AN IMPROVED RECURSIVE LEAST SQUARES ALGORITHM ROBUST TO INPUT POWER VARIATION

Charles S. Ludovico

Dept. of Electrical Engineering
State University of Londrina and UNOPAR
Londrina-PR, Brazil
E-mail:ludovico@ieee.org

José C. M. Bermudez

Dept. of Electrical Engineering
Federal University of Santa Catarina
Florianópolis-SC, Brazil
E-mail:j.bermudez@ieee.org

ABSTRACT

This paper proposes a new recursive least-squares adaptive algorithm that improves the steady-state performance of the recently proposed Variable Memory Length (VML) algorithm. Most RLS-type algorithms tend to increase the error in the estimated weight vector during reduced power situations. Like VML, the new algorithm, called Robust VML (RVML), is robust in system identification applications in which the input power is significantly reduced during operation. The RVML algorithm, however, improves the robustness of the VML algorithm when there is significant input power variations during convergence. It should encounter application in systems such as automotive suspension fault detection and adaptive control, and system identification using speech signals. In both cases, considerable periods of power variation during operation are common.

I. INTRODUCTION

Automotive suspension systems often employ model-based fault detection to detect and isolate several faults during operation [1]. System identification is also used for the development of adaptive controllers in the semi-active automotive suspension systems [2]. A major problems in this application is the occurrence of poor excitation [3], where the input is not permanently excited, or non-persistently excited for considerable periods of time. In this case, the lack of persistent excitation is caused by a low-power input signal. Other applications face the same type of challenge, for instance system identification using speech signals, since natural speech tends to have considerable windows of silence or very low power.

Several algorithms have been proposed to improve system response in this application. Examples are the conventional RLS Algorithm with forgetting factor [4], the Self-Tuning Regulator with variable forgetting factor (VFF) [5], the Directional Forgetting algorithm [6], the Restricted Exponential Forgetting algorithm [7], and the Modified Least Squares algorithm incorporating exponential resetting and forgetting [8]. The Directional Forgetting, Restricted Exponential Forgetting and Modified Least Squares algorithms use a directional forgetting factor [10], which is appropriate when the data is spectrally deficient. In [3], it was verified that the VFF algorithm converges and leads to the best results among the above algorithms. Unfortunately, however, the algorithm's estimated weight vector error increases for low-power input signals. Recently, the Variable Memory Length (VML) algorithm was proposed to overcome the low-power input limitations of VFF [9]. The VML algorithm converges and keeps the weight errors small even with low-power input.

Many times, however, the input signal power can become larger after a period of non-persistent excitation, when that algorithm is in steady-state. In these cases, the VML forgetting factor is reduced, and so is algorithm's memory length. This affects the learning behavior of the algorithm. Due to the reduction in memory length, the VML algorithm may not have the ability to further improve is estimate of the system's impulse response if the input signal power (and thus the signal-to-noise ratio) is raised above the level originally used for convergence. This is a desirable feature of the VFF algorithm.

This paper proposes an improvement on the VML algorithm so that it overcomes the above mentioned drawback without compromising performance during low-power input operation (an advantage over VFF). The new algorithm is called Robust VML (RVML), and requires very little increase of computational effort when compared to VML. We first present a brief review of the VML algorithm. Then, the new algorithm is described, and its stochastic behavior behavior is analyzed. It is shown that the mean-square weight deviation becomes dependent on the input power, but differently from VFF. The new algorithm behaves like VFF for large input powers and like VML for low input powers. Simulation results confirm the accuracy of the theoretical model and the performance of the new algorithm.

II. THE VML ALGORITHM

Consider the linear estimation problem where a desired signal $y(n)$ is recursively estimated by a linear FIR adaptive filter with input $x(n)$ and coefficient vector $\mathbf{w}(n) = [w_o(n) \ \ldots \ w_{M-1}(n)]^T$. It is assumed that $y(n)$ and $x(n)$ are related through the linear model

$$y(n) = \mathbf{x}^T(n)\mathbf{w}^{\mathbf{o}} + z(n) \tag{1}$$

where $\mathbf{w}^{\mathbf{o}} = [w_{o_1} \ \ldots \ w_{o_{M-1}}]$ is the optimum solution in the mean-square sense, $\mathbf{x}(n) = [x(n) \ \ldots \ x(n-M+1)]^T$ is the input observation vector and $z(n)$ is a zero-mean white Gaussian noise sequence with variance σ_z^2 and independent of $x(n)$. In system identification, $\mathbf{w}^{\mathbf{o}}$ is the impulse response to be identified.

