A Recursive Least Squares Algorithm Robust to Low-Power Excitation

Charles S. Ludovico
Dept. of Electrical Engineering
Federal University of Santa Catarina
Florianópolis-SC, 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, called the Variable Memory Length (VML) algorithm. The new algorithm is robust in system identification problems in which the input power can be significantly reduced during operation. Most RLS-type algorithms tend to increase the error in the estimated weight vector in such situations. The VML algorithm keeps the mean square deviation of the weight unchanged during the absence of signal power. It should encounter application in systems such as automotive suspension fault detection and system identification using speech signals. In both cases, considerable periods of low input power during operation are common.**

I. Introduction

Nowdays, electronics play an essential role in the automotive industry, whether in the development of products or in on-board equipment. Fault detection systems is a important tool to increase reliability and safety, and therefore essential for today's needs.

For automotive suspension systems, model-based fault detection using system identification is able of detect and isolate several faults during operation [1]. One of the major problems in this application is the occurrence of poor excitation [2], 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.

Another application that faces the same type of challenge is 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 [3], the Self-Tuning Regulator with variable forgetting factor (VFF) [4], the Directional Forgetting algorithm [5], the Restricted Exponential Forgetting algorithm [6] and the Modified Least Squares algorithm incorporating exponential resetting and forgetting [7]. The Directional Forgetting, Restricted Exponential Forgetting and Modified Least Squares algorithms use a directional forgetting factor [8], which is appropriate when the data is spectrally deficient. In [2], it was verified that the VFF algorithm converges and has the best results among the other algorithms. However, when the problem is the low power of input signal, the mean-square deviation of the estimated weight vector increases. In the fault detections application, this characteristic of the algorithm can compromise the performance and reliability of the system.

In this paper, a novel algorithm is proposed which overcomes the problem of the VFF algorithm when the input is of very low power without compromising performance in normal operation and with very little increase of computational effort. Section II presents a brief review of the derivation of the VFF algorithm. Section III presents an analysis of the VFF steady-state behavior when the input power is severely reduced. It is shown that the mean square deviation becomes inversely proportional to the input power. The new algorithm is proposed and analyzed in Section IV. Simulation results are presented in Section V, which confirm the accuracy of the theorical models and the superior performance of the new algorithm under the condition of reduced input power.

II. Self-Tuning Regulator with variable forgetting factor (VFF)

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 $\mathbf{w}(n) = [w_o(n)\ \ldots\ w_{M-1}(n)]^T$. The desired signal and $x(n)$ are assumed 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 VFF algorithm proposed in [4] given by the following set of equations:

$$\begin{aligned}
e(n) &= y(n) - \mathbf{x}^T(n)\mathbf{w}(n) && (2)\\
\mathbf{k}(n) &= \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{1+\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)} && (3)\\
\mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n)e(n) && (4)\\
\lambda_0(n) &= 1 - \frac{[1-\mathbf{x}^T(n)\mathbf{k}(n)]e^2(n)}{\Sigma_0} && (5)\\
\lambda(n) &= \begin{cases} \lambda_0(n) & \text{if } \lambda(n)_0 > \lambda_{min} \\ \lambda_{min} & \text{elsewhere} \end{cases} \\
\mathbf{P}(n) &= \lambda^{-1}(n)\mathbf{P}(n-1) - \lambda^{-1}(n)\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1) && (6)
\end{aligned}$$

where $e(n)$ is the *a priori* 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 correlation matrix and Σ_0 is a constant related to the amount of information retained by the adaptive filter at each iteration.

Compared to the conventional RLS algorithm [10], the VFF algorithm introduces a variable forgetting factor $\lambda(n)$ and a modified Kalman gain vector, which is not explicitly dependent on $\lambda(n)$. This modification was introduced to reduce computational complexity [4]. Though the simpler algorithm behaves similarly to RLS in most cases, it requires a test on $\lambda(n)$ to avoid very small or negative values.

