A Model for the Behavior of the Least Mean Kurtosis (LMK) Adaptive Algorithm with Gaussian Inputs

Pedro I. Hübscher and José C. M. Bermudez*
Federal University of Santa Catarina, Department of Electrical Engineering
Florianópolis - SC - Brasil
e-mails: pedro@eel.ufsc.br, j.bermudez@ieee.org

Abstract—**The LMK algorithm is a stochastic gradient algorithm which seeks to minimize the negated kurtosis of the error signal. It outperforms the LMS algorithm in several applications of practical interest, with little increase in computational complexity. This paper presents a statistical analysis of the Least Mean Kurtosis (LMK) adaptive algorithm. Deterministic nonlinear recursive equations are derived for the mean weight behavior and for the weight error correlation matrix, for a Gaussian reference signal and slow learning. The new model describes the algorithm behavior during transient and steady-state for a white measurement noise with any even probability density function (pdf). The accuracy of the model is demonstrated by Monte Carlo simulations.**

I. INTRODUCTION

The LMS algorithm is largely used in real-time adaptive filtering applications for communications, control engineering, processing of biological signals and, more recently, active noise [1] and vibration [2] control. It is very popular due to its simplicity. Analytical models are available for predicting its behavior under different input conditions, facilitating its design. On the other hand, adaptive algorithms based on higher order moments of the error signal have been shown to outperform LMS in some important applications. The practical use of such algorithms, however, has been largely restricted due to the lack of accurate analytical models to predict their behavior. The Least Mean Kurtosis (LMK) algorithm is one of such algorithms [3].

The LMK algorithm seeks to minimize an approximation for the negative of the error signal *kurtosis*. It belongs to the family of stochastic gradient algorithms [3], [4]. The *kurtosis* is related to the fourth order cumulant of the error. Since cumulants of orders greater than two are equal to zero for zero-mean Gaussian processes, a *kurtosis*-based cost function makes the algorithm convergence behavior independent of Gaussian measurement noise. The measurement noise can frequently be considered to be composed of contributions from several additive independent sources. Thus, its *pdf* tends to a Gaussian (Central Limit Theorem [5]). This fact has raised interest for the LMK algorithm.

Previous results [3], [6], [7] show that the LMK algorithm can outperform the LMS algorithm, especially when noise contamination degrades the algorithm convergence. In addition, the computational complexity of both algorithms are very similar for high order adaptive filters, as LMK requires $2N+5$ multiplications and $N+3$ additions per iteration, while LMS requires $2N+1$ multiplications and $N+1$ additions per iteration for a filter of length N. [7] presented an analytical model for the LMK algorithm behavior, which is valid only for a white Gaussian reference input.

The available results on the LMK algorithm show that it has great potential for several practical applications. However, such conclusions are largely based on simulation results, with little analytical backup. For instance, implementation of the algorithm requires an approximation for the mean squared error. Such approximation modifies the behavior expected from a *kurtosis*-based algorithm.

This paper presents a new analytical model for the LMK algorithm behavior, which is valid for correlated reference inputs. Recursive deterministic equations are derived for the behaviors of the mean weight vector, the weight error correlation matrix and the mean square error (MSE). The model assumes a zero-mean, wide-sense stationary Gaussian reference signal and any zero-mean white additive noise with an even probability density function (*pdf*). The new model permits to study the effects of initialization and signal-to-noise ratio (*SNR*) on the algorithm behavior. In addition, it shows that the approximation made on the *kurtosis* expression to allow for a practical implementation makes the algorithm behavior dependent on both the input and noise statistics, even when both are Gaussian. On the other hand, it is verified that the LMK algorithm can outperform the LMS algorithm even for Gaussian reference signals. Monte Carlo simulations illustrate the model's accuracy for both the transient and steady-state phases of adaptation.

II. THE LMK ALGORITHM

Figure 1 shows the block diagram of the problem studied. $W^0 = [w_1^0, w_2^0, ..., w_N^0]^T$ is the vector of the impulse response of a linear system. $W(n) = [w_1(n), w_2(n), ..., w_N(n)]^T$ is the weight vector of the adaptive transversal FIR linear filter. $x(n)$ is assumed stationary, zero-mean and Gaussian with variance σ_x^2. $X(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ is the observed data vector. $z(n)$ is the measurement noise, assumed stationary, white, zero-mean with variance σ_z^2, uncorrelated with any other signal and with an even *pdf* ($f_z(z) = f_z(-z)$). $y(n)$ is the adaptive filter output and $e(n)$ is the error signal to be minimized in the *kurtosis* sense.