The weight update equations for the VML algorithm are [9]:

$$e(n) = y(n) - \mathbf{x}^T(n)\mathbf{w}(n-1) \tag{2}$$

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{1 + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)} \tag{3}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)e(n) \tag{4}$$

$$\lambda_o(n) = 1 - \frac{e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)}{[1+\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)]\Sigma_0 M} \tag{5}$$

$$\lambda(n) = \begin{cases} \lambda_o(n) & \text{if } \lambda_o(n) > \lambda_{min} \\ \lambda_{min} & \text{elsewhere} \end{cases} \tag{6}$$

$$\mathbf{P}(n) = \lambda(n)^{-1}\mathbf{P}(n-1) - \lambda(n)^{-1}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1) \tag{7}$$

where $e(n)$ is the estimation error, $\mathbf{k}(n)$ the $M \times 1$ modified Kalman gain vector, $\lambda(n)$ is the variable forgetting factor, $\mathbf{P}(n)$ is the $M \times M$ estimated inverse of the input autocorrelation matrix and Σ_0 is a constant related to the amount of information retained by the adaptive filter at each iteration [5].

The VML algorithm was proposed in [9] to solve the deficiency of the VFF algorithm, which has a limited memory length for low-power input signals. It was shown that the VFF converged mean-square weight deviation is inversely proportional to the input power [9]. On the other hand, the VML algorithm's memory length becomes unlimited as the input power tends to zero. This leads to an improved performance when the input power drops after convergence. This advantage of VML over VFF comes at a price. The steady-state mean-square weight deviation tends to be larger for VML than for VFF for larger signal-to-noise ratios. In the next section, we propose the RVML algorithm which combines the good performance of the VML algorithm for low input powers with the good steady-state performance of VFF for large input powers.

III. THE RVML ALGORITHM

To remove the VML algorithm's deficiency, it is necessary to reduce it's sensitivity to input power variations. Similarly to the technique used in [11] to create the PEDF2 algorithm, we replace the forgetting factor equation (5) with

$$\lambda(n) = 1 - \frac{e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)}{[1+\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)]\Sigma_0 M[1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}]} \tag{8}$$

Eq. (8) defines the Robust Variable Memory Length (RVML) algorithm. Note that the only difference relative to (5) is the factor $[1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}]$ in the denominator. This modification has little effect in the low-power input case, since the added term becomes close to unity for $\mathbf{x}^T(n)\mathbf{x}(n) \ll M$. However, the new factor tends to cancel the effect of the numerator term $\mathbf{x}^T(n)\mathbf{x}(n)/M$ when the input power becomes larger. In this case, the RVML algorithm has the desirable transient and steady-state behaviors of the VFF algorithm.

III-A. Analysis

We now analyze the steady-state behavior of the RVML algorithm. The following conditions are assumed:

a) The system has converged to a near optimum set of parameters during a period of persistent excitation;
b) The input signal $x(n)$ is zero-mean, white and Gaussian, with $M \times M$ autocorrelation matrix $\mathbf{R_{xx}} = \sigma_x^2\mathbf{I}$;
c) The input vector sequence is i.i.d. Thus, $E\{\mathbf{x}(n)\mathbf{x}^T(m)\} = \sigma_x^2\delta(n-m)\mathbf{I}$;
d) The non-persistent input begins only after convergence;
e) $1+\mathbf{x}^T(n)\mathbf{P}(n)\mathbf{x}(n) \approx 1$.[1]

Using (2)–(7), leads to

$$\mathbf{P}(n) = \lambda(n)^{-1}\left[1 - \frac{\mathbf{P}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)}{1+\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)}\right]\mathbf{P}(n-1) \tag{9}$$

Using the matrix inversion lemma, the input autocorrelation matrix can be recursively estimated by

$$\mathbf{P}^{-1}(n) = \lambda(n)\mathbf{P}^{-1}(n-1) + \lambda(n)\mathbf{x}(n)\mathbf{x}^T(n). \tag{10}$$

The solution of (10) gives a closed form expression for $\mathbf{P}^{-1}(n)$:

$$\mathbf{P}^{-1}(n) = \sum_{j=0}^{n}\prod_{k=j}^{n}\lambda(k)\mathbf{x}(j)\mathbf{x}(j)^T + \delta\prod_{k=0}^{n}\lambda(k)\mathbf{I}. \tag{11}$$