The expression for $\lambda(n)$ was derived to keep the amount of information in the filter constant over time. The measure of information $\Sigma(n)$ is based on the weighted sum of the *a posteriori* errors and is estimated by the recursive equation [9]

$$\Sigma(n) = \lambda(n)\Sigma(n-1) + [1-\mathbf{x}^T(n)\mathbf{k}(n)]e^2(n) \tag{7}$$

Making $\Sigma(n) = \Sigma(n-1) = \Sigma_0$ in (7) yields

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

where

$$N(n) = \frac{\Sigma_0}{[1 - \mathbf{x}^T(n)\mathbf{k}(n)]e^2(n)} = \frac{1}{1 - \lambda(n)} \tag{9}$$

would be the asymptotic filter memory length [10] if $\lambda = \lambda(n)$ would be used throughout the estimation. One possible choice for Σ_0 is [4]

$$\Sigma_0 = \sigma_z^2 N_0 \tag{10}$$

which leads to $\lim_{n\to\infty} E\{N(n)\} = N_0$.

III. The Low Power Input Case

The main concern in this work is the behavior of the adaptive algorithm in the particular case of non-persistent excitation that occurs when the input signal power is very much reduced during the system operation. The following analysis of the VFF algorithm behavior is done under the following conditions:

a) The system is assumed to have converged to a near optimum set of parameters during a period of persistent excitation;
b) The non-persistent input begins only after the system has been converged;
c) It is assumed that $1 + \mathbf{x}^T(n)\mathbf{P}(n)\mathbf{x}(n) \simeq 1$ when $x(n)$ is non-persistent;
d) Since $\lambda(n)$ is very close to unity after convergence, the behavior of $\mathbf{k}(n)$ can be approximated by the behavior of the true Kalman gain [3]. Thus, except for the variable $\lambda(n)$ the algorithm equations coincide with those of the conventional RLS algorithm.

Assuming steady-state, $\mathbf{x}^T(n)\mathbf{k}(n) \ll 1$ (low input power) and using Σ_0 chosen as in (10), (9) leads to

$$N_{max} = N_0. \tag{11}$$

This limitation in the maximum filter memory length degrades the algorithm's performance for low-power inputs, since the filter memory should always increase ($\lambda(n)$ approaching 1) as $x(n)$ gets smaller. The consequences of the memory limitation when the input power drops is now analyzed in more detail.

Considering the estimate of the input autocorrelation matrix

$$\hat{\mathbf{R}}(n) = \sum_{j=0}^{n} \lambda^{n-j}(j)\mathbf{x}(j)\mathbf{x}^T(j) \tag{12}$$

and defining the weight error vector as

$$\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_o \tag{13}$$

it can be shown that [10]

$$\hat{\mathbf{R}}(n)\mathbf{v}(n) = \lambda(n)\hat{\mathbf{R}}(n-1)\mathbf{v}(n-1) + \mathbf{x}(n)z(n) \tag{14}$$

which pre-multiplied by $\hat{\mathbf{R}}^{-1}(n)$ yields

$$\mathbf{v}(n) = \lambda(n)\hat{\mathbf{R}}^{-1}(n)\hat{\mathbf{R}}(n-1)\mathbf{v}(n-1) + \hat{\mathbf{R}}^{-1}(n)\mathbf{x}(n)z(n). \tag{15}$$

After convergence, $\hat{\mathbf{R}}^{-1}(n)\hat{\mathbf{R}}(n-1) \simeq \mathbf{I}$, and (15) becomes

$$\mathbf{v}(n) \simeq \lambda(n)\mathbf{v}(n-1) + \hat{\mathbf{R}}^{-1}(n)\mathbf{x}(n)z(n). \tag{16}$$

Post-multiplying (16) by $\mathbf{v}^T(n)$ and taking the expected value,

$$\begin{aligned} E\{\mathbf{v}(n)\mathbf{v}^T(n)\} \simeq & E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &+ E\{\lambda(n)z(n)\mathbf{v}(n-1)\mathbf{x}^T(n)\hat{\mathbf{R}}^{-1}(n)\} \\ &+ E\{\lambda(n)z(n)\hat{\mathbf{R}}^{-1}(n)\mathbf{x}(n)\mathbf{v}^T(n-1)\} \\ &+ E\{z^2(n)\hat{\mathbf{R}}^{-1}(n)\mathbf{x}(n)\mathbf{x}^T(n)\hat{\mathbf{R}}^{-1}(n)\}. \end{aligned} \tag{17}$$

