dependence of algorithm stability on the initial conditions of the weights. Analytical expressions are derived presenting the relationship between the step size, the initial weight error vector, and mean-square stability. The analysis assumes a white zero-mean Gaussian reference signal and an independent, identically distributed (i.i.d.) measurement noise with any even probability density function (pdf). It has been shown in [1] that the LMF algorithm is not mean-square stable for reference signals whose pdfs have infinite support. However, the probability of divergence as a function of the step size value tends to rise abruptly only when it moves past a given threshold. Our analysis provides a simple (and yet precise) estimate of the region of quick rise in the probability of divergence. Hence, the present analysis is useful for predicting algorithm instability in most practical applications.

I. Introduction

Adaptive algorithms based on higher order moments of the error signal have been shown to perform better mean square estimation than the well known Least Mean Square (LMS) algorithm in some important applications. The

This work was partly supported by CNPq under grants No. 308095/2003-0 and 303361/2004-2, and by FAPESP, under grant No. 2004/15114-2.

Pedro Hübscher is with INPE - National Institute for Space Research, 12227-010, São José dos Campos, SP, Brazil, e-mail:pedro@lit.inpe.br. Vítor Nascimento is with the Dept. of Electronic Systems Engineering, University of São Paulo, São Paulo, SP, Brazil, e-mail:vitor@lps.usp.br. José Carlos Bermudez is with the Dept. of Electrical Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil, e-mail:j.bermudez@ieee.org.

SP EDICS: 2-ADAPT, Corresponding author: José Bermudez.

Least-Mean Fourth (LMF) is one of such algorithms [2]. It seeks to minimize the mean fourth error, which is a convex (and thus unimodal) function of the adaptive weight vector [2], [3]. Over the years, LMF has been shown to have desirable properties for different applications [2], [4]–[8]. It has been shown that the LMF algorithm can outperform LMS for Gaussian, uniform and sinusoidal noise distributions [2], [9]. These results have increased the interest in a more detailed analysis of the LMF algorithm behavior, since its practical use has been limited in great part due to the lack of good analytical models to predict its performance. In [9], a statistical analysis has been presented, which led to accurate analytical models for the mean and mean-square behavior of the LMF algorithm for small step sizes. Another important aspect of the algorithm's behavior, which was not addressed in [9], is its stability.

There are several approaches to analyze the convergence of adaptive algorithms: deterministic (worst-case) [10], [11] and stochastic (in the mean, in the mean-square [11], and almost-sure [12], [13]). Deterministic approaches such as in [10] tend to be very conservative, requiring the step size to be quite small in order to guarantee stability, while almost-sure analysis may exaggerate, and conclude that an algorithm is stable when its performance is not good at all (an explanation for this can be found in [14]). Walach and Widrow [2] studied the convergence properties of the LMF algorithm in the mean-square sense. Their analysis was restricted to steady-state, and the stability limit was not expressed as a function of the initial conditions, even though the reported simulation results indicated this dependence. In [12], the ODE method was used to analyze general fixed-step adaptive algorithms, including LMF. However, no analytical expression is given for the LMF stability conditions. In [15], the authors comment on the dependence of LMF's stability on its initial conditions. An expression is provided for the maximum adaptation constant for convergence in the mean. However, the analysis in [15] does not consider the mean-square case, and assumes that both the input signal and the measurement noise are Gaussian. Reference [16] has shown that the stability of the LMF algorithm depends on the initial conditions, but such dependence was not explicitly determined. In [17] it is shown that LMF stability depends on the initial conditions, and this dependence is described analytically, using an elegant argument. However, the analysis in [17] is restricted to Gaussian noise. More recently, a different approach to stability analysis of the LMF algorithm was proposed in [1], which shows that there is always a nonzero probability of divergence in any given realization when the input signal has a probability density function (pdf) with infinite support. The probability of divergence was approximated for white Gaussian inputs.

This paper presents a new convergence analysis of the LMF algorithm in the mean-square sense[1]. The analysis considers a white zero-mean Gaussian reference signal and an independent, identically distributed (i.i.d.) zero-mean measurement noise with any even pdf. Thus, the derived results are also valid for the important applications in which the LMF algorithm is employed with non-Gaussian measurement noise [2]. Our results agree with [17] in the particular case of Gaussian noise. The dependence on the initial conditions is explicitly shown through analytical expressions.

Strictly speaking, it has been shown in [1] that the LMF algorithm is never stable in the mean-square sense for Gaussian regressors. Nevertheless, results based on standard mean-square stability analyses are useful for practical design purposes. This is because the probability of divergence as a function of the step size value tends to rise abruptly only when it moves past a given threshold. Before that, the probability of divergence tends to be sufficiently small to grant the practical applicability of the LMF algorithm[2]. Moreover, signal amplitudes are necessarily limited in practical applications, which contributes to reduce the probability of divergence for step sizes smaller than the threshold mentioned above. Our analysis provides a simple (and yet precise) estimate of the region of quick rise in the probability of divergence. One important contribution of this paper is then a useful interpretation of standard mean-square analyses of LMF in view of the results presented in [1].

Another relevant aspect of the LMF algorithm behavior is its steady-state stability. Depending on the step size and on the initial condition, the LMF probability of divergence may increase considerably with the number of iterations. Our model gives a very precise estimate of the useful step size range for the initial algorithm convergence. However, if the algorithm is initialized close to the optimum solution and one chooses a large step-size, it may have a significant probability of divergence also after initial convergence. This aspect (which we call “steady-state divergence”) is not covered in the model presented here (see comments in Sec. III-A).

The paper is organized as follows. In Section II we present a brief definition of the estimation problem considered. In Section III we derive the analytical model for the second-order moments of the weight-error vector, which

[1]Initial results on this work have been presented in [18].

[2]In practical applications it may be of interest to include a re-initialization scheme in case, for instance, the error signal tends to increase without bound.

determine the stability of the algorithm. In Section IV we perform the stability analysis, where the conditions on the step size and on the algorithm initialization are derived. In Section V we present simulation examples to illustrate the application of the theoretical results. Throughout the paper we use lowercase bold letters to represent column vectors, capital bold letters for matrices and regular (non-bold) lowercase letters for scalars. Capital regular letters are used for constants.

II. PROBLEM DEFINITION

Fig. 1 shows a block diagram of the problem studied here. The adaptive filter attempts to estimate a desired signal $d(n)$, which is linearly related to the input signal $x(n)$ by the stationary model

$$d(n) = \boldsymbol{w}^{\mathrm{o}T}\boldsymbol{x}(n) + z(n) \tag{1}$$

where $\boldsymbol{w}^{\mathrm{o}} = [w_0^0, w_1^0, ..., w_{N-1}^0]^T$ is the vector of the model parameters, $x(n)$ is assumed stationary, white, zero-mean and Gaussian with variance σ_x^2. $\boldsymbol{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ is the observed data vector with correlation matrix $\boldsymbol{R} = \mathrm{E}[\boldsymbol{x}(n)\boldsymbol{x}^T(n)] = \sigma_x^2\boldsymbol{I}$, with $\boldsymbol{I}$ being the $N \times N$ identity matrix. The sequence $z(n)$ is a zero-mean i.i.d. random sequence, with variance σ_z^2 and statistically independent of any other signal. Moreover, it is assumed that $z(n)$ can have any distribution with an even pdf. The sequence $z(n)$ in (1) accounts for measurement noise and modeling errors. Vector $\boldsymbol{w}(n) = [w_0(n), w_2(n), ..., w_{N-1}(n)]^T$ is the adaptive weight vector, and $e(n)$ is the error signal.

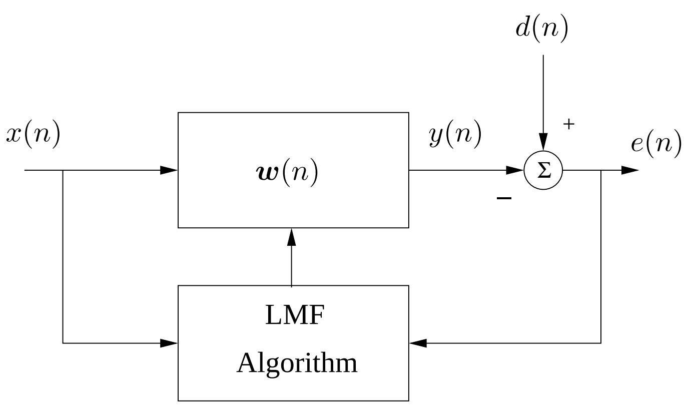

Fig. 1. Adaptive system under study.

The LMF algorithm weight update equation is given by [2]

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu e^3(n)\boldsymbol{x}(n), \tag{2}$$

where μ is the step size and

$$e(n) = d(n) - y(n) = \boldsymbol{w}^{\mathrm{o}T}\boldsymbol{x}(n) + z(n) - \boldsymbol{w}^T(n)\boldsymbol{x}(n). \tag{3}$$

Defining the weight error vector $\boldsymbol{v}(n) = \boldsymbol{w}(n) - \boldsymbol{w}^{\mathrm{o}} = [v_0(n), \ldots, v_{N-1}(n)]^T$ about the optimal solution, (2) can be written as

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \mu e^3(n)\boldsymbol{x}(n) \tag{4}$$

with

$$e(n) = z(n) - \boldsymbol{v}^T(n)\boldsymbol{x}(n). \tag{5}$$

The convergence properties of this algorithm are studied in the following.