A similar derivation leads to a recursive expression for the cross-correlation vector between input and desired signals:

$$\mathbf{c}(n) = \lambda(n)\mathbf{c}(n-1) + \lambda(n)\mathbf{x}(n)y(n). \tag{12}$$

where

$$\mathbf{c}(n) = \sum_{i=0}^{n}\prod_{j=i}^{n}\lambda(i)\mathbf{x}(i)y(i) \tag{13}$$

Using the normal equations [4]

$$\mathbf{P}^{-1}(n)\mathbf{w}(n) = \mathbf{c}(n), \tag{14}$$

the equation for the additive noise

$$z(n) = y(n) - \mathbf{x}^T(n)\mathbf{w}^{\mathbf{o}}, \tag{15}$$

and the equations (10) and (12), it can be shown that

$$\mathbf{P}^{-1}(n)\mathbf{v}(n) = \lambda(n)\mathbf{P}^{-1}(n-1)\mathbf{v}(n-1) + \lambda(n)\mathbf{x}(n)z(n) \tag{16}$$

where $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_o$ is the weight error vector.

Assuming that $\mathbf{P}(n)\mathbf{P}^{-1}(n-1) \simeq \mathbf{I}$ after convergence, and left-multiplying both sides of (16) by $\mathbf{P}(n)$ yields

$$\mathbf{v}(n+1) = \lambda(n)\mathbf{v}(n) + \lambda(n)\mathbf{P}(n)\mathbf{x}(n)z(n). \tag{17}$$

Multiplying (17) by its transpose and taking the expectation,

$$\begin{aligned} E\{\mathbf{v}(n)\mathbf{v}^T(n)\} &= E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &\quad + \sigma_z^2 E\{\lambda^2(n)\mathbf{P}(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{P}(n)\}. \end{aligned} \tag{18}$$

Assuming that in steady state $\mathbf{P}(n) \approx E\{\mathbf{P}(n)\}$ yields the approximation

$$\begin{aligned} E\{\mathbf{v}(n)\mathbf{v}^T(n)\} &= E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &\quad + \sigma_z^2 E\{\mathbf{P}(n)\}E\{\lambda^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\}E\{\mathbf{P}(n)\}. \end{aligned} \tag{19}$$

To simplify (19), we further assume that $E\{e^2(n)\} \approx E\{z^2(n)\} = \sigma_z^2$ in steady-state. Thus,

$$\begin{aligned} E\{\lambda^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\} &\simeq \frac{3\sigma_z^4}{\Sigma_0^2 M^2}E\left\{\frac{[\mathbf{x}^T(n)\mathbf{x}(n)]^2\mathbf{x}(n)\mathbf{x}^T(n)}{\left(1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}\right)^2}\right\} \\ &\quad + E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\} - \frac{2\sigma_z^2}{\Sigma_0 M}E\left\{\frac{\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{x}(n)\mathbf{x}^T(n)}{1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}}\right\} \end{aligned} \tag{20}$$

[1] This approximation is less valid during transient as the order M of the adaptive filter increases.

To proceed with the analysis, we approximate the first and third terms in (20) by the quotient of the expected values of numerator and denominator. Then,

$$E\{\lambda^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\} \simeq \frac{3\sigma_z^4}{\Sigma_0^2 M^2}\frac{E\left\{[\mathbf{x}^T(n)\mathbf{x}(n)]^2\mathbf{x}(n)\mathbf{x}^T(n)\right\}}{E\left\{\left(1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}\right)^2\right\}} + E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\} - \frac{2\sigma_z^2}{\Sigma_0 M}\frac{E\left\{\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{x}(n)\mathbf{x}^T(n)\right\}}{E\left\{1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}\right\}}. \quad (21)$$

Solving the expected values in (21) leads to

$$E\{\lambda^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\} \simeq \left[\frac{12\sigma_z^4\sigma_x^4(M+4)}{\Sigma_0^2 M^2\left(1+2\sigma_x^2+\frac{(M+2)\sigma_x^4}{M}\right)} +1-\frac{2\sigma_z^2\sigma_x^2(M+2)}{\Sigma_0 M(1+\sigma_x^2)}\right]\sigma_x^2\mathbf{I}. \quad (22)$$