Since $z(n)$ is zero-mean and independent, (17) reduces to

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

After convergence, it is reasonable to assume that $\lambda(n)$ is weakly correlated with $\mathbf{v}(n)$. Thus,

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

Using the approximation $E\{e^2(n)\} \simeq \sigma_z^2$ in steady-state and assumption (c) for low-power input yields, after calculations

$$E\{\lambda^2(n)\} \simeq \left(1 - \frac{2\sigma_z^2}{\Sigma_0} + \frac{3\sigma_z^4}{\Sigma_0^2}\right). \tag{20}$$

Under the same conditions, (12) can be approximated as

$$\begin{aligned} E\{\hat{\mathbf{R}}(n)\} =& E\{\sum_{j=0}^{n} \lambda^{n-j}(j)\mathbf{x}(j)\mathbf{x}^T(j)\} \\ =& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)}{(1 + \mathbf{x}^T(j)\mathbf{P}(j)\mathbf{x}(j))\,\Sigma_0}\right)^{n-j} \right. \\ & \left. \times\ \mathbf{x}(j)\mathbf{x}^T(j)\right\} \\ \simeq& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)}{\Sigma_0}\right)^{n-j}\mathbf{x}(j)\mathbf{x}^T(j)\right\} \\ \simeq& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)}{\Sigma_0}\right)^{n-j}\right\}\mathbf{R_x} \end{aligned} \tag{21}$$

where $\mathbf{R_x}$ is the correlation matrix of the input vector. Also, noting that $E\left\{\sum_{j=0}^{n}\left(1 - e^2(j)/\Sigma_0\right)^{n-j}\right\} \simeq \sum_{j=0}^{n}\left(1 - \sigma_z^2/\Sigma_0\right)^{n-j}$ under the conditions considered,

$$\begin{aligned} E\{\hat{\mathbf{R}}(n)\} \simeq& \sum_{j=0}^{n}\left(1 - \frac{\sigma_z^2}{\Sigma_0}\right)^{n-j}\mathbf{R_x} \\ =& \frac{\Sigma_0 \mathbf{R_x}}{\sigma_z^2} \qquad \text{for large } n. \end{aligned} \tag{22}$$

Using the approximation $\hat{\mathbf{R}}^{-1}(n) \simeq E\{\hat{\mathbf{R}}^{-1}(n)\} \simeq E\{\hat{\mathbf{R}}(n)\}^{-1}$ [11], (22) yields

$$\hat{\mathbf{R}}^{-1}(n) \simeq \frac{\sigma_z^2}{\Sigma_0}\mathbf{R_x}^{-1}. \tag{23}$$

Then, (19) can be written as

$$\begin{aligned} E\{\mathbf{v}(n)\mathbf{v}^T(n)\} \simeq& \left(1 - \frac{2\sigma_z^2}{\Sigma_0} + \frac{3\sigma_z^4}{\Sigma_0^2}\right)E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\ &+ \frac{\sigma_z^6}{\Sigma_0^2}\mathbf{R_x}^{-1}. \end{aligned} \tag{24}$$

In steady-state, $\lim_{n\to\infty}\mathbf{v}(n) = \lim_{n\to\infty}\mathbf{v}(n-1) = \mathbf{v}(\infty)$. Then,

$$E\{\mathbf{v}(\infty)\mathbf{v}^T(\infty)\} \simeq \frac{\sigma_z^4}{2\Sigma_0 - 3\sigma_z^2}\mathbf{R_x}^{-1} \tag{25}$$

and the steady-state mean-square weight vector deviation for the VFF algorithm can be approximated by

$$\mathcal{D}(\infty) = tr[E\{\mathbf{v}(\infty)\mathbf{v}^T(\infty)\}] \simeq \frac{\sigma_z^4}{2\Sigma_0 - 3\sigma_z^2}\mathrm{tr}[\mathbf{R_x}^{-1}] \tag{26}$$

where tr[$\cdot$] is the trace of the matrix.