III. Second Moment Analysis

Though the conditions for convergence in the mean can provide some insight on algorithm behavior, second moment stability is far more important in determining conditions for algorithm convergence [19], [20]. Thus, we restrict the analysis to the study of the conditions for mean-square convergence. As shown in [1], under the assumption of Gaussian regressors, the LMF algorithm is not mean-square stable no matter how small the step size. However, as the step size decreases (if the initial condition is close enough to the optimum weight vector $\boldsymbol{w}^{\mathrm{o}}$), the probability of good behavior (convergence) of a single realization of the algorithm increases, tending to 1 as the step size decreases to zero. The analysis provided here (and other analyses in the literature) therefore give a step size for which the probability of divergence is still small, as our simulations show.

For white inputs and neglecting the statistical dependence between $\boldsymbol{x}(n)\boldsymbol{x}^T(n)$ and $\boldsymbol{v}(n)$, straightforward calculation using (5) shows that the second order moments of the weights are related to the mean-square error (MSE) through [20]

$$\mathrm{E}[e^2(n)] = \sigma_z^2 + \sigma_x^2\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)], \tag{6}$$

Hence, the MSE convergence can be studied through the convergence properties of $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$. The convergence analysis then reduces to the study of a recursive scalar equation.

A recursive expression for the behavior of $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ could be easily obtained by taking the trace of the recursion derived in [9, Eq. (22)] for the weight error correlation matrix $\boldsymbol{K}(n) = \mathrm{E}[\boldsymbol{v}(n)\boldsymbol{v}^T(n)]$ of the LMF algorithm. However, terms neglected in [9] which were not significant for the analysis done there become important for a stability analysis, since large values of the step size μ must be considered in this case[3]. The model derived for the MSE in [9] indicates that the MSE tends to $-\infty$ when the algorithm becomes unstable, which is obviously incorrect. The model in [9] is valid only in the stability region. Therefore, a new recursive expression for $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ must be determined for the convergence analysis by starting again from the LMF weight-error update equation (4).

Pre-multiplying (4) by its transpose, using (5), taking the expected value and using the statistical properties of $z(n)$,[4] leads to

$$\begin{aligned}\mathrm{E}[\boldsymbol{v}^T(n+1)\boldsymbol{v}(n+1)] = {}& \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] - 2\mu\overbrace{\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4\}}^{(1)} - 6\mu\,\mathrm{E}[z^2(n)]\overbrace{\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\}}^{(2)} \\ & + \mu^2\overbrace{\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^6\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}}^{(3)} + 15\mu^2\,\mathrm{E}[z^2(n)]\overbrace{\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}}^{(4)} \\ & + 15\mu^2\,\mathrm{E}[z^4(n)]\overbrace{\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}}^{(5)} + \mu^2\,\mathrm{E}[z^6(n)]\overbrace{\mathrm{E}[\boldsymbol{x}^T(n)\boldsymbol{x}(n)]}^{(6)}.\end{aligned} \tag{7}$$

The following analysis assumes that the effects of the statistical dependence of $\boldsymbol{x}(n)\boldsymbol{x}^T(n)$ and $\boldsymbol{v}(n)$ can be neglected. This assumption is weaker than assuming $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$ statistically independent [21]. The expected values in (7) are then calculated as follows:

[3] The recursion for $\boldsymbol{K}(n)$ was derived in [9] by neglecting the higher-order terms $\mathrm{E}[(\boldsymbol{x}^T(n)\boldsymbol{v}(n))^{2k}\boldsymbol{x}(n)\boldsymbol{x}^T(n)]$ for $k > 1$, and considering a small step size μ.

[4] The two important properties of $z(n)$ used in evaluating (7) were its independence of any other signal and the even pdf, which leads to zero odd-order moments.

A. Expected Value 1: $\mathrm{E}[(\boldsymbol{x}^T(n)\boldsymbol{v}(n))^4]$

For $\boldsymbol{x}(n)$ zero-mean, Gaussian and independent of $\boldsymbol{v}(n)$, $\boldsymbol{x}^T(n)\boldsymbol{v}(n)$ is also zero-mean Gaussian when conditioned on $\boldsymbol{v}(n)$. Thus, we can write [22]

$$\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^{2k}|\boldsymbol{v}(n)\} = \mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2|\boldsymbol{v}(n)\}^k \prod_{m=1}^{k}(2m-1). \tag{8}$$

and then

$$\begin{aligned}\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4|\boldsymbol{v}(n)\} &= 3\,\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2|\boldsymbol{v}(n)\}^2 \\ &= 3\big\{\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{v}(n)|\boldsymbol{v}(n)]\big\}^2 \\ &\approx 3\big\{\boldsymbol{v}^T(n)\boldsymbol{R}\boldsymbol{v}(n)\big\}^2 = 3\big\{\sigma_x^2\boldsymbol{v}^T(n)\boldsymbol{v}(n)\big\}^2 \\ &= 3\sigma_x^4\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n)\end{aligned} \tag{9}$$

Averaging (9) over $\boldsymbol{v}(n)$ requires extra approximations, since the pdf of $\boldsymbol{v}(n)$ is unknown. We use the following approximation:

$$\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \approx \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{10}$$

Approximation (10) assumes that the variance of $\boldsymbol{v}^T(n)\boldsymbol{v}(n)$ is much smaller than its mean value[5]. This assumption is valid in the beginning of the adaptation process, since its is reasonable to expect that the adaptive weights will be initialized far (relative to the steady-state standard deviation $\sqrt{\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]}$) from the optimal weights. Then, the mean value of $v_i(n)$, $i = 0, \ldots, N-1$, can be assumed much larger than its fluctuations in the beginning of adaptation.[6] When instability does occur, its onset is usually during the initial adaptation phase and due to the unbounded increase of the second order moments, which can be detected using (10). Extensive simulation results have shown that this approximation leads to good accuracy in determining the stability conditions.

Reference [23] approximates (10) by assuming that $\boldsymbol{v}(n)$ is Gaussian-distributed. However, the analysis in [1] shows that this assumption is not satisfied. [23] does not provide estimates of the range of step sizes for stable

[5]Note that $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n)] = E^2[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] + \sigma^2_{\boldsymbol{v}^T(n)\boldsymbol{v}(n)}$, where $\sigma^2_{\boldsymbol{v}^T(n)\boldsymbol{v}(n)}$ is the variance of $\boldsymbol{v}^T(n)\boldsymbol{v}(n)$.

[6]For the purpose of this analysis, beginning of adaptation is the phase during which $\mathrm{E}[\boldsymbol{v}(n)]$ is much larger than its fluctuations. The actual duration of this phase depends on the weight initialization, the step size, the adaptive filter length, the noise power and the input signal power.

behavior, mainly because of the complexity of the model.

Using (10) and the property that $\mathrm{E}\{XY\} = E_Y\{E_X\{XY|Y\}\}$ for any random variables X and Y [22], (9) becomes

$$\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4\} \approx 3\sigma_x^4\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \approx 3\sigma_x^4\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{11}$$

B. Expected Value 2: $\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\}$

Conditioning on $\boldsymbol{v}(n)$,

$$\begin{aligned}\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2|\boldsymbol{v}(n)\} &= \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{v}(n)|\boldsymbol{v}(n)] \\ &= \boldsymbol{v}^T(n)\,\mathrm{E}[\boldsymbol{x}(n)\boldsymbol{x}^T(n)|\boldsymbol{v}(n)]\boldsymbol{v}(n) \\ &\approx \boldsymbol{v}^T(n)\boldsymbol{R}\boldsymbol{v}(n) = \sigma_x^2\boldsymbol{v}^T(n)\boldsymbol{v}(n) \end{aligned} \tag{12}$$

Averaging (12) over $\boldsymbol{v}(n)$ gives

$$\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\} \approx \sigma_x^2\ \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{13}$$

Next, we evaluate the expected values that are multiplied by μ^2 in (7). They are evaluated using the same methodology presented in [9] and [24], and also using approximations similar to (10).

