Multiple Plant Identifier via Adaptive LMS Convex Combination

Jerónimo Arenas-García, Manel Martínez-Ramón, Vanessa Gómez-Verdejo and Aníbal R. Figueiras-Vidal
Department of Signal Theory and Communications,
Universidad Carlos III de Madrid, 28911 Leganés-Madrid, Spain
Email: jarenas@tsc.uc3m.es

Abstract* – The Least Mean Square (LMS) algorithm has become a very popular algorithm for adaptive filtering due to its robustness and simplicity. A difficulty concerning LMS filters is their inherent compromise between tracking capabilities and precision, that is imposed by the selection of a fixed value for the adaption step. An adaptive convex combination of one fast LMS filter (high adaption step) and one slow LMS filter (low adaption step) was proposed as a way to break this balance. In this paper, we propose to generalize this idea, combining multiple LMS filters with different adaption steps. Additional speeding up procedures are necessary to improve the performance of the basic scheme. Some simulation work has been carried out to show the appropriateness of this approach when identifying plants that vary at different rates.

***Keywords* – Least Mean Square (LMS), plant identification, convex combination**

I. INTRODUCTION

Widrow and Hoff's Least Mean Square (LMS) [1] algorithm has become the most popular algorithm for adaptive filtering due to its robustness, good tracking capabilities, and simplicity in terms of computational load and easiness of implementation. It has therefore been successfully used in a wide variety of applications such as plant identification among others [2].

There are two main difficulties concerning LMS filters. The first arises in scenarios with a high eigenvalue spread in the correlation matrix of the input process. The second difficulty is the inherent balance between speed of convergence and final misadjustment in stationary situations that is imposed by the selection of a certain value for the adaption step [2]. In this paper, we will pay attention to this second issue.

Some previous proposals try to improve the speed vs precision balance by using non-quadratic error functions (e.g. the Least Mean Fourth (LMF) algorithm [3]) that get a faster convergence. The idea of combining the LMS and LMF cost functions has also been explored in the literature [4], [5].

This work has been partly supported by CICYT grant TIC2002-03713.

Following a different way, many researchers have proposed schemes that manage the value of the learning rate. In [6] we studied several of such approaches, being the algorithm in [7] the one which reported a best behavior. Harris el al. propose to adaptively change the value of the adaption step by multiplying or dividing it by a constant factor, depending on the signs of each of the components of the gradient vector along some of the last iterations. This algorithm has the drawback of introducing some extra parameters which must be fixed to a priori values. This selection itself implies a compromise between stability, speed of convergence, precision and tracking capabilities. Therefore, the optimal values for these parameters are highly dependent on the particular environment in which the filter operates.

In [8] we proposed an alternative approach that had its roots in the idea of model mixtures of [9], [10]. In these papers (which consider the prediction of autoregressive processes) it is proposed to combine adaptive filters with all possible orders up to m, to obtain a universal predictor that is, in the mean square error sense, at least as good as the best individual filter. Our scheme, however, is aimed at improving the speed of convergence vs precision balance of LMS filters. In [6] we combined one fast and one slow LMS filters with the objective of getting the advantages of both of them: fast convergence and good tracking capabilities from the fast LMS filter, and reduced steady-state error from the slow filter. In order to obtain a better behavior from the combination, it was necessary to design procedures that speed up the convergence of the slow filter and improve the switching between the fast and slow filters.

In this paper, we introduce a new algorithm that combines multiple LMS filters (M-CLMS algorithm) which is the natural extension of the combination of two LMS filters (CLMS) of [6]. Although the component filters in the scheme can be of any type, we will just consider the case of LMS filters with different values for the adaption steps. This way, each filter will be particularly good in tracking changes that occur in the plant at a certain rate, and the combination of all the filters will be good at tracking any type of changes. Besides, we will also design some procedures that improve the performance of the basic M-CLMS algorithm.