The ideal performance surface of the LMK algorithm is given by the negative of the fourth order cumulant of the error signal

* This work has been supported in part by INPE (Brazilian Space Research Institute) and by CNPq (Brazilian Ministry of Science and Technology) under Grant No. 352084/92-8.

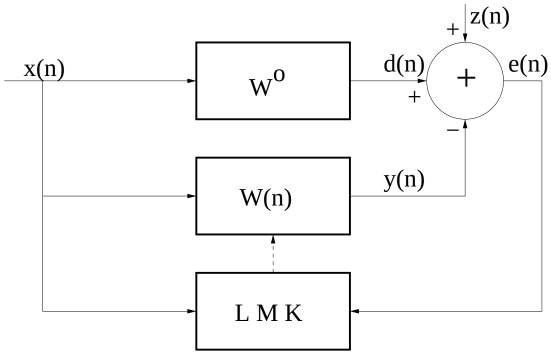

Fig. 1. Block Diagram (System Identification).

[3]:

$$J_{LMK}(n) = 3E^2[e^2(n)] - E[e^4(n)] \quad (1)$$

Using a stochastic approximation for the gradient of (1) with respect to the weight vector, the weight update equation for LMK algorithm is given by [3]:

$$W(n+1) = W(n) + \mu \left\{3E[e^2(n)]e(n) - e^3(n)\right\} X(n) \quad (2)$$

where μ is the step size. Defining the weight error vector $V(n) = W(n) - W^0$ about optimal solution, (2) can be written as:

$$V(n+1) = V(n) + \mu \left\{3E[e^2(n)]e(n) - e^3(n)\right\} X(n) \quad (3)$$

For real-time algorithm implementation, $E[e^2(n)]$ has to be estimated. This can be done using the recursion [3]

$$E[e^2(n)] = \beta E[e^2(n-1)] + e^2(n), \quad 0 < \beta < 1 \quad (4)$$

with $E[e^2(-1)] = 0$. A non-recursive estimation, which is accurate for small β, is obtained by truncating the solution of (4) to three terms:

$$E[e^2(n)] \approx e^2(n) + \beta e^2(n-1) + \beta^2 e^2(n-2) \quad (5)$$

Algorithm implementations using (4) or (5) do not show significant differences in behavior for practical parameter values. Thus, (5) is employed in the analysis.

III. MEAN WEIGHT BEHAVIOR

Using (5) in (3) yields:

$$\begin{aligned} V(n+1) = V(n) + \mu\big[&2e^2(n) + 3\beta e^2(n-1) \\ &+ 3\beta^2 e^2(n-2)\big]e(n)X(n) \end{aligned} \quad (6)$$

Taking the expected value of (6):

$$\begin{aligned} &E[V(n+1)] = E[V(n)] \\ &+ \mu\Big[2E[e^3(n)X(n)] + 3\beta E[e^2(n-1)e(n)X(n)] \\ &+ 3\beta^2 E[e^2(n-2)e(n)X(n)]\Big] \end{aligned} \quad (7)$$

From Fig. 1, $e(n) = z(n) - X^T(n)V(n)$. As $z(n)$ is zero-mean, *i.i.d.* and has odd moments equal to zero, the expected values of (7) are given by:

$$\begin{aligned} &E[e^3(n)X(n)] = -3E[z^2(n)]E[X(n)X^T(n)V(n)] \\ &- E[(X^T(n)V(n))^3X(n)] \\ \\ &E[e^2(n-1)e(n)X(n)] \\ &= -E[z^2(n-1)E[X(n)X^T(n)V(n)] \\ &- E[(X^T(n-1)V(n-1))^2X^T(n)V(n)X(n)] \\ \\ &E[e^2(n-2)e(n)X(n)] \\ &= -E[z^2(n-2)E[X(n)X^T(n)V(n)] \\ &- E[(X^T(n-2)V(n-2))^2X^T(n)V(n)X(n)] \end{aligned} \quad (8)$$

Since the *pdf* of $V(n)$ is not known, statistical assumptions are required to proceed with the analysis. It is assumed that: **A1:** The statistical dependence of weight and data vectors is not as significant as the statistical dependence of delayed data vectors in determining the algorithm behavior.