Taking the expectations on both sides of (11) and ignoring the initialization effects gives

$$\begin{aligned} E\{\mathbf{P}^{-1}(n)\} &\simeq E\left\{\sum_{j=0}^{n}\prod_{k=j}^{n}\left(1-\frac{e^2(k)\mathbf{x}^T(k)\mathbf{x}(k)}{\Sigma_0 M\left[1+\frac{\mathbf{x}^T(k)\mathbf{x}(k)}{M}\right]}\right) \times\ \mathbf{x}(j)\mathbf{x}^T(j)\right\} \\ &\simeq E\left\{\sum_{j=0}^{n}\prod_{k=j+1}^{n}\left(1-\frac{e^2(k)\mathbf{x}^T(k)\mathbf{x}(k)}{\Sigma_0 M\left[1+\frac{\mathbf{x}^T(k)\mathbf{x}(k)}{M}\right]}\right)\right. \\ &\quad\left.\times\left(1-\frac{e^2(j)\mathbf{x}^T(j)\mathbf{x}(j)}{\Sigma_0 M\left[1+\frac{\mathbf{x}^T(j)\mathbf{x}(j)}{M}\right]}\right)\mathbf{x}(j)\mathbf{x}^T(j)\right\}, \end{aligned} \quad (23)$$

under the conditions considered.

Using assumptions (a)–(c), and neglecting the correlation between $\mathbf{x}^T(n)\mathbf{x}(n)$ and $\mathbf{x}(n)\mathbf{x}^T(n)$,

$$\begin{aligned} E\{\mathbf{P}^{-1}(n)\} &\simeq \sum_{j=0}^{n}\prod_{k=j+1}^{n}E\left\{1-\frac{e^2(k)\mathbf{x}^T(k)\mathbf{x}(k)}{\Sigma_0 M\left[1+\frac{\mathbf{x}^T(k)\mathbf{x}(k)}{M}\right]}\right\} \\ &\quad\times E\left\{\left(1-\frac{e^2(j)\mathbf{x}^T(j)\mathbf{x}(j)}{\Sigma_0 M\left[1+\frac{\mathbf{x}^T(j)\mathbf{x}(j)}{M}\right]}\right)\mathbf{x}(j)\mathbf{x}^T(j)\right\} \\ &\simeq \sum_{j=1}^{n}\left(1-\frac{\sigma_z^2 M\sigma_x^2}{\Sigma_0 M[1+\sigma_x^2]}\right)^{n-j}\left(\sigma_x^2-\frac{\sigma_z^2\sigma_x^4(M+2)}{\Sigma_0 M[1+\sigma_x^2]}\right)\mathbf{I} \\ &\simeq \frac{\Sigma_0 M[1+\sigma_x^2]-\sigma_z^2\sigma_x^2(M+2)}{\sigma_z^2 M}\mathbf{I}. \end{aligned} \quad (24)$$

for n large.

Using the approximation $\mathbf{P}(n) \simeq E\{\mathbf{P}(n)\} \simeq E\{\mathbf{P}^{-1}(n)\}^{-1}$ [13], (24) yields

$$\mathbf{P}(n) \approx \frac{M\sigma_z^2}{\Sigma_0 M[1+\sigma_x^2]-(M+2)\sigma_z^2\sigma_x^2}\mathbf{I}. \quad (25)$$

The term $E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\}$ of equation (19) can finally be written as

$$\begin{aligned} &E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &=E\left\{\left(\frac{[\mathbf{x}^T(n)\mathbf{x}(n)]^2}{\Sigma_0^2 M^2\left(1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}\right)^2}[z^4(n)+(\mathbf{v}^T(n-1)\mathbf{x}(n))^4\right.\right. \\ &\quad -2z^3(n)(\mathbf{v}^T(n-1)\mathbf{x}(n))^3+z^2(n)(\mathbf{v}^T(n-1)\mathbf{x}(n))^2 \\ &\quad -2z^2(n)\mathbf{v}^T(n-1)\mathbf{x}(n)]-\Big[(\mathbf{v}^T(n-1)\mathbf{x}(n))^2\mathbf{x}^T(n)\mathbf{x}(n) \\ &\quad -2z(n)\mathbf{v}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{x}(n)+z^2(n)\mathbf{x}^T(n)\mathbf{x}(n)\Big] \\ &\quad\left.\left.\times\ \frac{2}{\Sigma_0 M\left(1+\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{M}\right)}+1\right)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\right\} \end{aligned} \quad (26)$$