The organization of the paper is as follow: in the next section we summarize the CLMS algorithm, which has already been presented in [6]. The extension of these ideas to the combination of multiple filters (M-CLMS algorithm) plus some procedures to improve the performance of the basic algorithm are presented in Section III. Section IV is devoted to some computer experiments that show the advantages of our proposal. Finally, we present some conclusions and enlighten some lines for future research.

II. THE CLMS ALGORITHM

In [8], [6] we proposed to use, for plant identification, an adaptive combination of two adaptive filters, the first being a fast LMS filter (i.e. with a high adaption step μ_1) and the second a slow one (low adaption step μ_2). In principle, both algorithms operate completely decoupled from each other, using standard LMS adaption rules:

$$\mathbf{w}_i[k+1] = \mathbf{w}_i[k] + \mu_i e_i[k]\mathbf{x}[k]; \;\; i = 1, 2 \tag{1}$$

where $\mathbf{x}[k]$ is the filter input at instant k, and $e_i[k]$ is the error incurred by each filter, i.e., $e_i[k] = d[k] - \mathbf{w}_i^T[k]\mathbf{x}[k]$, $d[k]$ being the desired output.

The CLMS filter uses a convex combination of the weights of the two LMS filters

$$\mathbf{w}_{eq}[k] = \lambda[k]\mathbf{w}_1[k] + (1-\lambda[k])\mathbf{w}_2[k] \tag{2}$$

where parameter $\lambda[k]$ is kept in the interval $[0, 1]$ by defining it as $\lambda[k] = sgm\,(a[k]) = 1/(1 + e^{-a[k]})$. The combination parameter is adapted to minimize the error of the overall adaptive filter, also using LMS adaption rule:

$$\begin{aligned} a[k+1] &= a[k] - \frac{\mu_a}{2}\frac{\partial e_{eq}^2[k]}{\partial a[k]} \\ &= a[k] + \mu_a\left(d[k] - \mathbf{w}_{eq}^T[k]\mathbf{x}[k]\right)\left(\lambda[k]\,(1-\lambda[k])\right) \\ &\quad \left(\mathbf{w}_1[k] - \mathbf{w}_2[k]\right)^T \mathbf{x}[k]. \end{aligned} \tag{3}$$

In the above equation, μ_a must be fixed to a value much higher than μ_1, so the combination is adapted even faster than the fastest of the LMS filters. Finally, we limit the values of $a[k]$ to the interval $[-4, 4]$, to prevent the algorithm from stopping because either $\lambda[k]$ or $1 - \lambda[k]$ is too close to 0.

This scheme has a very intuitive interpretation: in situations where a high speed would be desirable, the fast LMS will outperform the slow one, making $\lambda[k]$ evolve towards 1, and $\mathbf{w}_{eq}[k] \approx \mathbf{w}_1[k]$. However, in stationary intervals, it is the slow filter which operates best, making $\lambda[k]$ get close to 0, and $\mathbf{w}_{eq}[k] \approx \mathbf{w}_2[k]$.

It is possible to further improve the performance of the basic combination algorithm by using the good convergence properties of the fast filter to speed up the convergence of the slow LMS filter. We do this by step-by-step transferring a part of weight vector $\mathbf{w}_1$ to $\mathbf{w}_2$. So, in this case, the adaption rule for the slow filter becomes

$$\mathbf{w}_2[k+1] = \alpha\left(\mathbf{w}_2[k] + \mu_2 e_2[k]\mathbf{x}[k]\right) + (1-\alpha)\mathbf{w}_1[k+1] \tag{4}$$

The weight transfer must only be applied when the fast LMS is performing significantly better than the slow one.

A second modification serves to reduce the pernicious effect of factor $\mathbf{w}_1[k] - \mathbf{w}_2[k]$ in equation (3), when both weights are similar. To alleviate this problem we can include a momentum procedure for the adaption of parameter $a[k]$:

$$a[k+1] = a[k] - \frac{\mu_a}{2}\frac{\partial e^2[k]}{\partial a[k]} + \rho(a[k] - a[k-1]). \tag{5}$$

Although CLMS algorithm requires the introduction of some extra parameters, we showed in [6] that their selection is not critical, and the optimal values are not very dependent on the concrete scenario in which the filter is being used.