This assumption has been supported by extensive numerical simulations. The first expected value on the *rhs* of (8) is then given by $E[X(n)X^T(n)]E[V(n)]$=$RE[V(n)]$. The second term requires more elaborate calculations. Conditioning on V(n) and using **A1** yields

$$\begin{aligned} &E[(X^T(n)V(n))^3X(n)|V(n)] \\ &= E[(X^T(n)V(n))^2X(n)X^T(n)V(n)|V(n)] \\ &= E[(X^T(n)V(n))^2X(n)X^T(n)|V(n)]V(n) \end{aligned} \quad (9)$$

Using the same methodology used to obtain [8, A.13], (9) can be written as

$$\begin{aligned} &E[(y_1)^2X(n)X^T(n)|V(n)] \\ &= E[X(n)X^T(n)|V(n)]E[(y_1)^2|V(n)] \\ &+ E[y_1X(n)|V(n)]E[y_1X^T(n)|V(n)]B(y_1|V(n)) \end{aligned} \quad (10)$$

where $B(y_1|V(n)) = \frac{1}{E[y_1^2|V(n)]}\left(\frac{E[y_1^4|V(n)]}{E[y_1^2|V(n)]} - E[y_1^2|V(n)]\right)$ and $y_1 = X^T(n)V(n)$. Note that for $x(n)$ zero-mean Gaussian, $y_1(n)$ is also zero-mean Gaussian when conditioned on $V(n)$. Using **A1**, the terms in (10) are given by

$$\begin{aligned} &E[y_1^2|V(n)] = V^T(n)RV(n) \\ &E[y_1^4|V(n)] = 3E^2[y_1^2|V(n)] \\ &B(y_1) = 2 \\ &E[y_1X(n)|V(n)] = RV(n) \end{aligned} \quad (11)$$

Using the above results in (9) leads to

$$\begin{aligned} &E[(X^T(n)V(n))^3X(n)|V(n)] \\ &= V^T(n)RV(n)RV(n) + 2RV(n)V^T(n)RV(n) \end{aligned} \quad (12)$$

Averaging (12) over $V(n)$ requires extra approximations, as the *pdf* of $V(n)$ is not known. Assuming slow learning and N sufficiently large, the following approximation is used:

$$\begin{aligned} &E[V^T(n)RV(n)RV(n)] \\ &\approx E[V^T(n)RV(n)]RE[V(n)] \\ &= tr(RK(n))RE[V(n)] \end{aligned} \quad (13)$$

where $K(n) = E[V(n)V^T(n)]$ is the correlation matrix of $V(n)$ and $tr(.)$ is the trace of a matrix. Approximation (13) is based on the fact that each component $v_i(n)$ of $V(n)$ contributes to only N of the N^2 terms in $V^T(n)RV(n)$. Thus, for large N, each $v_i(n)$ can be considered weakly correlated with $V^T(n)RV(n)$. Using (13), the expected value of (12) becomes

$$E[(X^T(n)V(n))^3X(n)] \approx 3tr(RK(n))RE[V(n)] \quad (14)$$

The other terms in (8) are determined using the same approach, plus the assumption that $E[V(n-i)V^T(n-i)] \approx E[V(n)V^T(n-i)] \approx E[V(n)V^T(n)]$, for $i = 1, 2$ and for sufficiently small μ. Thus,

$$\begin{aligned} &E[(X^T(n-1)V(n-1))^2X^T(n)V(n)X(n)] \\ &= E[V^T(n-1)RV(n-1)RV(n) \\ &+ 2R_1V(n-1)V^T(n-1)R_{-1}V(n)] \\ &\approx tr(RK(n))RE[V(n)] \\ &+ 2tr(R_{-1}K(n))R_1E[V(n-1)] \end{aligned} \quad (15)$$

$$\begin{aligned} &E[(X^T(n-2)V(n-2))^2X^T(n)V(n)X(n)] \\ &= E[V^T(n-2)RV(n-2)RV(n) \\ &+ 2R_2V(n-2)V^T(n-2)R_{-2}V(n)] \\ &\approx tr(RK(n))RE[V(n)] \\ &+ 2tr(R_{-2}K(n))R_2E[V(n-2) \end{aligned} \quad (16)$$

where $R_i = E[X(n)X^T(n-i)]$ and $R_{-i} = E[X(n-i)X^T(n)]$, for $i = 1, 2$.

