WHEN IS THE LEAST-MEAN FOURTH ALGORITHM MEAN-SQUARE STABLE?

Vítor H. Nascimento[*]

Deptartment of Electronic Systems Engineering
University of São Paulo
São Paulo - SP - Brazil

José Carlos M. Bermudez[†]

Deptartment of Electrical Engineering
Federal University of Santa Catarina
Florianópolis - SC - Brazil

ABSTRACT

We show that the least-mean fourth and the least-mean mixed-norm algorithms are not mean-square stable when the input regressor is Gaussian-distributed. For the LMF algorithm, we propose an upper bound for the algorithm's probability of divergence, given the input and noise statistics, the step-size and the filter length. We show that the upper bound can also be used for the LMMN algorithm.

1. INTRODUCTION

Since its introduction by Walach and Widrow in 1984 [1], the least-mean fourth (LMF) algorithm has been the subject of several publications. The reduced mean-square error of LMF in comparison to the least-mean squares algorithm (LMS) for sub-Gaussian measurement noise was considered in [1] itself, and analyses for Gaussian regressors were given in [2, 3, 4], among others.

The original analysis in [1] considered only the algorithm behavior close to steady-state, concluding that LMF outperforms LMS for measurement noise following a distribution with shorter tail than the Gaussian. The algorithm's transient behavior was considered in [2, 3]. An example was given in [3] showing that the compromise between steady-state error and convergence speed could be better solved using the LMF algorithm, rather than LMS, even for Gaussian noise. The LMF stability was also considered in [5], where it is proven that the algorithm is stable for sufficiently small step-size if the regressor and noise sequences are bounded (i.e., belong to ℓ_∞.)

The potential instability of LMF was recognized right from the start. Walach and Widrow noted in [1] that the stability region should depend on the initial conditions. Since they only studied the algorithm in steady-state, this dependence did not appear in their stability condition. This point was clarified later in [2, 4], in which approximate expressions for this dependence were proposed for Gaussian regressors. Other works took another path, and proposed a combination of LMS and LMF: the least mean mixed-norm algorithm (LMMN) [6].

We recently revisited the mean-square stability of LMF. In [7] we prove that the scalar (one coefficient) version of LMF is not mean-square stable when the regressors follow a slightly modified Gaussian distribution, showing that a given realization has always a nonzero probability of divergence. This result is further expanded in [8] (recently submitted for possible publication), where we propose an approximation for the LMF probability of divergence for filters of any length, always with Gaussian regressors. In the present paper we review some of the results proposed in [8], provide some new examples comparing our approximate expression for the probability of divergence with simulations, now considering tap-delay line regressors and different choices of measurement noise, and show that some of our results also apply to the LMMN algorithm.

The LMF algorithm computes approximations $\boldsymbol{W}(n)$ to a vector $\boldsymbol{W}_o$ through the recursion

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu e(n)^3 \boldsymbol{X}(n), \tag{1}$$

where $e(n) = d(n) - \boldsymbol{W}^T(n)\boldsymbol{X}(n)$ is the error between a desired signal $d(n)$ and the output of an FIR adaptive filter $\boldsymbol{W}(n)$ with an input regressor $\boldsymbol{X}(n)$.[1]

The LMMN algorithm uses a mixture of the LMF and LMS recursions,

$$\begin{aligned}\boldsymbol{W}_m(n+1) = \boldsymbol{W}_m(n) + \mu e_m(n)\big[\delta +\\ + 2(1-\delta)e_m^2(n)\big]\boldsymbol{X}(n),\end{aligned} \tag{2}$$

where, similarly, $e_m(n) = d(n) - \boldsymbol{W}_m^T(n)\boldsymbol{X}(n)$, and $0 \le \delta \le 1$ is a design parameter.

The main conclusion of our analysis in [7] is valid for both LMF and LMMN: they have a finite *region of attraction* for the initial weight vector because they have a third-order nonlinearity in their weight update equations. If the inputs (regressors or noise) drive the estimation error to a

[*]e-mail: vitor@lps.usp.br. Work supported in part by CNPq (Brazilian Ministry of Science and Technology) under grant 303929/2002-2

[†]e-mail: j.bermudez@ieee.org. Work supported in part by CNPq under grants 308095/2003-0 and 472762/2003-6

[1]All vectors belong to $\mathbb{R}^M$ in this paper.

value too large at any iteration of a given realization, the weight estimates are taken out of this region of attraction and the algorithm diverges. Thus, both LMF and LMMN are sensitive to bursts of large noise, or to unpredicted increases in the input power. Some sort of normalization, for example as proposed in [9], might prove useful.