Solving (26) using the same approximation as in (21) yields

$$\begin{aligned} &E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} = E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &\times\left\{1-\frac{2\left(\sigma_z^2\sigma_x^2 M+(M+2)\sigma_x^4 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\right)}{\Sigma_0 M[1+\sigma_x^2]}\right. \\ &+\Big[6(M^2+3M+11)\sigma_x^6\sigma_z^2 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\ &+3(M^2+4M+27)\sigma_x^8 E^2\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\ &\left.+3\sigma_x^4\sigma_z^4 M(M+2)\Big]\ \frac{1}{\Sigma_0^2 M^2\left(1+2\sigma_x^2+\frac{(M+2)\sigma_x^4}{M}\right)}\right\}. \end{aligned} \quad (27)$$

Now, (19) can be written as

$$\begin{aligned} &E\{\mathbf{v}(n)\mathbf{v}^T(n)\} \simeq E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &\times\left\{1-\frac{2\left(\sigma_z^2\sigma_x^2 M+(M+2)\sigma_x^4 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\right)}{\Sigma_0 M[1+\sigma_x^2]}\right. \\ &+\Big[6(M^2+3M+11)\sigma_x^6\sigma_z^2 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\ &+3(M^2+4M+27)\sigma_x^8 E^2\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\ &\left.+3\sigma_x^4\sigma_z^4 M(M+2)\Big]\ \frac{1}{\Sigma_0^2 M^2\left(1+2\sigma_x^2+\frac{(M+2)\sigma_x^4}{M}\right)}\right\} \\ &+\sigma_x^2\mathbf{I}\left(\frac{M\sigma_z^3}{\Sigma_0 M[1+\sigma_x^2]-(M+2)\sigma_z^2\sigma_x^2}\right)^2 \\ &\times\left[1-\frac{2\sigma_z^2\sigma_x^2(M+2)}{\Sigma_0 M(1+\sigma_x^2)}+\frac{12\sigma_z^4\sigma_x^4(M+4)}{\Sigma_0^2 M^2\left(1+2\sigma_x^2+\frac{(M+2)\sigma_x^4}{M}\right)}\right] \end{aligned} \quad (28)$$

Taking the trace on both sides of (28), rearranging the terms and assuming that in steady-state $\lim_{n\to\infty}\mathbf{v}(n) = \lim_{n\to\infty}\mathbf{v}(n-1) = \mathbf{v}(\infty)$, results in a third order equation in $\mathcal{D}(\infty) = E\{\mathbf{v}^T(\infty)\mathbf{v}(\infty)\}$. Neglecting the higher order terms and assuming $\sigma_z^2 << 1$, the mean-square deviation of the RVML algorithm can be approximated by

$$\mathcal{D}(\infty) \simeq \frac{\sigma_z^4 M}{2\Sigma_0(1+\sigma_x^2)} \quad (29)$$

Different from the original VML algorithm, this modified version isn't independent from the input power. But for low-input power ($\sigma_x^2 << 1$) the modified algorithm behaves similar to the original VML, and for unity or bigger than unity input power, the modified version has lower steady-state mean-square weight deviation.

IV. SIMULATIONS

Monte Carlo simulations were performed to compare the performances of the VFF, VML and RVML algorithms. Using a Gaussian white noise as input and as additive noise with $\sigma_z^2 = 10^{-8}$, $\sigma_x^2 = 1$, $M = 50$, $N_0 = 50$ and starting the matrix $\mathbf{P}(n)$ as $\mathbf{P}(0) = 100 \cdot \mathbf{I}$, the three algorithms converged. After the input sample 1000, the input for the three algorithms is attenuated $30dB$, then it returns to unity after the sample 2000. Starting from sample 3000, the input power changes again and is increased 10dB for the three algorithms. Figure 1 shows the mean-square deviation of the weight vector for the three algorithms (average of 500 realizations) and the analytical predictions for the steady state behavior obtained from (29) for the modified VML algorithm. Note that the model accurately predicts the modified algorithms steady-state response. It is also clear that both, VML and modified VML algorithms avoids false identifications, but the modified version has lower mean-square deviation of the weight vector (3dB lower). For input power bigger than unity, the VFF algorithm's mean-square deviation is reduced due the increase of the SNR and has better performance than the VML, but the RVML follows the VFF with a mean-square deviation 0,3dB lower than the VFF.