Using (14)-(16) in (7) and (8), leads to an expression for the mean weight behavior:

$$\begin{aligned} &E[V(n+1)] = E[V(n)] \\ &- \mu\left\{\left[6 + 3\beta + 3\beta^2\right]\left[\sigma_z^2 + tr(RK(n))\right]R\right\}E[V(n)] \\ &- 6\mu\beta tr(R_{-1}K(n))R_1E[V(n-1)] \\ &- 6\mu\beta^2 tr(R_{-2}K(n))R_2E[V(n-2)] \end{aligned} \quad (17)$$

Note from (17) that the stability of $E[V(n)]$ depends on the initialization $K(0) = V(0)V^T(0)$, which is related to the quadratic norm of the distance between $W(0)$ and W^o. A recursive expression for $K(n)$ is derived next for use in (17) and in the expression for the mean square error (MSE). An expression for the steady-state MSE in also derived.

IV. SECOND ORDER MOMENTS

Multiplying (6) by its transpose and taking the expected value yields

$$\begin{aligned} &E[V(n+1)V^T(n+1)] = E[V(n)V^T(n)] \\ &+ \mu E\left\{\left[2e^2(n) + 3\beta e^2(n-1) + 3\beta^2 e^2(n-2)\right]\right. \\ &\left. .e(n)\left[V(n)X^T(n) + X(n)V^T(n)\right]\right\} \\ &+ \mu^2 E\left\{\left[4e^6(n) + 12\beta e^2(n-1)e^4(n)\right.\right. \\ &+ 12\beta^2 e^2(n-2)e^4(n) + 9\beta^2 e^4(n-1)e^2(n) \\ &+ 18\beta^3 e^2(n-2)e^2(n-1)e^2(n) \\ &\left.\left. + 9\beta^4 e^4(n-2)e^2(n)\right]X(n)X^T(n)\right\} \end{aligned} \quad (18)$$

Using $e(n) = z(n) - X^T(n)V(n)$ in (18) and **A1**, first expectation yiels

$$\begin{aligned} &E[e^3(n)(V(n)X^T(n) + X(n)V^T(n))] \\ &= -3\sigma_z^2(RE[V(n)V^T(n)] + E[V(n)V^T(n)]R) \\ &- 2RE[V(n)V^T(n)RV(n)V^T(n)] \\ &- E[V^T(n)RV(n)RV(n)V^T(n)] \\ &- 2E[V(n)V^T(n)RV(n)V^T(n)]R \\ &- E[V(n)V^T(n)V^T(n)RV(n)]R \\ &= -3\sigma_z^2(RE[V(n)V^T(n)] + E[V(n)V^T(n)]R) \\ &- 3RE[V^T(n)RV(n)V(n)V^T(n)] \\ &- 3E[V(n)V^T(n)RV(n)V^T(n)]R \end{aligned} \quad (19)$$

Using the same reasoning used to obtain (13), the moments of $V(n)$ with order higher than 2 in (19) can be approximated by:

$$\begin{aligned} &E[V(n)V^T(n)RV(n)V^T(n)] \\ &= E[V^T(n)RV(n)V(n)V^T(n)] \\ &\approx E[V^T(n)RV(n)]E[V(n)V^T(n)] \\ &= tr(RK(n))K(n) \end{aligned} \quad (20)$$

Using (20) in (19) yields

$$\begin{aligned} &E[e^3(n)(V(n)X^T(n) + X(n)V^T(n))] \\ &\approx -3\left(\sigma_z^2 + tr(RK(n))\right)\left(RK(n) + K(n)R\right) \end{aligned} \quad (21)$$

Substituting $e(n-i) = z(n-i) - X^T(n-i)V(n-i)$, for $i = 1, 2$, using (20) and considering $E[V(n-i)V^T(n-i)] \approx E[V(n)V^T(n-i)] \approx E[V(n)V^T(n)]$, for $i = 1, 2$ and for small μ, the expected values of the other two terms multiplied by μ in (18) are given by

$$\begin{aligned} &E[e^2(n-1)e(n)\big(V(n)X^T(n) + X(n)V^T(n)\big)] \\ &\approx -\left(\sigma_z^2 + tr(RK(n))\right)\left(RK(n) + K(n)R\right) \\ &- tr\left(R_{-1}K(n) + R_1K(n)\right)\left(K(n)R_{-1} + R_1K(n)\right) \end{aligned} \quad (22)$$

$$E[e^2(n-2)e(n)\big(V(n)X^T(n)+X(n)V^T(n)\big)]$$
$$\approx -\Big(\sigma_z^2+tr(RK(n))\Big)\Big(RK(n)+K(n)R\Big) \quad (23)$$
$$-\,tr\Big(R_{-2}K(n)+R_2K(n)\Big)\big(K(n)R_{-2}+R_2K(n)\big)$$