2. DIVERGENCE IN LMF

The LMF and LMS algorithms behave somewhat differently when they start to diverge. LMS has a ”soft” transition between convergence and divergence. There is a range of step-sizes for which LMS converges in the mean-square sense, a range for which it diverges in the mean-square, but converges with probability one (for the zero measurement noise case) and a range for which the weigth error vector diverges to infinity with probability one [10]. The LMF behavior depends on the step-size and on the regressor probability distribution. If the regressors and measurement noise have probability density functions (pdfs) with bounded support[2], then there is a range of step-sizes for which the algorithm is stable [5].

For pdfs with unbounded support (such as the Gaussian), LMF will always have a nonzero probability of divergence. The filter may have a nice behavior in most realizations, but the weight estimates will tend to infinity for a few realizations. This behavior is shown in Fig. 1. Here we show three realizations, two converging and one diverging[3]. Here $\boldsymbol{X}(n)$ was a unity variance white Gaussian vector, $\mu = 0.02$, and the filter length was $M = 10$.

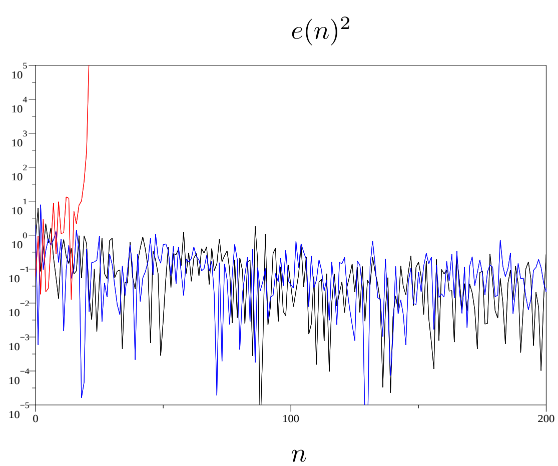

Fig. 1. Three runs of LMF with $M = 10$, true $\boldsymbol{W}_o = [0\ 0\ \ldots\ 0]^T$, $\boldsymbol{W}(0) = [1\ 0\ \ldots\ 0]^T$, $\mu = 0.02$, $\boldsymbol{X}(n)$ zero-mean Gaussian with covariance equal to I.

[2]If all values are strictly bounded.

[3]Note that divergence does not take long to become clear, which we noticed to be a typical behavior for LMF.

Given this behavior, we define divergence as follows.

Definition 1 (Divergence). In this work we say that a realization (a single run) of the LMF recursion diverged if $\lim_{n\to\infty} \|\boldsymbol{W}(n)\| = \infty$. We shall also say that a realization of the algorithm converged if it did not diverge.

Any vector norm could be used in this definition. We shall use the Euclidean norm.

3. PROBABILITY OF DIVERGENCE FOR LMF

In [8] we pose the following question: given the initial condition $\boldsymbol{W}(0)$, the step-size μ, the filter length M, and the noise and regressor statistics, what is the probability that a realization of the filter will diverge?

Our solution is an iterative procedure that computes an approximation for the probability of divergence. We show here briefly how the approximation was derived. First recall that the Wiener solution to a linear estimation problem is such that

$$d(n) = \boldsymbol{W}_o^T \boldsymbol{X}(n) + e_0(n),$$

where $e_0(n)$ is uncorrelated with $\boldsymbol{X}(n)$. We assume that $e_0(n)$ is also *independent* of $\boldsymbol{X}(n)$, and that $\boldsymbol{X}(n)$ is independent of $\boldsymbol{X}(k)$, for $k \neq n$.[4] Defining the weight error vector $\boldsymbol{V}(n) \triangleq \boldsymbol{W}_o - \boldsymbol{W}(n)$ and $p(n) \triangleq \boldsymbol{V}^T(n)\boldsymbol{X}(n)$, (1) may be rewritten as

$$\begin{aligned} \boldsymbol{V}(n+1) = \boldsymbol{V}(n) - \mu\Big[p(n)^3 + 3p(n)^2 e_0(n) + \\ + 3p(n)e_0(n)^2 + e_0(n)^3\Big]\boldsymbol{X}(n). \end{aligned} \tag{3}$$

From this, a recursion for $y(n) \triangleq \boldsymbol{V}^T(n)\boldsymbol{V}(n)$ is easily obtained. Our goal is to estimate the probability that $\lim_{n\to\infty} y(n) = \infty$.