III. COMBINING SEVERAL LMS FILTERS: THE M-CLMS ALGORITHM

The main topic of this paper is the extension of the CLMS algorithm to allow the combination of an arbitrary number of individual filters. When doing so, the weight vector of the combined M-CLMS algorithm becomes:

$$\mathbf{w}_{eq}[k] = \sum_{i=1}^{L} \lambda_i[k]\mathbf{w}_i[k] \tag{6}$$

L being the total number of LMS filters that are placed into the combination, and $\mathbf{w}_i$ the weights of the i-th LMS filter with μ_i adaption step (as before, $\mu_1 > \mu_2 > \cdots > \mu_L$). As we previously explained for the CLMS algorithm, the L component filters operate, in principle, completely decoupled, and adapted using standard LMS rule.

During the derivation of the CLMS algorithm, we were able to see the importance of using a convex combination and limiting the values of $\lambda[k]$ for the good performance of the algorithm. Similarly, in this case we will use a *softmax* activation function to obtain the weights assigned to each individual filter:

$$\lambda_i[k] = \frac{exp(a_i[k])}{\sum_{j=1}^{L} exp(a_j[k])}; \;\; i = 1, \cdots, L \tag{7}$$

which guarantees that $0 \le \lambda_i[k] \le 1$, and $\sum_{i=1}^{L} \lambda_i[k] = 1$.

Using LMS for the adaption rules of the $a_i[k]$ parameters we obtain

$$\begin{aligned} a_i[k+1] &= a_i[k] - \frac{\mu_a}{2}\frac{\partial e_{eq}^2[k]}{\partial a_i[k]} \\ &= a_i[k] + \mu_a\left(d[k] - \mathbf{w}_{eq}^T[k]\mathbf{x}[k]\right) \\ &\quad \left(\sum_{j=1}^{L}\frac{\partial \lambda_j[k]}{\partial a_i[k]}\mathbf{w}_j[k]\right)^T \mathbf{x}[k]; \;\; i = 1, \cdots, L. \end{aligned} \tag{8}$$

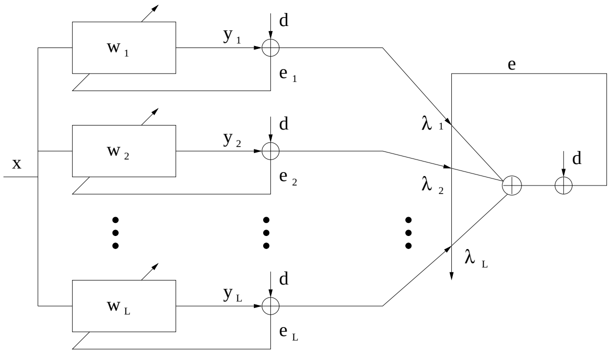


Fig. 1. The proposed M-CLMS adaptive filter. L independent LMS filters with different adaption steps combine their outputs to produce the overall output of the filter. The error achieved by the combined scheme is used to update the values of the mixing coefficients.

The partial derivatives of $\lambda_j[k]$ with respect to $a_i[k]$ are given by

$$\frac{\partial \lambda_i[k]}{\partial a_i[k]} = \lambda_i[k] - \lambda_i^2[k] \qquad (9)$$
$$\frac{\partial \lambda_j[k]}{\partial a_i[k]} = -\lambda_i[k]\lambda_j[k]; \;\; j \neq i.$$

Now, substituting (9) into (8), we finally get the adaption rule for the mixing parameters

$$a_i[k+1] = a_i[k] + \mu_a \left(d[k] - \mathbf{w}_{eq}^T[k]\mathbf{x}[k]\right)\lambda_i[k] \qquad (10)$$
$$\left(\mathbf{w}_i[k] - \mathbf{w}_{eq}[k]\right)^T \mathbf{x}[k]; \;\; i = 1, \cdots, L.$$

Again, parameter μ_a must be fixed to a value much higher than the highest adaption step of any individual LMS filter. For practical considerations, we also impose a limit on the absolute value of the $a_i[k]$ parameters to guarantee that $\lambda_i[k] < 0.95$. It can be shown that this condition is satisfied if we keep $|a_i[k]| \leq \frac{1}{2}\ln(19(L-1))$.