Entering the expressions for $e(n)$, $e(n-1)$ and $e(n-2)$ in the term multiplying μ^2 in (18) leads to terms of the form $\mu^2 E[(X^T(n-i)V(n-i))^{2k}X(n)X^T(n)]$, for $i = 0,1,2$. Those terms corresponding to $k>1$ are neglected in the present analysis, as they have small influence both during transient (higher order moments) and in steady-state (small $V(n)$). The terms $E[(X^T(n-i)V(n-i))^2X(n)X^T(n)]$, $i=0,1,2$, are evaluated using the same methodology used to determine the mean weight behavior. Besides, it is considered that $E[V(n-i)V^T(n-i)] \approx E[V(n)V^T(n)]$, for $i=1,2$ and sufficiently small μ. Using these approximations, the expected values of the terms in μ^2 of (18) are given by:

$$E[e^6(n)X(n)X^T(n)] \approx E[z^6(n)]R$$
$$+\,15E[z^4(n)]\Big(tr(RK(n))R+2RK(n)R\Big) \quad (24)$$

$$E[e^2(n-i)e^4(n)X(n)X^T(n)]$$
$$\approx \Big[(6\sigma_z^4+E[z^4(n)])tr(RK(n))+\sigma_z^2E[z^4(n)]\Big]R \quad (25)$$
$$+\,12\sigma_z^4RK(n)R+2E[z^4(n)]R_iK(n)R_{-i},\ \ i=1,2$$

$$E[e^4(n-i)e^2(n)X(n)X^T(n)]$$
$$\approx \Big[(6\sigma_z^4+E[z^4(n)])tr(RK(n))+\sigma_z^2E[z^4(n)]\Big]R \quad (26)$$
$$+\,2E[z^4(n)]RK(n)R+12\sigma_z^4R_iK(n)R_{-i},\ \ i=1,2$$

$$E[e^2(n-2)e^2(n-1)e(n)X(n)X^T(n)]$$
$$\approx \big(\sigma_z^6+3\sigma_z^4tr(RK(n))\big)R \quad (27)$$
$$+\,2\sigma_z^4\big(RK(n)R+R_1K(n)R_{-1}+R_2K(n)R_{-2}\big)$$

Substituting (21) - (27) in (18), a recursive expression is obtained for the correlation matrix $K(n)=E[V(n)V^T(n)]$:

$$K(n+1)=K(n)$$
$$-\,\mu\Bigg\{C_0\Big[\sigma_z^2+tr(RK(n))\Big]\Big[K(n)R+RK(n)\Big]$$
$$+\,3\beta\Big[tr(R_{-1}K(n)+R_1K(n))$$
$$.(K(n)R_{-1}+R_1K(n))\Big]$$
$$+\,3\beta^2\Big[tr(R_{-2}K(n)+R_2K(n))$$
$$.(K(n)R_{-2}+R_2K(n))\Big]\Bigg\} \quad (28)$$
$$+\,\mu^2\Bigg\{C_1R+C_2RK(n)R+C_3R_1K(n)R_{-1}$$
$$+\,C_4R_2K(n)R_{-2}\Bigg\}$$

where:
$C_0=6+3\beta+3\beta^2$;
$C_1=4E[z^6(n)]+60E[z^4(n)]tr(RK(n))$ $+\Big(12\beta+21\beta^2+12\beta^4\Big)\Big\{tr(RK(n))\big(6\sigma_z^4+E[z^4(n)]\big)+\sigma_z^2E[z^4(n)]\Big\}+18\beta^3\Big(\sigma_z^6+3\sigma_z^4tr(RK(n))\Big)$;
$C_2=120E[z^4(n)]+144(\beta+\beta^2)\sigma_z^4$ $+18(\beta^2+\beta^4)E[z^4(n)]+36\beta^3\sigma_z^4$;
$C_3=24\beta E[z^4(n)]+108\beta^2\sigma_z^4+36\beta^3\sigma_z^4$;
$C_4=24\beta^2E[z^4(n)]+108\beta^4\sigma_z^4+36\beta^3\sigma_z^4$.