It is clear from (26) that $\mathcal{D}(\infty)$ increases if the input power is reduced. For $x(n)$ white with $\sigma_x^2\mathbf{R}_{\mathbf{x}} = \mathbf{I}$,

$$\mathcal{D}(\infty) \simeq \frac{\sigma_z^4 M}{(2\Sigma_0 - 3\sigma_z^2)\sigma_x^2} \tag{27}$$

which explicitly shows the effect of a reduced input power.

IV. The New Algorithm Proposed

It has been verified that the VFF algorithm establishes an upper limit on the filter memory when the input power becomes very low. When this memory is filled with small amplitude input samples, the mean square deviation $\mathcal{D}$ increases. Although $\mathcal{D}$ remains bounded, the increasing error in the adaptive weights makes the algorithm unreliable for system identification. A solution to this problem is to make the filter memory dependent on the input power.

Consider replacing N_o in (10) with

$$N_m(n) = \frac{N_0 M}{\mathbf{x}^T(n)\mathbf{x}(n)}. \tag{28}$$

Using (28) makes the memory length inversely proportional to the input power and unbounded as $\sigma_x^2 \to 0$. The amount of information becomes

$$\Sigma_m(n) = \sigma_z^2 N_m(n) = \sigma_z^2 \frac{N_0 M}{\mathbf{x}^T(n)\mathbf{x}(n)} = \frac{\Sigma_0 M}{\mathbf{x}^T(n)\mathbf{x}(n)} \tag{29}$$

and the new forgetting factor is given by

$$\lambda(n) = 1 - \frac{[1 - \mathbf{x}^T(n)\mathbf{k}(n)]e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)}{\Sigma_0 M}. \tag{30}$$

which defines the new algorithm, termed Variable Memory Length (VML) algorithm.

To study of the VML algorithm behavior, the same considerations assumed for the VFF algorithm are used. However, $\lambda(n)$ and $\mathbf{v}(n)$ can not be assumed uncorrelated because of the extra term $\mathbf{x}^T(n)\mathbf{x}(n)$ in the numerator of $\lambda(n)$. Also, to make the mathematics manageable, $x(n)$ is assumed zero-mean white and Gaussian. The term $E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\}$ in (18) can be expanded as

$$\begin{aligned}
&E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} = \\
&E\left\{\left(1 - \frac{e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)}{(1+\mathbf{x}^T(n)\mathbf{P}(n)\mathbf{x}(n))\,\Sigma_0 M}\right)^2 \mathbf{v}(n-1)\mathbf{v}^T(n-1)\right\} \\
&= E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&- \frac{2}{\Sigma_0 M}\Big\{E\{z^2(n)\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&- 2E\{z(n)\mathbf{v}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&+ E\{[\mathbf{v}^T(n-1)\mathbf{x}(n)]^2\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\}\Big\} \\
&+ \frac{1}{\Sigma_0^2 M^2}\Big\{E\{z^4(n)[\mathbf{x}^T(n)\mathbf{x}(n)]^2\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&- 2E\{z^3(n)\mathbf{v}^T(n-1)\mathbf{x}(n)[\mathbf{x}^T(n)\mathbf{x}(n)]^2\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&+ E\{z^2(n)[\mathbf{v}^T(n-1)\mathbf{x}(n)]^2[\mathbf{x}^T(n)\mathbf{x}(n)]^2 \\
&\times \mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&- 2E\{z^2(n)[\mathbf{v}^T(n-1)\mathbf{x}(n)]^3[\mathbf{x}^T(n)\mathbf{x}(n)]^2 \\
&\times \mathbf{v}(n-1)\mathbf{v}^T(n-1)\} \\
&+ E\{[\mathbf{v}^T(n-1)\mathbf{x}(n)]^4[\mathbf{x}^T(n)\mathbf{x}(n)]^2\mathbf{v}(n-1)\mathbf{v}^T(n-1)\}\Big\}.
\end{aligned} \tag{31}$$