C. Expected Value 3: $\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^6\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}$

Using the properties of higher moments of zero-mean Gaussian variables [22], the expected value conditioned on $\boldsymbol{v}(n)$ can be written as:

$$\begin{aligned} E\Big\{\big[\boldsymbol{x}^T(n)\boldsymbol{v}(n)\big]^6\boldsymbol{x}^T(n)\boldsymbol{x}(n)|\boldsymbol{v}(n)\Big\} &= \mathrm{tr}\Big\{E\big[[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^6\boldsymbol{x}(n)\boldsymbol{x}^T(n)|\boldsymbol{v}(n)\big]\Big\} \\ &\approx (15N+90)\sigma_x^8\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{v}(n) \end{aligned} \tag{14}$$

To average (14) over $\boldsymbol{v}(n)$, we again neglect the higher order moments of $\boldsymbol{v}^T(n)\boldsymbol{v}(n)$. Thus, $\boldsymbol{v}^T(n)\boldsymbol{v}(n) \approx \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ and

$$\mathrm{E}\{[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]^3\} \approx \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{15}$$

Thus,

$$\begin{aligned}\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^6\boldsymbol{x}^T(n)\boldsymbol{x}(n)\} &= \mathrm{tr}\Big\{\mathrm{E}[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^6\boldsymbol{x}(n)\boldsymbol{x}^T(n)\Big\} \\ &\approx (15N+90)\sigma_x^8 E^3[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]\end{aligned} \tag{16}$$

D. Expected Value 4: $\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}$

Using the same methodology as above,

$$\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^4\boldsymbol{x}^T(n)\boldsymbol{x}(n)\} \approx (3N+12)\sigma_x^6 E^2[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{17}$$

E. Expected Value 5: $\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\boldsymbol{x}^T(n)\boldsymbol{x}(n)\}$

Using again the same technique,

$$\mathrm{E}\{[\boldsymbol{x}^T(n)\boldsymbol{v}(n)]^2\boldsymbol{x}^T(n)\boldsymbol{x}(n)\} \approx (N+2)\sigma_x^4\,\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)] \tag{18}$$

F. Expected Value 6: $\mathrm{E}[\boldsymbol{x}^T(n)\boldsymbol{x}(n)]$

$$\mathrm{E}[\boldsymbol{x}^T(n)\boldsymbol{x}(n)] = N\sigma_x^2 \tag{19}$$

Using the expected values 1-6 in (7), we obtain the following expression for $\mathrm{E}[\boldsymbol{v}^T(n+1)\boldsymbol{v}(n+1)]$:

$$y(n+1) = (1-a)y(n) - by^2(n) + cy^3(n) + d \tag{20}$$

where $y(n) = \mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ and

$$a = A_1\mu - A_2\mu^2 \qquad b = B_1\mu - B_2\mu^2 \qquad c = C\mu^2 \qquad d = D\mu^2 \tag{21}$$

with

$$\begin{aligned} &A_1 = 6\sigma_z^2\sigma_x^2 && A_2 = 15\,\mathrm{E}[z^4(n)]\sigma_x^4(N+2) && B_1 = 6\sigma_x^4 \\ &B_2 = 15\sigma_z^2\sigma_x^6(3N+12) && C = \sigma_x^8(15N+90) && D = \mathrm{E}[z^6(n)]\sigma_x^2 N. \end{aligned} \tag{22}$$

In the following, we study the convergence properties of (20).

At this point, a clarification is necessary. The model (20) has been derived to predict algorithm instabilities caused by unbounded growth of second order moments. These are usually the instabilities of greatest interest, as they occur at the initial stages of adaptation and for small or moderate step size values. However, we have also verified experimentally that using large step sizes may lead to a significant probability of divergence after LMF has initially converged to a small region about the optimum weights, which corresponds to the steady-state solution of (20). This is specially true for low-order filters. This so-called steady-state divergence is a function of higher moments of the weight vector and cannot be predicted using the approximation in (10). As we show in Section IV, large step-sizes imply that the initial weights are very close to their optimum values (not a practical situation), which implies that approximation (10) does not hold. If the step-size is chosen reasonably smaller then the maximum for a given value of $y(0)$, the phenomenon of steady-state divergence becomes less likely. Thus, the model (20) should be useful for most practical designs[7]. It should also be added that the possibility of steady-state divergence cannot be detected by the model in [23]. Experimental results on the steady-state instability will be presented in Section V.

IV. STABILITY ANALYSIS

Expression (20) is a nonlinear difference equation that approximates the dynamics of the LMF algorithm. Its convergence depends in general on the initial condition $y(0) = \mathrm{E}[\boldsymbol{v}^T(0)\boldsymbol{v}(0)]$, the squared Euclidean norm of the initial weight error vector. To study the stability conditions for (20), we need to find its equilibrium points.

Making $y(n+1) = y(n) = y_\infty$ in (20), we obtain

$$y_\infty = (1-a)y_\infty - by_\infty^2 + cy_\infty^3 + d. \tag{23}$$

Since $c > 0$, we can rewrite (23) as

$$y_\infty^3 - \frac{b}{c}y_\infty^2 - \frac{a}{c}y_\infty + \frac{d}{c} = 0. \tag{24}$$

[7]For practical situations, when the initial condition is reasonably far from the optimum weights and the step size is relatively smaller than the maximum value predicted by our model ($\mu_{\max}$), we never did observe "steady-state divergence". If a filter is designed so that the initial probability of divergence is reasonably small (say, less than 0.1%), the probability of "steady-state divergence" must be very small.

Equation (24) has three roots, which represent the equilibrium points. These roots can be expressed in analytical form as follows [25]:

$$
\begin{aligned}
y_{1\infty} &= (s_1 + s_2) + \frac{b}{3c}, \\
y_{2\infty} &= -\frac{1}{2}(s_1 + s_2) + \frac{b}{3c} + \frac{j\sqrt{3}}{2}(s_1 - s_2), \\
y_{3\infty} &= -\frac{1}{2}(s_1 + s_2) + \frac{b}{3c} - \frac{j\sqrt{3}}{2}(s_1 - s_2),
\end{aligned}
\tag{25}
$$

where

$$
\begin{aligned}
s_1 &= \left(r + \sqrt{q^3 + r^2}\right)^{\frac{1}{3}}, \\
s_2 &= \left(r - \sqrt{q^3 + r^2}\right)^{\frac{1}{3}},
\end{aligned}
\tag{26}
$$

with

$$
\begin{aligned}
q &= -\frac{a}{3c} - \frac{b^2}{9c^2}, \\
r &= \frac{1}{6}\left(\frac{ab}{c^2} - \frac{3d}{c}\right) + \frac{b^3}{27c^3}.
\end{aligned}
\tag{27}
$$

Depending on the values of q and r, three cases can occur:

A. *Case 1: $q^3 + r^2 < 0$ (three real roots)*

In this case, (24) has either three negative real roots or one negative and two distinct positive real roots[8]. The first option is of no interest, since $y(n)$ is a squared magnitude and thus must be non-negative. The second option has two non-negative roots which are of interest to our study. Fig. 2 illustrates the case of one negative and two positive roots, denoted y_{neg}, y_{c} and $y(0)_{\max}$ respectively. Root $y(0)_{\max}$ corresponds to a stability limit[9]. Thus $y(0)_{\max}$ is the maximum value of $y(0)$ that guarantees stability of (20) for a specific value of μ. The smaller the value of μ, the larger the value of $y(0)_{\max}$. As $\mu \to 0$, $y(0)_{\max} \to \infty$. Root y_{c} corresponds to the stable equilibrium point, and thus is the steady-state solution $y(\infty)$ for this case.

To study the behavior of the curve $y(n+1)$ in Fig. 2 in the range $0 < y(n) < y(0)_{\max}$ we determine its slope in the region. Given that $d > 0$ and that we assume the two roots of (25) to be real and positive, the derivative

[8]Note that one root is always negative in this case because $d > 0$ in (24) implies that $y(n+1) > 0$ for $y(n) = 0$, and $c > 0$ implies that $y(n+1) \to -\infty$ when $y(n) \to -\infty$.

[9]Note from Fig. 2 that $y(0) > y(0)_{\max}$ implies $y(n+1) > y(n)$ for all $n > 0$, and thus instability.

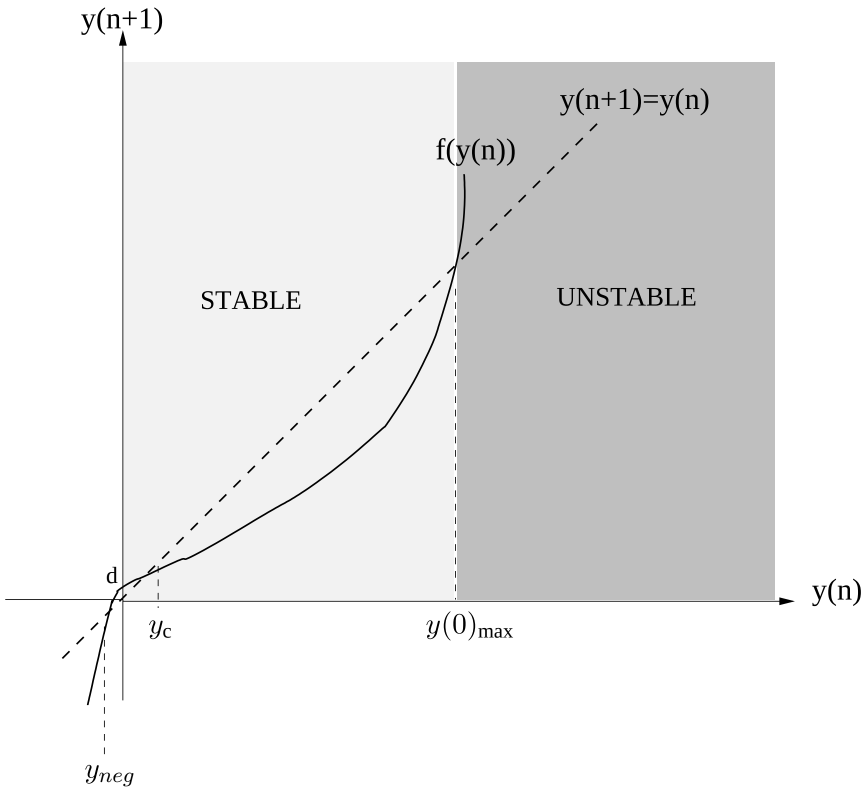

Fig. 2. Equilibrium points: case 1 ($y_\mathrm{c} << y(0)_\mathrm{max}$).