Equation (28) can be used in conjunction with (17) to determine the mean weight behavior. Note that (28) is a nonlinear equation in $K(n)$. Expression (28) evidences the dependence of the algorithm behavior on the noise distribution function. $E[z^4(n)]$ affects stability, convergence speed and steady-state behavior. $E[z^6(n)]$ affects only the steady-state behavior, since it appears only in the driving term of (28).

Using the results just derived, it is possible to determine the behavior of the MSE, which is given by

$$\xi_{LMK}(n)=E[e^2(n)]=\sigma_z^2+tr[RK(n)] \quad (29)$$

The analytical model for the LMK algorithm is then composed by expressions (17), (28) and (29). Though the LMK algorithm seeks to minimize a higher order function of the error, the MSE behavior is emphasized here because it is related to the error power. Thus, the MSE is the best measure to compare performances of algorithms that employ different figures of merit, such as LMS and LMK.

A. *Steady-State Excess Mean Square Error*

Assuming convergence, the steady-state excess MSE can be derived from (28) by making $\lim_{n\to\infty}K(n+1)=\lim_{n\to\infty}K(n)=K(\infty)$. In addition, the following approximations are used to determine the steady-state solution of (28):
1. Only terms of first order in $K(n)$ are considered to determine the terms of order μ in (28). The higher order terms in $K(n)$ are neglected in steady-state since $V(n)$ is already very small.
2. All terms on $K(n)$ which are multiplied by μ^2 in (28) are neglected as $V(n)$ is very small in steady-state and μ is also assumed to be small.

With these aproximations, (28) leads to

$$\lim_{n\to\infty}\{K(n+1)-K(n)\}$$
$$\approx -\mu(6+3\beta+3\beta^2)\sigma_z^2(K(n)R+RK(n))$$
$$+\,\mu^2\Big(4E[z^6(n)] \quad (30)$$
$$+\,(12\beta+21\beta^2+12\beta^4)\sigma_z^2E[z^4(n)]$$
$$+\,18\beta^3\sigma_z^6\Big)R=0$$

Equation (30) is a Lyapunov equation, whose solution is given by:

$$K(\infty)=\mu\frac{4E[z^6(n)]+\alpha\sigma_z^2E[z^4(n)]+18\beta^3\sigma_z^6}{2(6+3\beta+3\beta^2)\sigma_z^2}I \quad (31)$$

where $\alpha=(12\beta+21\beta^2+12\beta^4)$ and I is the identity matrix. Equation (31) implies that the steady-state weight fluctuations are uncorrelated with each other.

From (29) and (31), the steady-state MSE in excess is given by

$$\begin{aligned}\xi_{LMKex}(\infty) &= tr(RK(\infty)) \\ &= \mu\frac{4E[z^6(n)] + \alpha\sigma_z^2 E[z^4(n)] + 18\beta^3\sigma_z^6}{2(6+3\beta+3\beta^2)\sigma_z^2}tr(R)\end{aligned} \quad (32)$$

V. EXAMPLES

This section presents simulation results to illustrate the accuracy of the analytical model and to compare the behaviors of the LMK and LMS adaptive algorithms in a system identification setup.

Example 1. Consider the system in Fig. 1 with 30 weights, W^0 = [-0.0043; -0.0768; -0.3342; 0.2108; 0.1917; 0.3493; -0.1202; -0.0359; -0.1508; 0.1639; -0.1283; 0.1881; -0.2670; 0.1306; 0.0630; -0.0963; -0.2902; -0.1029; 0.4487; -0.2310; -0.1305; 0.1917; 0.0631; -0.1296; 0.1303; 0.0259; -0.0879; 0.0147; -0.1041; 0.0066]^T, $W^{0T}W^0 = 1$, $W(0) = 0$, $\beta = 0.5$, and *z(n)* zero-mean Gaussian with $\sigma_z^2 = 0.01$. The input signal was generated by the autoregressive model $x(n) = ax(n-1) + y(n)$, with $y(n)$ white with unit variance and $a = 0.5$. The eigenvalue spread of the correlation matrix R is equal to 8.82. The value used for μ was equal to $0.00025 \approx \frac{\mu_{max}}{4}$, where μ_{max} was determined by simulation. Fig. 2 shows the mean behavior of the weights number three ($w_3^0 = -0.3342$), fifteen ($w_{15}^0 = 0.0630$) and nineteen ($w_{19}^0 = 0.4487$), obtained using (17) and by simulation (50 runs). All other weights have similar behavior. Fig. 3 shows the MSE for the same value of μ, averaged over 50 simulation runs and obtained using (28) and (29). The bottom curve shows the detail of the transient adaptation, averaged over 200 runs. It can be verified that the model prediction matches very closely the algorithm behavior.