In order to simplify our task, we assume that $y(n) \approx \mathrm{E}\{y(n)|\ y(n-1),\ \boldsymbol{X}(n)\}$, where $E\{\cdot\}$ means statistical expectation. This approximation replaces the noise $e_0(n)$ and its powers by their means (in our simulations we observed that the influence of the noise on the probability of divergence is secondary to the influence of $\boldsymbol{X}(n)$). Assuming that $\mathrm{E}\, e_0(n) = \mathrm{E}\, e_0(n)^3 = \mathrm{E}\, e_0(n)^5 = 0$, and defining $\sigma_0^2 = \mathrm{E}\, e_0(n)^2$, $\psi_0^4 = \mathrm{E}\, e_0(n)^4$ and $\eta_0^6 = \mathrm{E}\, e_0(n)^6$, the recursion for $y(n)$ becomes

$$\begin{aligned} y(n+1) = y(n) - 2\mu\Big[p(n)^4 + 3p(n)^2\sigma_0^2\Big] + \\ + \mu^2\Big[p(n)^6 + 15p(n)^4\sigma_0^2 + 15p(n)^2\psi_0^4 + \eta_0^6\Big]\|\boldsymbol{X}(n)\|^2. \end{aligned} \tag{4}$$

[4]These are the standard independence assumptions [11].

Under the assumptions of iid, Gaussian $\boldsymbol{X}(n)$, we approximate (the proof for the equality is given in [8])

$$\frac{\left(\boldsymbol{V}(n)^T\boldsymbol{X}(n)\right)^{2k}}{\left(\|\boldsymbol{V}(n)\|\|\boldsymbol{X}(n)\|\right)^{2k}} \approx \mathrm{E}\left\{\frac{\left(\boldsymbol{V}(n)^T\boldsymbol{X}(n)\right)^{2k}}{\left(\|\boldsymbol{V}(n)\|\|\boldsymbol{X}(n)\|\right)^{2k}}\right\} \triangleq \alpha_M(k), \tag{5}$$

with $\alpha_M(1) = \dfrac{1}{M}$, $\alpha_M(2) = \dfrac{3}{M(M+2)}$, and $\alpha_M(3) = \dfrac{15}{M(M+2)(M+4)}$.

Substituting $\left(\boldsymbol{V}(n)^T\boldsymbol{X}(n)\right)^{2k}$ by $\alpha_M(k)y(n)^k\|\boldsymbol{X}(n)\|^{2k}$ in (4), we obtain

$$\begin{aligned} y(n+1) \approx \Bigg[& 1-\mu\left(6\sigma_0^2-15\mu\psi_0^4\|\boldsymbol{X}(n)\|^2\right)\frac{\|\boldsymbol{X}(n)\|^2}{M} \\ & -3\mu\big(2-15\mu\sigma_0^2\|\boldsymbol{X}(n)\|^2\big)\frac{\|\boldsymbol{X}(n)\|^4}{M(M+2)}y(n) \\ & +15\mu^2\frac{\|\boldsymbol{X}(n)\|^8}{M(M+2)(M+4)}y(n)^2\Bigg]y(n)+\mu^2\eta_0^6\|\boldsymbol{X}(n)\|^2. \end{aligned} \tag{6}$$

Denote by $D(n)$ the term in square brackets. It can be shown that $D(n)$ is always nonnegative, and that (6) converges if $D(n) < 1$ always. Note that $D(n) < 1$ is a sufficient, but not necessary, condition for stability: $y(n)$ may grow occasionaly and still remain bounded. Thus, $P_c \triangleq \Pr\big\{0 < D(n) < 1 \text{ for all } n \geq 0\big\}$ is a lower bound for the probability of convergence, and $P_d \triangleq 1 - P_c$ is an upper bound for the probability of divergence. To evaluate P_c, we:

1. Find the probabilities $\Pr\big\{D(n) < 1|\ y(n) = \hat{y}(n)\big\}$, for $0 \leq n \leq N$, starting from a given $y(0)$,
2. Find an approximation $\hat{y}(n)$ for $y(n)$ in the previous item,
3. Make

$$P_c \approx \prod_{n=0}^{N} \Pr\big\{D(n) < 1|\ y(n) = \hat{y}(n)\big\}. \tag{7}$$

We tested different approximations $\hat{y}(n)$, and the choice that gave best results was

$$\begin{aligned} \hat{y}(n+1) &= \left[1-3\mu\sigma_x^2(\sigma_0^2+\sigma_x^2\hat{y}(n))\right]^2\hat{y}(n), \\ \hat{y}(0) &= y(0) = \mathrm{E}\,\|\boldsymbol{V}(0)\|^2, \end{aligned} \tag{8}$$

which is equivalent to define $\hat{y}(n) = \big\{\mathrm{E}\,\boldsymbol{V}(n)\big\}^T\big\{\mathrm{E}\,\boldsymbol{V}(n)\big\}$. Since $\boldsymbol{X}(n)$ is a Gaussian vector, $\|\boldsymbol{X}(n)\|^2$ follows a χ^2 distribution with M degrees of freedom, and the probabilities in (7) may be evaluated by

$$\Pr\left\{\|\boldsymbol{X}(n)\|^2 \leq z_0(n)\big|\ y(n) = \hat{y}(n)\right\},$$

where $z_0(n)$ is the only positive root of (if $\hat{y}(n) = 0$ and $\sigma_0^2 = 0$, $P_c = 1$)

$$\begin{aligned} Q(z) = 6\sigma_0^2 + \left(\frac{6}{M+2}\hat{y}(n) - 15\mu\psi_0^4\right)z - \\ -\frac{45\mu\sigma_0^2}{M+2}\hat{y}(n)z^2 - \frac{15\mu}{(M+2)(M+4)}\hat{y}(n)^2z^3. \end{aligned} \tag{9}$$

4. EXAMPLES

We now compare our approximation for the probability of divergence with actual probabilities observed in simulations of the LMF algorithm. In all simulations, we ran 10^5 realizations of the algorithm, starting with $y(0) = 1$, and labelled a realization as ”diverging” whenever $\|\boldsymbol{V}(n)\| \geq 10^{100}$ for any n.[5] In [8], we only present simulations for which the vectors $\boldsymbol{X}(n)$ were independent. Here, we present simulations for which $\boldsymbol{X}(n)$ is the output of a tap-delay line, and for uniform measurement noise.

Fig. 2 shows the results for uniform noise. Fig. 3 shows the results for Gaussian noise, with regressors generated through a tap-delay line. Fig. 4 presents the observed probability of divergence for LMMN with tap-delay regressors and $\delta = 0.8$. The theoretical curve is obtained by the same algorithm as before, but with step-size $\mu_{\text{eq}} = 2\mu(1-\delta)$.

Observed probability of divergence and P_d, $M = 100$

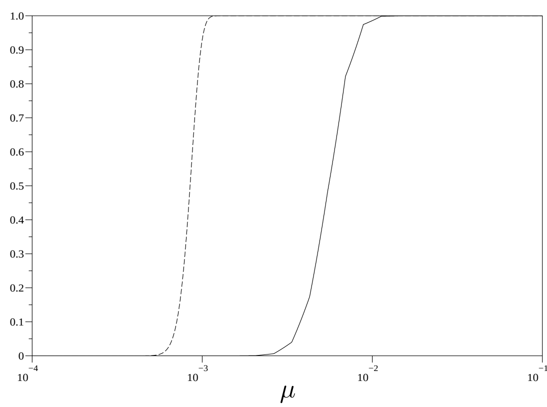

Fig. 2. Probability of divergence for LMF, iid regressors, uniform noise, $M = 100$, $\sigma_0^2 = 0.01$. Dark curve: observed probability in 10^5 runs with 100 time-steps. Broken curve: P_d.

5. CONCLUSIONS

We showed that the LMF and LMMN algorithms are not mean-square stable when the regressor vector is Gaussian

[5] Smaller choices for this bound did not change the results.