of $y(n+1)$ with respect to $y(n)$ at $y(n) = y_\mathrm{c}$ is less than one. Therefore, the condition for y_c to be a stable equilibrium point is that $dy(n+1)/dy(n) > -1$ at $y(n) = y_\mathrm{c}$. Differentiating (20) with respect to $y(n)$ yields

$$\frac{dy(n+1)}{dy(n)} = 1 - a - 2by(n) + 3cy^2(n) \tag{28}$$

Differentiating again with respect to $y(n)$ and equating the result to zero results in the condition for a stationary point of $dy(n+1)/dy(n)$:

$$\frac{d^2y(n+1)}{dy(n)^2} = 6cy(n) - 2b = 0 \tag{29}$$

Values of $y(n)$ satisfying (29) may correspond to a maximum or to a minimum of $dy(n+1)/dy(n)$. Another differentiation with respect to $y(n)$ yields

$$\frac{d^3y(n+1)}{dy(n)^3} = 6c \tag{30}$$

Since $c > 0$, $y(n)$ satisfying (29) corresponds to a minimum of $dy(n+1)/dy(n)$. Using $y(n) = b/3c$ obtained from (29) in (28) yields the value of the smallest derivative:

$$\min\left\{\frac{dy(n+1)}{dy(n)}\right\} = \frac{3c(1-a)-b^2}{3c}. \tag{31}$$

Thus, y_{c} will be a stable equilibrium point if (31) is greater than -1. In particular, the convergence will be monotonic if (31) is greater or equal to zero (see below). A necessary and sufficient condition for (31) to be greater or equal than zero is

$$b^2 \leq 3c(1-a). \tag{32}$$

We determine in Appendix I the conditions for (32) to be satisfied for large values of N. It is shown that extra restrictions may apply to the possible values of μ, depending on the relationship between $\mathrm{E}[z^4(n)]$ and σ_z^4.

If (32) is satisfied, $y(n+1)$ is monotonically increasing with $y(n)$ in $[0, y(0)_{\mathrm{max}}]$, and $y(n)$ converges monotonically to y_{c} when $y(n)$ is initialized in the range $0 < y(0) < y(0)_{\mathrm{max}}$, as shown in Fig. 3. Thus, y_{c} will be a stable equilibrium point of (20) whenever the conditions derived in Appendix I are satisfied.

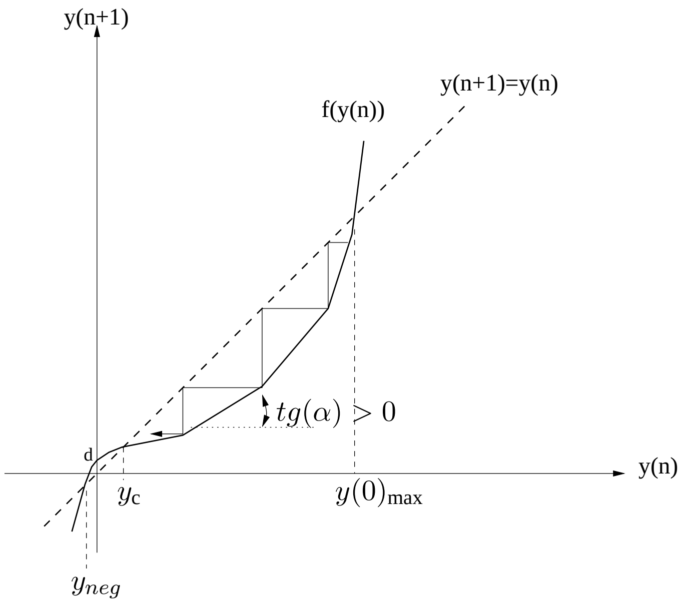

Fig. 3. Equilibrium points: case 1 - derivatives.

If (32) is not satisfied, there is still a possibility that y_{c} be a stable equilibrium point, though the convergence to it will not be monotonic. It is easy to verify that y_{c} will be a stable equilibrium point with non-monotonic convergence if

$$-1 < \frac{dy(n+1)}{dy(n)}\Big|_{y(n)=y_\mathrm{c}} < 0. \tag{33}$$

In this case $y(n)$ will show decreasing oscillations about y_c. This situation is illustrated in Fig. 4. Applying (33) to (31) we obtain a condition for convergence in this situation:

$$3c(1-a) < b^2 < 3c(2-a). \tag{34}$$

The conditions for (34) to be satisfied are determined in Appendix II. These conditions are similar to those derived for (32).

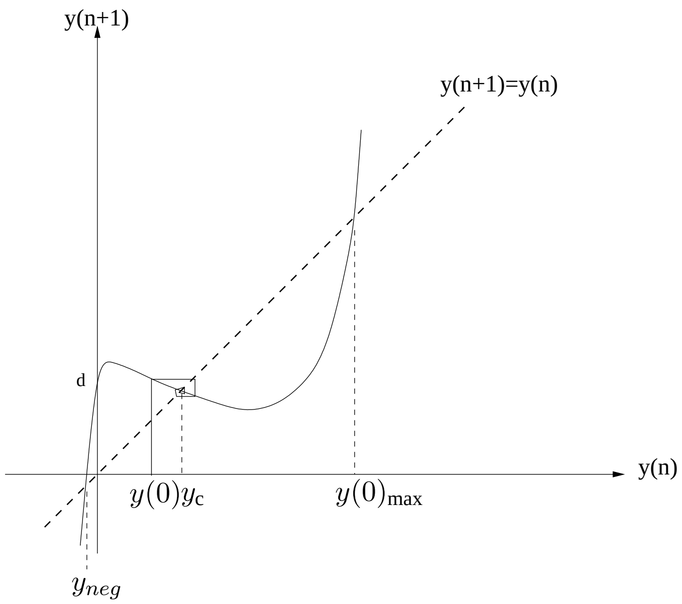

Fig. 4. Equilibrium points: case 1 - non-monotonic convergence.

B. Case 2: $q^3 + r^2 = 0$ (only real roots, and two of them are equal and nonzero)

In this case, (24) has two real, positive, and equal roots (y_c and $y(0)_\mathrm{max}$ coincide) and one negative root (y_{neg}), which is again not of interest for the convergence analysis. The curve $y(n+1)$ is tangent to the line $y(n+1) = y(n)$ at the saddle point $y_\mathrm{c} = y(0)_\mathrm{max}$ as shown in Fig. 5.

This case clearly determines the stability limit for the step size μ, as it defines the value of μ for which there is no gap between y_c and $y(0)_\mathrm{max}$. To determine the step size stability limit, we write $q^3 + r^2$ as a function of a, b, c

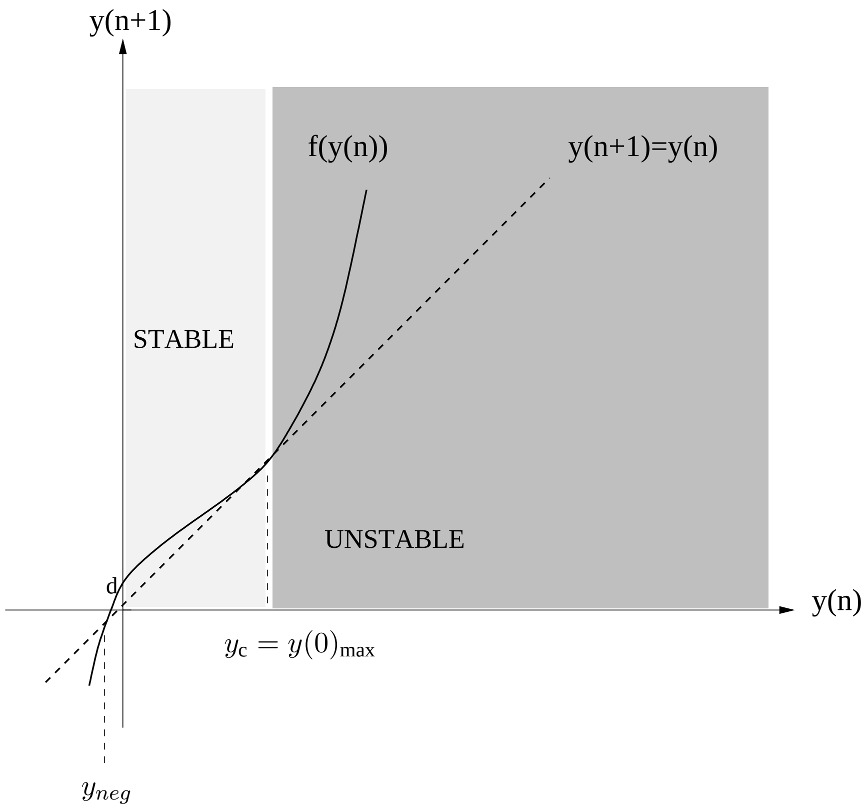

Fig. 5. Equilibrium points: case 2 ($y_c = y(0)_{\max}$).