Example 2. This example compares the convergence rates of the LMK and LMS algorithms. It also illustrates how the LMK algorithm behavior is affected by noises with different *pdfs*. Consider the system in Fig. 1 with W^0 = [0.1085; 0.2169; 0.3254; 0.4339; 0.5423; 0.4339; 0.3254; 0.2169; 0.1085]^T, $W^{0T}W^0 = 1$, $W(0) = 0$ and $\beta = 0.5$. Fig. 4 shows LMK algorithm's MSE error for two different noise signals, one Gaussian and one sinusoidal. The sinusoidal noise is described by $z(n) = \sqrt{2\sigma_z^2}sin(377n + \phi)$, where ϕ is random and uniformly distributed between $-\pi$ and π. In both cases $\sigma_z^2 = 0.1$. The input signal $x(n)$ is the same used in Example 1. For comparison purposes, the evolution of the MSE for the LMS algorithm with Gaussian noise with $\sigma_z^2 = 0.1$ is also shown. The step sizes were adjusted to maintain the same steady-state excess MSE for the three cases. For the LMK algorithm, $\mu = 0.000180$ was used with the Gaussian noise and $\mu = 0.000586$ was used with the sinusoidal noise. Fig. 4 clearly shows a faster convergence for the LMK algorithm, even with Gaussian noise. Note also that the proposed LMK analytical model is accurate for the two different noise signals. The maximum value obtained for μ was equal to 0.0009 ($z(n)$ Gaussian), determined by simulation. The LMS step size was $\mu = 0.000215$. Only one curve is plotted for the LMS algorithm because its MSE behavior depends on the value of σ_z^2, but is not affected by different noise *pdfs*.

VI. CONCLUSION

This paper presented a new statistical analysis of the LMK (Least Mean Kurtosis) adaptive algorithm for a wide-sense stationary Gaussian reference signal. Recursive nonlinear equations were derived for the mean weight behavior, for the weight error correlation matrix and for the mean square error. Examples presented show that the model accurately describes the algorithm behavior in both the transient phase of adaptation and in steady-state. The model is valid for zero-mean white noise with any even probability density function. It also clearly shows the dependence of the convergence properties on the initial conditions.

Though the analysis was based on assumptions of slow learning and large number of weights, the results show excellent match between theory and simulations even for reasonably large μ. Nevertheless, a small μ is usually preferred in practical applications, where only one realization of the process is available and where the fluctuations about the mean weight values should be as small as possible.

REFERENCES

[1] Morgan, D.R. and Kuo, S.M., *Active Noise Control Systems: Algorithms and DSP Implementations*, first edition, NY:John Wiley and Sons, 1996.

[2] Woodard, S.E. and Nagchaudhuri, A., “Application of Least Mean Square Algorithms to Spacecraft Vibration Compensation”, *The Journal of the Astronautical Sciences*, v. 48, n.1, p.83-90, Jan-Mar, 1998.

[3] Tanrikulu, O. and Constantinides, A.G., ”Least Mean Kurtosis: A Novel Higher-Order Statistics Based Adaptive Filtering Algorithm”, *Electronic Letters*, 30(3): 189-190, Feb., 1994.

[4] Constantinides, A.G. and Pazaitis, D.I., “A Novel Kurtosis Driven Variable Step-Case Adaptive Algorithm”, *IEEE Transactions on Signal Processing*, v. 47, n.3, Mar., 1999.

[5] Papoulis, A., *Probability, Random Variables and Stochastic Processes*, 3. ed., McGraw-Hill, 1991.

[6] Tanrikulu, O., *Adaptive Signalling Processing Algorithms with Accelerated Convergence and Noise Immunity*, Ph. D. Thesis, Department of Electrical Engineering, Imperial College, London, 1995.

[7] Hübscher, P. I. and Bermudez, J. C. M., “Análise Estatística do Comportamento do Algoritmo LMK (Least Mean Kurtosis)” (in Portuguese), in Proc. 19th Brazilian Telecommunications Symposium, Fortaleza, CE, Brazil, 2001.