Considering the properties of $z(n)$ and $x(n)$ and neglecting the fourth order moments of $\mathbf{v}(n-1)$, (31) becomes

$$\begin{aligned}
&E\{\lambda^2(n)\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} = \\
&\Big\{1 - \frac{2}{\Sigma_0 M}\left[\sigma_z^2\sigma_x^2 M + (M+2)\sigma_x^4 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\right] \\
&+ \frac{1}{\Sigma_0^2 M^2}\Big[\sigma_x^4(M+2)M3\sigma_z^4 + (M^2+3M+11)\sigma_x^6\sigma_z^2 \\
&\times E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} + 3(M^2+4M+27)\sigma_x^8 \\
&\times E^2\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\Big]\Big\}E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\}.
\end{aligned} \tag{32}$$

Similar to the calculations in (21),

$$\begin{aligned}
E\{\hat{\mathbf{R}}(n)\} =& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)\mathbf{x}^T(n)\mathbf{x}(n)}{(1+\mathbf{x}^T(j)\mathbf{P}(j)\mathbf{x}(j))\,\Sigma_0 M}\right)^{n-j} \times \mathbf{x}(n)\mathbf{x}^T(n)\right\} \\
\simeq& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)\mathbf{x}^T(n)\mathbf{x}(n)}{\Sigma_0 M}\right)^{n-j}\mathbf{x}(n)\mathbf{x}^T(n)\right\} \\
\simeq& E\left\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)\mathbf{x}^T(n)\mathbf{x}(n)}{\Sigma_0 M}\right)^{n-j}\right\}\mathbf{R}_{\mathbf{x}}
\end{aligned} \tag{33}$$

Making the approximation (verified through simulation)

$$E\{\sum_{j=0}^{n}\left(1 - \frac{e^2(j)\mathbf{x}^T(n)\mathbf{x}(n)}{(\Sigma_0 M)}\right)^{n-j}\} \simeq \sum_{j=0}^{n}\left(1 - \sigma_z^2\sigma_x^2/(\Sigma_0)\right)^{n-j} \tag{34}$$

yields

$$\begin{aligned}
E\{\hat{\mathbf{R}}(n)\} \simeq& \sum_{j=0}^{n}\left(1 - \frac{\sigma_z^2\sigma_x^2}{\Sigma_0}\right)^{n-j}\mathbf{R}_{\mathbf{x}} \\
\simeq& \frac{\Sigma_0\mathbf{R}_{\mathbf{x}}}{\sigma_z^2\sigma_x^2}.
\end{aligned} \tag{35}$$

Also, the following approximation is used (as done in (22)) [11]

$$\hat{\mathbf{R}}^{-1}(n) \simeq E\{\hat{\mathbf{R}}^{-1}(n)\} \simeq E\{\hat{\mathbf{R}}(n)\}^{-1} \simeq \frac{\sigma_z^2\sigma_x^2}{\Sigma_0}\mathbf{R}_{\mathbf{x}}{}^{-1}. \tag{36}$$

Now, (18) can be written as

$$\begin{aligned}
E\{\mathbf{v}(n)\mathbf{v}^T(n)\} =& \Big[1 - \frac{2}{\Sigma_0 M}\Big(\sigma_z^2\sigma_x^2 M + (M+2)\sigma_x^4 \\
&\times E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\Big) + \frac{1}{\Sigma_0^2 M^2}\Big(3\sigma_z^4\sigma_x^4 M(M+2) \\
&+ (M^2+3M+11)\sigma_x^6\sigma_z^2 E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\
&+ 3(M^2+4M+27)\sigma_x^8 E^2\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\}\Big)\Big] \\
&\times E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} + \frac{\sigma_z^6\sigma_x^4}{\Sigma_0^2}\mathbf{R}_{\mathbf{x}}{}^{-1}.
\end{aligned} \tag{37}$$

Taking the trace of both sides of (37) and rearranging terms yields

$$
\begin{aligned}
&E\{\mathbf{v}^T(n)\mathbf{v}(n)\} = \\
&\left[\frac{3(M^2+4M+27)\sigma_x^8}{\Sigma_0^2 M^2}\right] E^3\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\
&+\left[\frac{(M^2+3M+11)\sigma_x^6\sigma_z^2}{\Sigma_0^2 M^2} - \frac{2(M+2)\sigma_x^4}{\Sigma_0 M}\right] \\
&\times E^2\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\
&+\left[1+\frac{3\sigma_x^4\sigma_z^4(M+2)}{\Sigma_0^2 M} - \frac{2\sigma_z^2\sigma_x^2}{\Sigma_0}\right] E\{\mathbf{v}^T(n-1)\mathbf{v}(n-1)\} \\
&+\frac{\sigma_z^6\sigma_x^2 M}{\Sigma_0^2}.
\end{aligned}
\tag{38}
$$