and d, yielding

$$\begin{aligned} q^3 + r^2 &= \left\{-\frac{a}{3c} - \frac{b^2}{9c^2}\right\}^3 + \left\{\frac{1}{6}\left(\frac{ab}{c^2} - \frac{3d}{c}\right) + \frac{b^3}{27c^3}\right\}^2 \\ &= -\frac{1}{729c^6}\Big(3ac + b^2\Big)^3 \\ &\quad + \frac{1}{729c^6}\left(\frac{9}{2}abc - \frac{27}{2}dc^2 + b^3\right)^2. \end{aligned} \tag{35}$$

Since (35) is equal to zero for Case 2, we conclude that

$$4\Big(3ac + b^2\Big)^3 = \Big(9abc - 27dc^2 + 2b^3\Big)^2. \tag{36}$$

Writing (36) as a polynomial in μ, and substituting the variables a, b, c and d as functions of A_1, A_2, B_1, B_2, C, D according to (21) and (22) leads to

$$P_4\mu^4 + P_3\mu^3 + P_2\mu^2 + P_1\mu + P_0 = 0, \tag{37}$$

where:

$P_4 = -4A_2^3C + A_2^2B_2^2 + 18A_2B_2CD - 4B_2^3D - 27C^2D^2;$

$P_3 = 12A_1A_2^2C - 2(A_1A_2B_2^2 + A_2^2B_1B_2) - 18(A_1B_2 + A_2B_1)CD + 12B_1B_2^2D;$

$P_2 = -12A_1^2A_2C + A_1^2B_2^2 + A_2^2B_1^2 + 4A_1A_2B_1B_2 + 18A_1B_1CD - 12B_1^2B_2D;$

$P_1 = 4A_1^3C - 2(A_1A_2B_1^2 + A_1^2B_1B_2) + 4B_1^3D;$

$P_0 = A_1^2B_1^2.$

The smallest positive real root μ_{o} of (37) corresponds to the stability limit. If the weights were initialized equal to the optimum weight vector $\boldsymbol{w}^{\text{o}}$ or with values such that $\boldsymbol{v}^T(1)\boldsymbol{v}(1) \leq y_{\text{c}}$, the LMF recursion would start apparently converging, but any noise would drive the recursion to instability. This is the situation in which what we called "steady-state divergence" occurs: in practice, noise will be able to drive the filter to instability when y_{c} is close (not necessarily equal) to $y(0)_{\text{max}}$.

C. *Case 3: $q^3 + r^2 > 0$ (two complex roots)*

In this case (24) may have three real negative roots (a situation of no interest) or one negative real and two complex roots. There is no solution $y(n+1) = y(n)$ for $y(n) > 0$. Therefore, there is no value of $y(0) \geq 0$ that guarantees stability of (20). The complex roots case is illustrated in Fig. 6, and occurs for values of μ larger than the limit value μ_{o}.

The results derived in this section allow the explicit determination of the stability conditions for the LMF algorithm when applied to the system in Fig. 1. Given the system parameters, the maximum value of μ ($\mu = \mu_{\text{o}}$) can be determined from (37). Then, for any $\mu < \mu_{\text{o}}$, $y(0)_{\text{max}}$ can be determined from the solutions of (24). The value of $\sqrt{y(0)_{\text{max}}}$ is the maximum initial distance from the initial weight vector $\boldsymbol{w}(0)$ to the optimum weight vector that guarantees algorithm stability for a given value of μ. Thus, the use of this information directly for design purposes requires a reasonably good estimate of $\boldsymbol{w}^{\text{o}}$, the response of the system to be identified. This is a consequence of the fact that the stability limit is a function of the weight vector initialization. Such property is common to adaptive algorithms employing higher (greater than 2) order moments of the estimation error. Nevertheless, the analysis results provide an analytical model that can be used to study the robustness of the algorithm in solving practical problems, and to design the algorithm for an expected margin of error in the initial estimate of $\boldsymbol{w}^{\text{o}}$.

The theoretical model can also be used to estimate the range of step-sizes that guarantee stability of (20) for a

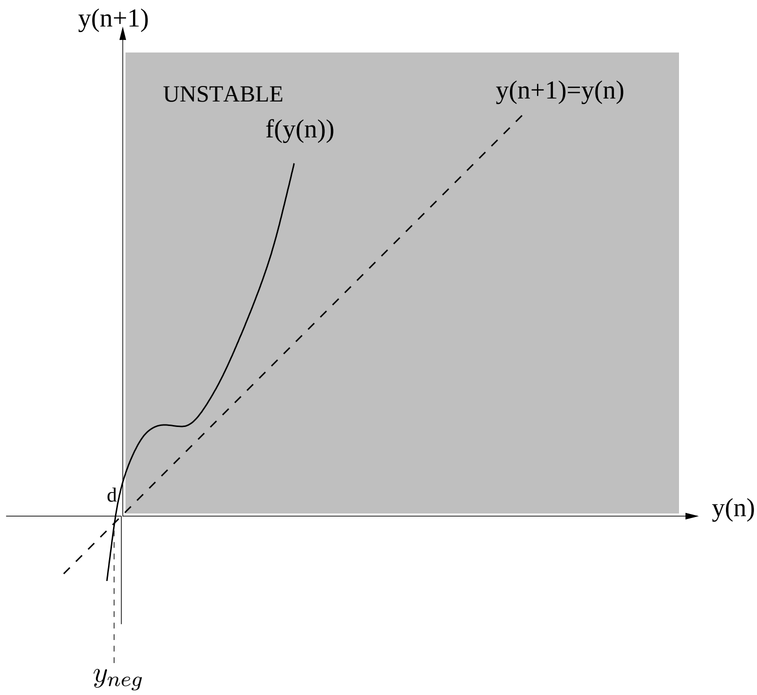

Fig. 6. Equilibrium points: case 3 (y_c and $y(0)_{\text{max}}$ complex).

given initialization $y(0)$. To this end, (24) can be used iteratively to determine the maximum step size associated with a desired initial condition $y_{\text{d}}(0)$. First, it is important to notice that there will be a different value of $y(0)_{\text{max}}$ for each value of μ. Define μ_k as the value of μ for which the maximum possible value of $y(0)$ is $y(0)_{\text{max}_k}$ (determined from (24) with $\mu = \mu_k$). Then, for $\mu = \mu_{\text{o}}$ evaluated from (37), $y(0)_{\text{max}_0} = \min\{y(0)_{\text{max}}\}$. Given a desired initial condition $y_{\text{d}}(0)$, we start by comparing $y_{\text{d}}(0)$ with $y(0)_{\text{max}_0}$. If $y_{\text{d}}(0) > y(0)_{\text{max}_0}$, μ must be reduced. We then make $\mu_1 = \mu_{\text{o}} - \Delta_\mu$. Using now μ_1 in (24) leads to $y(0)_{\text{max}_1}$, which must be compared with $y_{\text{d}}(0)$. If $y_{\text{d}}(0) > y(0)_{\text{max}_1}$, $\mu_2 = \mu_1 - \Delta_\mu$ is evaluated as a new candidate for μ_{max}. The cycle continues until $y_{\text{d}}(0) \leq y(0)_{\text{max}_k}$. Then, $\mu_{\text{max}} \approx \mu_k$.