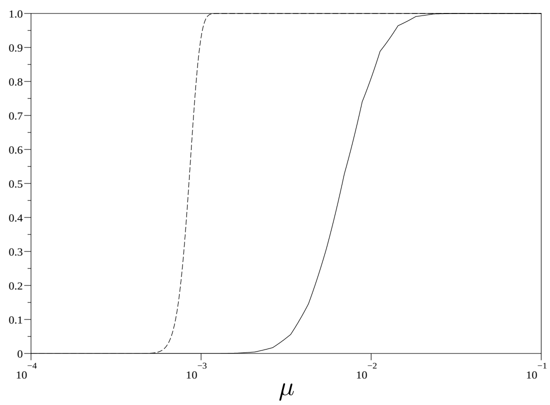

Fig. 3. Probability of divergence for LMF, tap-delay line regressors, Gaussian noise, $M = 100$, $\sigma_0^2 = 0.01$. Dark curve: observed probability in 10^5 runs with 100 time-steps. Broken curve: P_d.

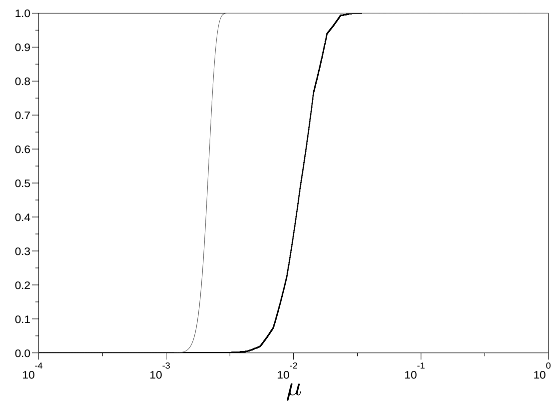

Fig. 4. Probability of divergence for LMMN, tap-delay line regressors, Gaussian noise, $M = 100$, $\sigma_0^2 = 0.01$, $\delta = 0.8$. Dark curve: observed probability in 10^5 runs with 100 time-steps. Light curve: P_d.

(or, in general, has not a pdf with a compact support), and proposed an upper bound for the probability of divergence of LMF. Since in practice all signals are bounded, our result means that the LMF algorithm is sensitive to variations in the power of input signals. The bound can also be used for LMMN, simply modifying the step-size μ used to compute the bound to $2\mu(1 - \delta)$.

Our upper bound gives designers a tool to predict when the performance of LMF and LMMN will be acceptable (recall that for small step-size, the performance of LMF and LMMN may be quite good with a large probability).

6. REFERENCES

[1] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 275–283, Mar. 1984.

[2] Shin'ichi 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, Nov. 2001, IEEE.

[3] P. I. Hübscher and J. C. M. Bermudez, "An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 664–671, Mar. 2003.

[4] 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 *ICASSP 2003*, vol. VI, pp. VI 369–VI 372.

[5] R. Sharma, W. A. Sethares, and J. A. Buclew, "Asymptotic analysis of stochastic gradient-based adaptive filtering algorithms with general cost functions," *IEEE Trans. Signal Processing*, vol. 44, no. 9, pp. 2186–2194, Sept. 1996.

[6] J. A. Chambers, O. Tanrikulu, and A. G. Constantinides, "Least mean mixed-norm adaptive filtering," *Electronics Letters*, vol. 30, no. 19, Sept. 15 1994.

[7] V. H. Nascimento and J. C. M. Bermudez, "On the stability of the least-mean fourth (LMF) algorithm," in *Anais do XXI Simpósio Brasileiro de Telecomunicações*, 2004, pp. 1–5, http://www.lps.usp.br/ vitor/sbrt2004.pdf.

[8] V. H. Nascimento and J. C. M. Bermudez, "Probability of divergence for the least-mean fourth (LMF) algorithm," Submitted for possible publication to the IEEE Trans. on Signal Processing, 2004.

[9] M. K. Chan, A. Zerguine, and C. F. N. Cowan, "An optimized normalised LMF algorithm for sub-Gaussian noise," in *ICASSP 2003*, Hong Kong, vol. VI, pp. VI–377–80.

[10] 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.

[11] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-Interscience, 2003.