[8] N. J. Bershad, P. Celka and J. M. Vesin, “ Stochastic Analyses Gradient Adaptive Identification of Nonlinear Systems with Memory for Gaussian Data and Noisy Input and Output Measurements,” *IEEE Transactions on Signal Processing*, v. 47, n. 3, pp. 675-689, Mar., 1999.

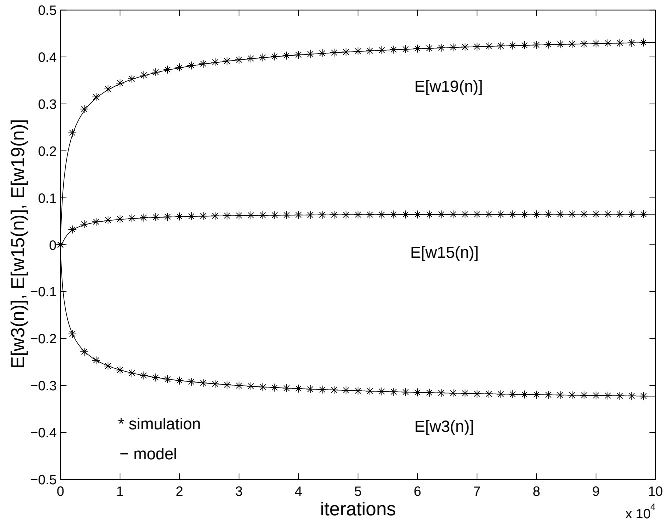

Fig. 2. *Example 1:* Mean Value (model and simulation); $\mu = 0.00025 \approx \frac{\mu_{max}}{4}$; $E[w3(n)]$, $E[w15(n)]$, $E[w19(n)]$.

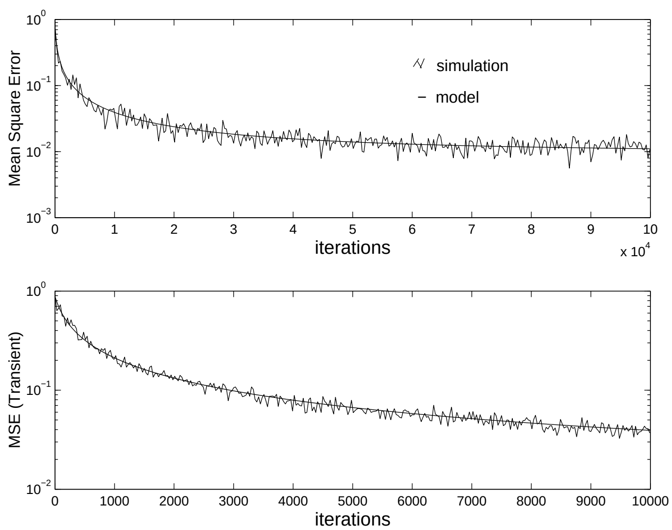

Fig. 3. *Example 1:* MSE (model and simulation); $\mu = 0.00025 \approx \frac{\mu_{max}}{4}$.

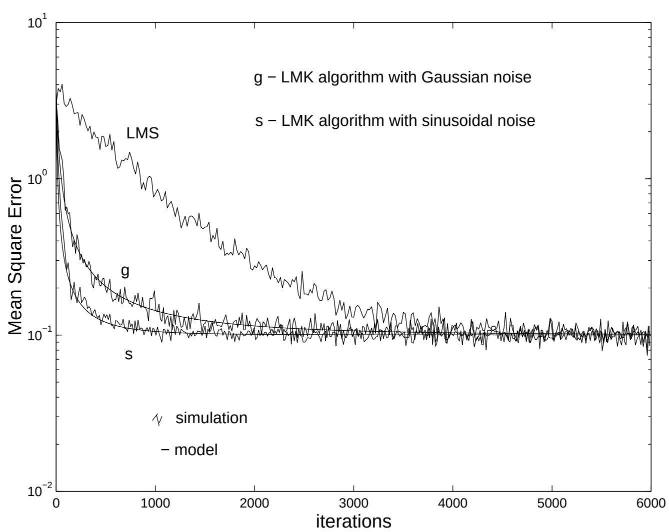

Fig. 4. *Example 2:* LMK algorithm x LMS algorithm.