V. Simulation Examples

This section presents simulation examples to verify the accuracy of the theoretical analysis. Initially, we illustrate the accuracy of (20) through an example with $\sigma_x^2 = 1$, $z(n)$ uniform with zero-mean and $\sigma_z^2 = 0.1$, $N = 29$, $\mu = 0.004$ for $\mu_{\text{max}} = 0.0468$, $y(0) = 1$. The plant's impulse response $\boldsymbol{w}^{\text{o}}$ was a delayed raised-cosine function given by [26]

$$\boldsymbol{w}^{\text{o}}(k) = \left\{ \frac{\sin\left[6\pi f_{\text{o}}(n - n_{\text{o}})/N\right]}{6\pi f_{\text{o}}(n - n_{\text{o}})/N} \right\} \left\{ \frac{\cos\left[6\pi r f_{\text{o}}(n - n_{\text{o}})/N\right]}{1 - 12 r f_{\text{o}}(n - n_{\text{o}})/N} \right\}, \tag{38}$$

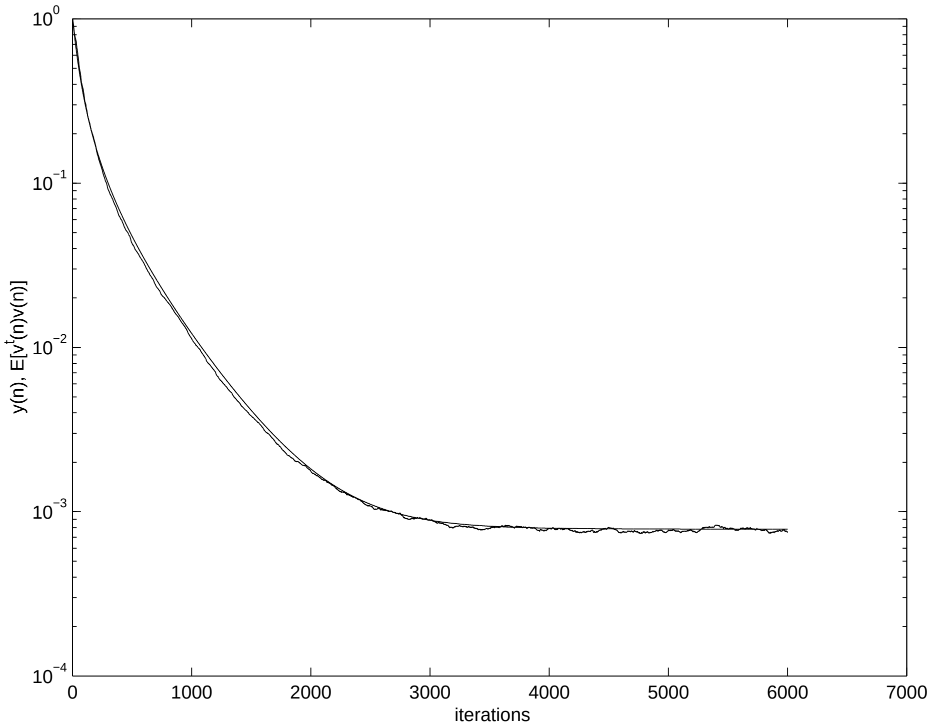

Fig. 7. $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ for LMF with white input and zero-mean uniform noise. $\sigma_x^2 = 1$, $\sigma_z^2 = 0.1$, $y(0) = 1$, $N = 29$ taps and $\mu \simeq \mu_{\max}/10$. Ragged curve: Monte Carlo simulation averaged over 50 runs. Smooth curve: Theoretical model (20).

where N is the number of coefficients, r is roll-off factor ($0 \leq r \leq 1$), n_o is the right-shift delay relative to the even function case and f_o is the expansion factor. For this example, $r = 0.1$, $n_\mathrm{o} = (N-1)/4 = 14$ and $f_\mathrm{o} = 0.58$. The mean-square deviation $\mathrm{E}[\boldsymbol{v}^T(n)\boldsymbol{v}(n)]$ is shown in Fig. 7. The simulation (ragged) curve was obtained from Monte Carlo simulation averaged over 50 realizations for which no steady-state divergence was observed. The theoretical (smooth) curve was obtained from (20).

To test the theoretical stability limit we estimate the algorithm's probability of divergence P_d obtained from several experiments (see also [1]). To estimate P_d, each experiment is repeated L times, starting from the same initial condition $y(0) = \|\boldsymbol{v}(0)\|^2$. A sample function is labeled as "diverging" if $\|\boldsymbol{v}(N_\mathrm{it})\| \geq 10^4$ after N_it iterations[10]. We then compute the observed probability of divergence as $P_\mathrm{d,o} = (\text{Number of curves diverging})/L$ and draw the curves for $P_\mathrm{d,o}$ versus μ. The instability onsets obtained from these curves can then be compared with the theoretical stability limit obtained using the procedure described in the last paragraph of the previous section.

Fig. 8 shows a typical example, comparing the probabilities of divergence observed when the noise is uniform

[10]The results reported in these simulations are very insensitive to divergence threshold used, as the divergence is "explosive" when it occurs. Thresholds varying from 10^4 to 10^{10} have been used with basically the same results.

with $\sigma_z^2 = 0.01$, $N = 100$, $\sigma_x^2 = 1$, $y(0) = 1$, $L = 10^3$, $N_{\text{it}} = 10^4$. The vector $\boldsymbol{w}^{\text{o}}$ was obtained from (38) with $r = 0$, $n_{\text{o}} = 2$ and $f_{\text{o}} = 2$. The value of $\mu_{\text{max}}(y(0))$ computed from (24) corresponds to the vertical line[11].

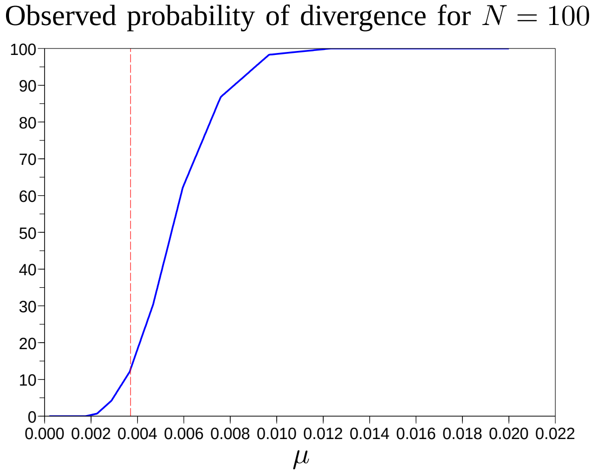

Fig. 8. Probability of divergence, uniform noise, $N = 100$, $\sigma_x^2 = 1$, $y(0) = 1$, $\sigma_z^2 = 0.01$, $L = 10^3$, $N_{\text{it}} = 10^4$. The vertical line shows the value of μ_{max}.

Table I lists the values of μ_{max} obtained for different noise variances σ_z^2, signal variances σ_x^2, filter lengths N and initial conditions $y(0)$, for both uniform and Gaussian noise distributions. The table also shows the step sizes for which the observed probability of divergence was 1% ($\mu_{1\%}$) and 99% ($\mu_{99\%}$). In order to keep the simulation time manageable, $\mu_{1\%}$ and $\mu_{99\%}$ were obtained by interpolation from a grid of actually measured values. Note that, except for three situations (cases 2, 6, and 28, out of the 33 displayed)[12], μ_{max} is always between $\mu_{1\%}$ and $\mu_{99\%}$.

We should also note that as $y(0)$ is decreased below $y(0)_{\max}$, the corresponding $\mu_{\max}$ initially increases. However, $\mu_{\max}$ cannot become larger than μ_{o}, so there is a limit y_{l} below which a reduction in $y(0)$ will not translate to an increase in $\mu_{\max}$.

Figures 9 and 10 show the probability of divergence as a function of the initialization $y(0)$, for six different values of μ. The vertical lines show the values of $y(0)_{\text{max}}$ computed from (24) for each value of μ.

In order to illustrate the LMF steady-state instability described at the end of Section III (but not predictable by the present model), Table II shows observed steady-state probabilities of divergence as a function of the number of iterations for three different cases. In the first four examples, the number of iterations does not affect significantly the observed probability of divergence. In the fifth case, this probability clearly increases with the number of

[11] The number of repetitions $L = 10^3$ used to estimate the probabilities was somewhat small to reduce simulation time. Since our goal with these simulations is simply to show the region of fast increase in $P_{\text{d,o}}$, there is no need to estimate these probabilities with great accuracy.

[12] This may be due the approximations used to estimate the values of $\mu_{1\%}$ and $\mu_{99\%}$.

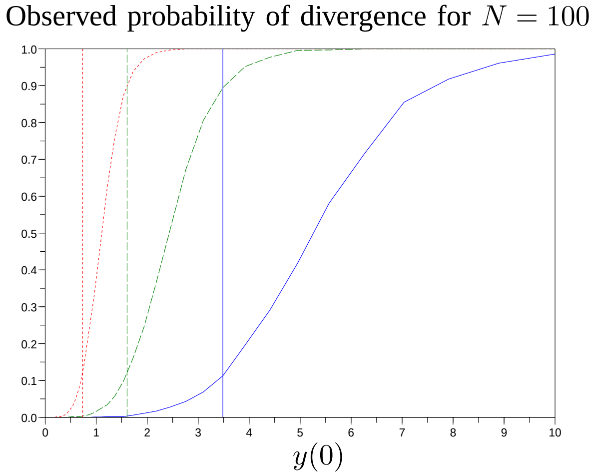

Fig. 9. Probability of divergence, uniform noise, $N = 100$, $\sigma_x^2 = 1$, $\mu = 0.0010772$ (solid), $\mu = 0.0023208$ (long dash), and $\mu = 0.005$ (short dash), $\sigma_z^2 = 0.01$, $L = 10^3$, $N_{\rm it} = 10^4$. The vertical lines show the values of $y(0)_{\rm max}$ computed for each value of μ.