Making $\lim\limits_{n\to\infty}\mathbf{v}(n) = \lim\limits_{n\to\infty}\mathbf{v}(n-1) = \mathbf{v}(\infty)$ in (38) yields

$$
\begin{aligned}
&\left[\frac{3(M^2+4M+27)\sigma_x^8}{\Sigma_0^2 M^2}\right]\mathcal{D}^3(\infty) \\
&+\left[\frac{(M^2+3M+11)\sigma_x^6\sigma_z^2}{\Sigma_0^2 M^2} - \frac{2(M+2)\sigma_x^4}{\Sigma_0 M}\right]\mathcal{D}^2(\infty) \\
&+\left[\frac{3\sigma_x^4\sigma_z^4(M+2)}{\Sigma_0^2 M} - \frac{2\sigma_z^2\sigma_x^2}{\Sigma_0}\right]\mathcal{D}(\infty) \\
&+\frac{\sigma_z^6\sigma_x^2 M}{\Sigma_0^2} = 0
\end{aligned}
\tag{39}
$$

For $\sigma_x^2 \ll 1$ the term in σ_x^2 dominates and (39) can be approximated by

$$
\mathcal{D}(\infty) \simeq \frac{\sigma_z^4 M}{2\Sigma_0} \tag{40}
$$

Note that (40) is independent of σ_x^2, indicating that the mean square deviation will not be affected when the input power drops after convergence. Thus, a false system response estimate is avoided.

V. Simulations

Monte Carlo simulations were performed to compare the performances of the VFF and VML algorithms. Using a Gaussian white noise as input and as additive noise with $\sigma_z^2 = 10^{-8}$, $\sigma_x^2 = 1$, $M = 10$, $N_0 = 50$ and starting the matrix $\mathbf{P}(n)$ as $\mathbf{P}(0) = 100 \cdot \mathbf{I}$, both algorithm converged. After the input sample 1000, the input for both algorithms is attenuated $30dB$. Figure 1 shows the mean-square deviation of the weight vector for both algorithms (average of 200 realizations) and the analytical predictions for the steady state behavior obtained from (27) and (40). Note that both models accurately predict the algorithms steady-state response after the reduction in input power. It is also clear the the VML algorithm avoids false identifications.

VI. Conclusions

This paper has proposed a new recursive least squares adaptive algorithm, called the Variable Memory Length (VML) algorithm. The new algorithm is robust in system identification problems in which the input power can be significantly reduced during operation. Most RLS-type algorithms tend to increase the error in the estimated weight vector in such situations. The VML algorithm keeps the mean square deviation of the weight unchanged during the absence of signal power. It should encounter application in systems such as automotive suspension fault detection and system identification using speech signals. In both cases, considerable periods of low input power during operation are common.

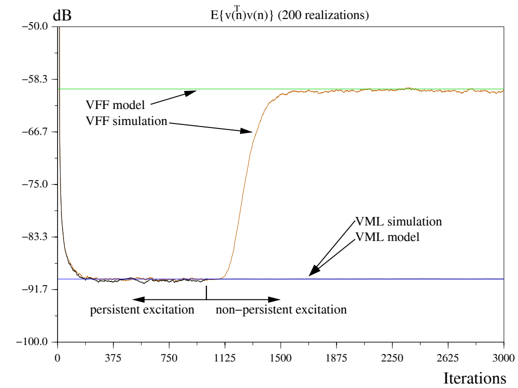

Fig. 1. Monte Carlo simulations: VFF algorithm (brown); modified VFF algorithm (black). Analytical models: VFF (green); modified VFF (blue).

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] 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.

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

[4] 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.

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

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

[7] 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.

[8] 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

[9] ALBERT, A., SITTLER, R. W. *A method for computing least squares estimates that keep up with the data*. SIAM J. Control, 3, 384.

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

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