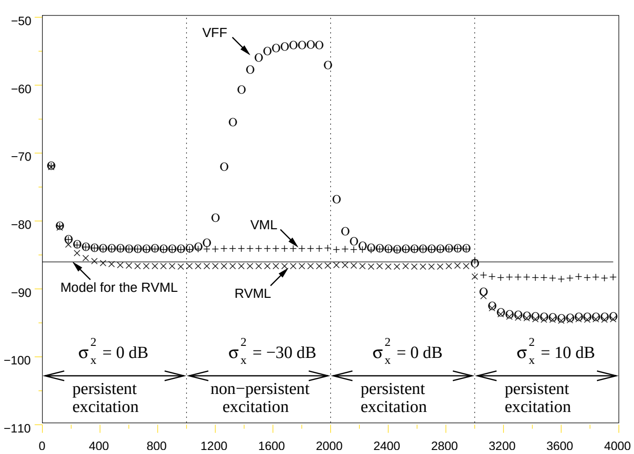

Fig. 1. Monte Carlo simulations: VFF algorithm (o); VML algorithm (+); RVML algorithm (×); analytical model for the RVML algorithm (continuous line)

V. CONCLUSIONS

This paper has proposed a modification on the Variable Memory Length algorithm. The new algorithm is robust in system identification problems in which the input power can vary from very small to very large values. Most RLS-type algorithms tend to increase the error in the estimated weight vector in low input power situations, while the VML algorithm doesn't have a satisfactory performance in large input power situations. The steady-state behavior of the RVML algorithm has been studied. Simulation results have shown the good performance of the new algorithm and the accuracy of the steady-state analysis. The RVML algorithm should encounter applicability in systems such as automotive suspension fault detection and adaptive control, and system identification using speech signals. In these cases, considerable periods of low and high input powers are common during operation.

VI. REFERENCES

[1] ISERMANN, Rolf *Diagnosis Methods for Electronically Controlled Vehicles*. 5th Int. Symposium on Advanced Vehicle Control (AVEC'2000), Ann Arbor, Michigan, USA, August, 2000.

[2] FISCHER, Daniel, ISERMANN, Rolf *Mechatronic Semi-Active and Active Vehicle Suspensions*. Control Engineering Practice, vol. 12, No. 11, p. 1353-1367, November, 2004.

[3] BORNER, Marcus, ZELE, Mina, ISERMANN, Rolf *Comparison of Different Fault Detection Algorithms for Active Body Control Components: Automotive Suspension System* Proc. of the American Control Conference, Arlington, VA, vol. 1 p. 476-481, June 2001.

[4] HAYKIN, Simon. *Adaptive Filter Theory*. 4 ed. New Jersey: Prentice Hall, 2002.

[5] FORTESCUE, T. R., KERSHENBAUM, L. S., YDSTIE, B. E. *Implementation of Self-tuning Regulators with Variable Forgetting Factor*. Automatica, vol. 17 no. 6 p. 831-835, 1981.

[6] KULHAVY, R., KARNY, M. *Tracking of slowly varying parameters by directional forgetting*. 9a. IFAC World Congress, Budapest, p. 79-83, 1984.

[7] KULHAVY, R. *Restricted exponential forgetting in real-time identification*. Automatica, vol. 23 no.5 p. 589-600, 1987.

[8] SALGADO, Mario E., GOODWIN, Graham C., MIDDLETON, Richard H. *Modified least squares algorithm incorporating exponential resetting and forgetting*. Int. J. Control. vol. 47, No. 8, p. 477-491, 1988.

[9] LUDOVICO, Charles S., BERMUDEZ, José C. M. *A Recursive Least Squares Algorithm Robust to Low-Power Excitation*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Quebec, Canada, May, 2004.

[10] ZHIYONG, Zhang, SHIFU, Wang, CHONGZHI, Fang, JINGLI, Kang *Unified framework of adaptive forgetting identification algorithms for time-varying systems*. TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering, vol. 4 p. 424 -427, 1993

[11] BERTIN, Daniele, BITTANTI, Sergio, BOLZERN, Paolo, *Tracking of Stationary Systems by Means of Different Prediction-Error Directional Forgetting Techniques*. 2nd IFAC Workshop on Adaptive Systems in Control ans Signal Processing, Lund, 1986.

[12] MANOLAKIS, D. G., INGLE, V. K., KOGON, S. M. *Statistical and adaptive signal processing*. McGraw Hill, Boston, 1999.

[13] SAYED, A. H. *Fundamentals of Adaptive Filtering*, John-Wiley, 20003.