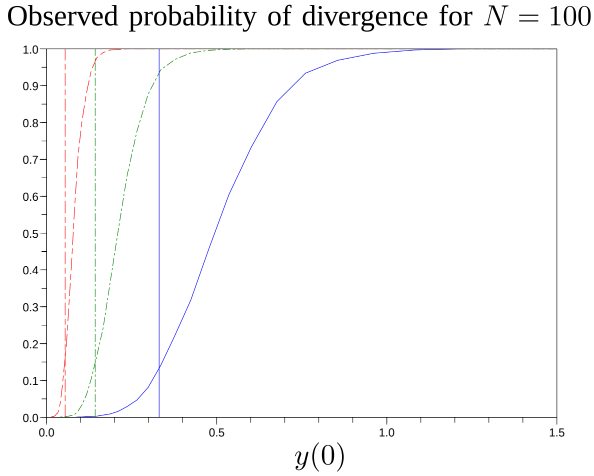

Fig. 10. Probability of divergence, uniform noise, $N = 100$, $\sigma_x^2 = 1$, $\mu = 0.0107722$ (solid), $\mu = 0.0232079$ (long dash), and $\mu = 0.05$ (short dash), $\sigma_z^2 = 0.01$, $L = 10^3$, $N_{\rm it} = 10^4$. The vertical lines show the values of $y(0)_{\rm max}$ computed for each value of μ.

iterations. As one would expect, the observed probability of divergence is less affected by the number of iterations when the noise variance is lower, and the initial condition $y(0)$ is larger, since these cases correspond to situations in which Approximation (10) is better.

The analytical results derived here and the experimental verification of the possibility of steady-state instability provide important guidelines for the use of the LMF algorithm in practical applications. First, the step size used should be a small fraction of the maximum step size $\mu_{\rm max}$ derived from the theory, given the expected range of values of $y(0)_{\rm max}$ inferred from the available knowledge about the problem at hand. Second, depending on the degree of confidence in the available information, it may be advisable to incorporate some form of re-initialization procedure to be applied if, for instance, the error signal starts diverging.

VI. Conclusions

This paper presented a new convergence analysis for the LMF adaptive algorithm, based on mean-square arguments. Although [1] showed that LMF with Gaussian regressors is not mean-square stable for any step size, the present analysis shows that mean-square results may still be very useful to predict the region of useful step sizes for a nonlinear adaptive filter, providing useful information at a much lower cost than the model in [1]. The analysis further improves previous results in that the dependence of stability on the initial conditions is explicitly shown, for additive noise having any even p.d.f. The results reveal a relationship between the initial conditions and

TABLE I

MAXIMUM STEP SIZES μ_{MAX} FOR DIFFERENT CONDITIONS, FOR GAUSSIAN REGRESSORS.

Case	N	σ_x^2	$y(0)$	σ_z^2	μ_{max}	$\mu_{1\%}$	$\mu_{99\%}$
Uniform noise							
1	10	0.1	0.1	0.01	9.7357	6.1	12.3
2	10	0.1	0.1	0.1	1.18	0.67	1.16
3	100	0.1	0.1	0.01	1.12753	0.93	1.78
4	100	0.1	0.1	0.1	0.14406	0.113	0.182
5	1000	0.1	0.1	0.01	0.131606	0.113	0.173
6	1000	0.1	0.1	0.1	0.014744	0.0148	0.0183
7	10	1	0.1	0.01	0.21549	0.135	1.38
8	10	1	0.1	0.1	0.0974	0.0614	0.172
9	100	1	0.1	0.01	0.0316	0.0210	0.0744
10	100	1	0.1	0.1	0.0127528	0.00931	0.0182
11	1000	1	0.1	0.01	0.00332	0.00211	0.00789
12	1000	1	0.1	0.1	0.0013161	0.00120	0.00278
13	10	0.1	1	0.01	2.1549	1.48	10.9
14	10	0.1	1	0.1	0.974	0.610	1.23
15	100	0.1	1	0.01	0.3164	0.202	0.757
16	100	0.1	1	0.1	0.12753	0.0928	0.177
17	1000	0.1	1	0.01	0.033195	0.0202	0.0586
18	1000	0.1	1	0.1	0.0131606	0.0113	0.0173
19	10	1	1	0.01	0.0246	0.0178	0.337
20	10	1	1	0.1	0.0215	0.0146	0.120
21	100	1	1	0.01	0.00370	0.00231	0.0108
22	100	1	1	0.1	0.00316	0.00199	0.00741
23	1000	1	1	0.01	0.000390	0.000221	0.000821
24	1000	1	1	0.1	0.000332	0.000204	0.000557
Gaussian noise							
25	10	1	0.01	0.01	0.7692308	0.352	0.848
26	10	1	0.01	0.1	0.0779754	0.0356	0.0849
27	100	1	0.01	0.01	0.0970874	0.0700	0.113
28	100	1	0.01	0.1	0.0128305	0.00701	0.0112
29	1000	1	0.01	0.01	0.0099701	0.00888	0.0135
30	1000	1	0.01	0.01	0.0009970	0.000892	0.00143
31	100	1	0.1	0.01	0.0313480	0.0197	0.0694
32	100	4	0.1	0.01	0.0022479	0.00124	0.00632
33	100	0.5	1	0.01	0.0145243	0.00822	0.0416

the step size in determining convergence. The smaller the value of μ, the larger the allowable values for the initial weight error vector. Simulations show that the theoretical predictions can be useful for design purposes.

APPENDIX I

CONDITIONS FOR $b^2 \leq 3c(1-a)$ IN (32)

Using the values of a, b and c from (21) in the expression $b^2 \leq 3c(1-a)$, yields:

$$B_1^2\mu^2 - 2B_1B_2\mu^3 + B_2^2\mu^4 \leq 3C\mu^2 - 3A_1C\mu^3 + 3A_2C\mu^4 \tag{39}$$

Using now A_1, A_2, B_1, B_2 and C from (22) in (39) yields

$$\begin{aligned} &\left\{36\,\sigma_x^8\mu^2 - 180\,\sigma_z^2\sigma_x^{10}(3N+12)\mu^3 + 225\,\sigma_z^4\sigma_x^{12}(3N+12)^2\mu^4\right\} \\ &\leq \left\{3\sigma_x^8(15N+90)\mu^2 - 18\,\sigma_z^2\sigma_x^{10}(15N+90)\mu^3 + 45\ \mathrm{E}[z^4(n)]\sigma_x^{12}(15N+90)(N+2)\mu^4\right\} \end{aligned} \tag{40}$$

Dividing (40) by $3\sigma_x^8\mu^2$, results

$$\begin{aligned} &\left\{12 - 60\,\sigma_z^2\sigma_x^2(3N+12)\mu + 75\,\sigma_z^4\sigma_x^4(3N+12)^2\mu^2\right\} \\ &\leq \left\{(15N+90) - 6\,\sigma_z^2\sigma_x^2(15N+90)\mu + 15\ \mathrm{E}[z^4(n)]\sigma_x^4(15N+90)(N+2)\mu^2\right\} \end{aligned} \tag{41}$$

For large N, the following approximations are valid (for $F = 90$ or $F = 78$):

$$\begin{aligned} &N + 2 \approx N \\ &3N + 12 \approx 3N \\ &15N + F \approx 15N \end{aligned} \tag{42}$$

Therefore, for large N and defining $p = \sigma_x^2 N\mu$, (41) can be written as

$$225\Big\{\ \mathrm{E}[z^4(n)] - 3\sigma_z^4\Big\}p^2 + 90\sigma_z^2 p + 15N \geq 0 \tag{43}$$

Straightforward analysis of inequality (43) shows that it is satisfied for any $\mu \geq 0$ if $\mathrm{E}[z^4(n)] \geq 3\sigma_z^4$ (i.e., $z(n)$ is Gaussian or has a distribution with longer tails than the Gaussian). If $\mathrm{E}[z^4(n)] < 3\sigma_z^4$ (i.e., $z(n)$ has a distribution with shorter tails than the Gaussian), (43) is satisfied for

$$0 \leq \mu \leq \frac{-\beta - \sqrt{\Delta_1}}{2N\sigma_x^2\alpha}, \tag{44}$$

where

$$\alpha = 225\{\mathrm{E}[z^4(n)] - 3\sigma_z^4\} \tag{45}$$

$$\beta = 90\sigma_z^2 \tag{46}$$

$$\Delta_1 = 900\left\{9\sigma_z^4 - 15N\left\{\,\mathrm{E}[z^4(n)] - 3\sigma_z^4\right\}\right\}. \tag{47}$$

APPENDIX II

CONDITIONS FOR $3c(1-a) < b^2 < 3c(2-a)$ IN (34)

The left inequality is the opposite of (32). Therefore from (43) we conclude that the convergence cannot be non-monotonic for large N if $\mathrm{E}[z^4(n)] \geq 3\sigma_z^4$. On the other hand, if $\mathrm{E}[z^4(n)] < 3\sigma_z^4$, the left inequality in (34) will be satisfied for

$$\mu > \frac{-\beta - \sqrt{\Delta_1}}{2N\sigma_x^2\alpha}$$

(see (44)).

If the above conditions are satisfied, we may check the second condition in (34). Using the values of a, b and c from (21) in the expression $b^2 < 3c(2-a)$, yields:

$$B_1^2\mu^2 - 2B_1B_2\mu^3 + B_2^2\mu^4 < 6C\mu^2 - 3A_1C\mu^3 + 3A_2C\mu^4 \tag{48}$$

Using now A_1, A_2, B_1, B_2 and C from (22) in (48) yields

$$\begin{aligned}&\left\{36\,\sigma_x^8\mu^2 - 180\,\sigma_z^2\sigma_x^{10}(3N+12)\mu^3 + 225\,\sigma_z^4\sigma_x^{12}(3N+12)^2\mu^4\right\}\\ &< \left\{6\,\sigma_x^8(15N+90)\mu^2 - 18\,\sigma_z^2\sigma_x^{10}(15N+90)\mu^3 + 45\ \mathrm{E}[z^4(n)]\sigma_x^{12}(15N+90)(N+2)\mu^4\right\}\end{aligned} \tag{49}$$

Dividing (49) by $3\sigma_x^8\mu^2$, results in

$$\begin{aligned}&\left\{12 - 60\,\sigma_z^2\sigma_x^2(3N+12)\mu + 75\,\sigma_z^4\sigma_x^4(3N+12)^2\mu^2\right\}\\ &< \left\{2(15N+90) - 6\,\sigma_z^2\sigma_x^2(15N+90)\mu + 15\ \mathrm{E}[z^4(n)]\sigma_x^4(15N+90)(N+2)\mu^2\right\}\end{aligned} \tag{50}$$

Using again approximations (42) for large N and making $p = \sigma_x^2 N\mu$, (50) can be written as

$$225\Big\{\mathrm{E}[z^4(n)] - 3\sigma_z^4\Big\}p^2 + 90\sigma_z^2 p + 30N > 0. \tag{51}$$

Straightforward analysis of inequality (51) shows that for $\mathrm{E}[z^4(n)] < 3\sigma_z^4$, (51) is satisfied for

$$0 \leq \mu \leq \frac{-\beta - \sqrt{\Delta_2}}{2N\sigma_x^2\alpha}, \tag{52}$$

where α and β are given by (45) and (46), respectively, and Δ_2 is given by

$$\Delta_2 = 900\Bigg\{9\sigma_z^4 - 30N\Big\{\mathrm{E}[z^4(n)] - 3\sigma_z^4\Big\}\Bigg\}. \tag{53}$$

REFERENCES

[1] V. H. Nascimento and J. C. M. Bermudez, “Probability of divergence for the least-mean fourth (LMF) algorithm,” *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1376–1385, 2006.

[2] E. Walach and B. Widrow, “The least mean fourth (LMF) adaptive algorithm and its family,” *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 275–283, February 1984.

[3] A. Gersho, “Some aspects pf linear estimation with nonmean-square error criteria,” in *Proc. of the Asilomar Circuits Syst. Conf.*, Asilomar, CA, USA, Nov. 1969.

[4] J. A. Chambers, O. Tanrikulu, and A. G. Constantinides, “Least mean mixed-norm adaptive filtering,” *Electronics Letters*, vol. 30, no. 19, pp. 1574–1575, 15th September 1994.

[5] O. Tanrikulu and J. A. Chambers, “Convergence and steady-state properties of the least-mean mixed-norm (LMMN) adaptive algorithm,” *IEE Proc.-Vis. Image Signal Processing*, vol. 143, no. 3, pp. 137–142, June 1996.

[6] A. Zerguine, M. Bettayeb, and C. F. N. Cowan, “Hibrid LMS-LMF algorithm for adaptive echo cancellation,” *Proc. Inst. Elect. Eng. – Vis. Image Signal Process.*, vol. 146, no. 4, pp. 173–180, Aug. 1999.

[7] S. V. Narasimham, “Application of the least mean fourth (LMF) adaptive algorithm to signals associated with gaussian noise,” *Int. J. Electron.*, vol. 62, no. 6, pp. 895–913, 1987.

[8] M. K. Chan, A. Zerguine, and C. F. N. Cowan, “An optimized normalised LMF algorithm for sub-gaussian noise,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6. IEEE, Apr. 2003, pp. 377–380.

[9] P. I. Hübscher and J. C. M. Bermudez, “An improved statistical analysis of the least mean fourth (lmf) adaptive algorithm,” *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 664–671, March 2003.

[10] W. A. Sethares, “Adaptive algorithms with nonlinear data and error functions,” *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2199–2206, September 1992.

[11] N. Kalouptsidis and S. Theodoridis, *Adaptive System Identification and Signal Processing Algorithms*. New Jersey: Prentice Hall, 1993.

[12] R. Sharma, W. A. Sethares, and J. A. Bucklew, “Asymptotic analysis of stochastic gradient-based adaptive filtering algorithms with general cost functions,” *IEEE Transactions on Signal Processing*, vol. 44, no. 9, pp. 2186–2194, September 1996.

[13] V. Solo, “The stability of LMS,” *IEEE Trans. Signal Processing*, vol. 45, no. 12, pp. 3017–3026, Dec. 1997.

[14] V. H. Nascimento and A. H. Sayed, “On the learning mechanism of adaptive filters,” *IEEE Trans. Signal Processing*, vol. 48, no. 6, pp. 1609–1625, June 2000.

[15] S. H. Cho, S. D. Kim, and K. Y. Jeon, “Statistical convergence of the adaptive least mean fourth algorithm,” in *Proc. of the ICSP’96*. IEEE, 1996, pp. 610–613.

[16] T. Al-Naffouri and A. Sayed, “Transient analysis of adaptive filters,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6. IEEE, 2001, pp. 3869–3872.

[17] S. Koike, “Stability conditions for adaptive algorithms with non-quadratic error criteria,” in *Proceedings of EUSIPCO 2000*, vol. II. Finland: IEEE, September 2000, pp. 131–134.

[18] P. I. Hübscher, V. H. Nascimento, and J. C. M. Bermudez, “New results on the stability analysis of the LMF (Least Mean Fourth) adaptive algorithm,” in *Proc. of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Hong Kong: IEEE, 2003.

[19] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice-Hall, 2002.

[20] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley-Interscience, 2003.

[21] J. Minkoff, “Comment: On the unnecessary assumption of statistical independence between reference signal and filter weights in feedforward adaptive systems,” *IEEE Transactions on Signal Processing*, vol. 49, no. 5, p. 1109, May 2001.

[22] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[23] S. Koike, “Analysis of the least mean fourth algorithm based on gaussian distributed tap weights,” in *IEEE International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2001*. Nashville, USA: IEEE, Nov. 2001, pp. 20–25.

[24] N. J. Bershad, P. Celka, and J. M. Vesin, “Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for gaussian data and noisy input and output measurements,” *IEEE Transactions on Signal Processing*, vol. 47, no. 3, pp. 675–689, March 1999.

[25] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*. NY: McGraw-Hill, 1968.

[26] L. W. Couch II, *Digital and Analog Communication Systems*, 2nd ed. New York: MacMillan, 1987.

TABLE II

OBSERVED PROBABILITY OF DIVERGENCE AS A FUNCTION OF THE NUMBER OF ITERATIONS.

N_{it}	$P_{\text{d,o}}$
$N = 10$, $\mu = 0.01$, $L = 10^3$, $\sigma_x^2 = 1$, $y(0) = 1$, $\sigma_z^2 = 0.01$, uniform noise	
10^4	0.0%
$5 \cdot 10^4$	0.1%
10^5	0.0%
$2 \cdot 10^5$	0.1%
$N = 10$, $\mu = 0.0246$, $L = 10^3$, $\sigma_x^2 = 1$, $y(0) = 1$, $\sigma_z^2 = 0.01$, uniform noise	
10^4	4.0%
$5 \cdot 10^4$	6.0%
10^5	6.5%
$2 \cdot 10^5$	5.7%
$N = 1000$, $\mu = 0.00025$, $L = 10^3$, $\sigma_x^2 = 1$, $y(0) = 1$, $\sigma_z^2 = 0.1$, uniform noise	
10^4	3.6%
$5 \cdot 10^4$	3.4%
10^5	4.5%
10^6	3.0%
$N = 1000$, $\mu = 0.000332$, $L = 10^3$, $\sigma_x^2 = 1$, $y(0) = 1$, $\sigma_z^2 = 0.1$, uniform noise	
10^4	22.2%
$5 \cdot 10^4$	19.6%
10^5	21.8%
10^6	23.7%
$N = 10$, $\mu = 0.0617917$, $L = 10^3$, $\sigma_x^2 = 1$, $y(0) = 0.1$, $\sigma_z^2 = 0.1$, uniform noise	
10^4	0.8%
$3 \cdot 10^5$	6.7%
$5 \cdot 10^5$	9.8%
10^6	18.5%
$2 \cdot 10^6$	35.4%