In Figure 1 we summarize the M-CLMS algorithm, indicating which error is used to adapt both the filters parameters and the mixing coefficients.

Again, to get a more practical M-CLMS implementation, it is advisable to include procedures of weight transfer and a momentum for the adaption of the mixing coefficients. Regarding the weight transfer, the problem arises in deciding which weights should be transferred, and to which filters. We will always transfer $\mathbf{w}_{eq}[k]$ to all the filters that are significantly different and are performing worse than the complete combined scheme. To be more precise, when $\lambda_i < \beta$ and $\frac{\|\mathbf{w}_{i\perp}\|_2}{\|\mathbf{w}_{i\|}\|_2} > \gamma$ (where $\mathbf{w}_{i\perp}$ and $\mathbf{w}_{i\|}$ are the components of $\mathbf{w}_i$ that are perpendicular and parallel to $\mathbf{w}_{eq}$, respectively), we replace the adaption of the weights of the i-th filter to be

$$\mathbf{w}_i[k+1] = \alpha\left(\mathbf{w}_i + \mu_i e_i[k]\mathbf{x}[k]\right) + (1-\alpha)\mathbf{w}_{eq}[k]. \qquad (11)$$

Finally, the inclusion of a momentum in (10) will serve to speed up the adaption during long steady intervals, where factor $\mathbf{w}_i[k] - \mathbf{w}_{eq}[k]$ is close to 0 for a number of filters.

$$a_i[k+1] = a_i[k] - \frac{\mu_a}{2}\frac{\partial e_{eq}^2[k]}{\partial a_i[k]} + \rho\left(a_i[k] - a_i[k-1]\right). \qquad (12)$$

IV. EXPERIMENTS

In this section we will use a simple example to illustrate the advantages of our algorithm in plant identification tasks. The plant we use is a 3-tap transversal filter with input $x[k]$ being a white, Gaussian, zero-mean, unit variance signal. At the output of the plant we add measurement noise $n[k]$, which is the same kind of signal as $x[k]$ but with variance 10^{-2}. The coefficients of the plant are initially selected to be

$$\mathbf{w} = [w_0, w_1, w_2]^T = [0.8212, 0.1620, -0.5897]^T$$

and then, changes are introduced in the third coefficient which linearly increases (see Fig. 2) towards $w_2 = 1.7691$ from $k = 3500$ to $k = 4500$, and then towards $w_2 = 2.3588$ and $w_2 = 2.5947$ at $k = 6500$ and $k = 8500$, respectively. Therefore, changes occur in the plant at different rates.

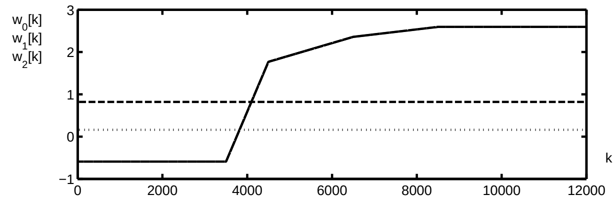


Fig. 2. Evolution of the coefficients of the 3-tap plant. $w_0[k]$ and $w_1[k]$ (dashed and dotted lines, respectively) remain constant during the example, while the third coefficient (solid line) introduces changes in the plant at different rates.

The learning rates that will be used are $\mu_1 = 0.1$, $\mu_2 = 0.03$, $\mu_3 = 0.01$, $\mu_4 = 0.003$, and $\mu_a = 200$ for the combination (so, $\mu_a \gg \mu_1 > \mu_2 > \mu_3 > \mu_4$). The parameters for the weight transfer procedure have been set to $\alpha = 0.8$, $\beta = 0.8$ and $\gamma = 0.03$, and the momentum used in the learning of the $a_i[k]$ parameters is $\rho = 0.5$.

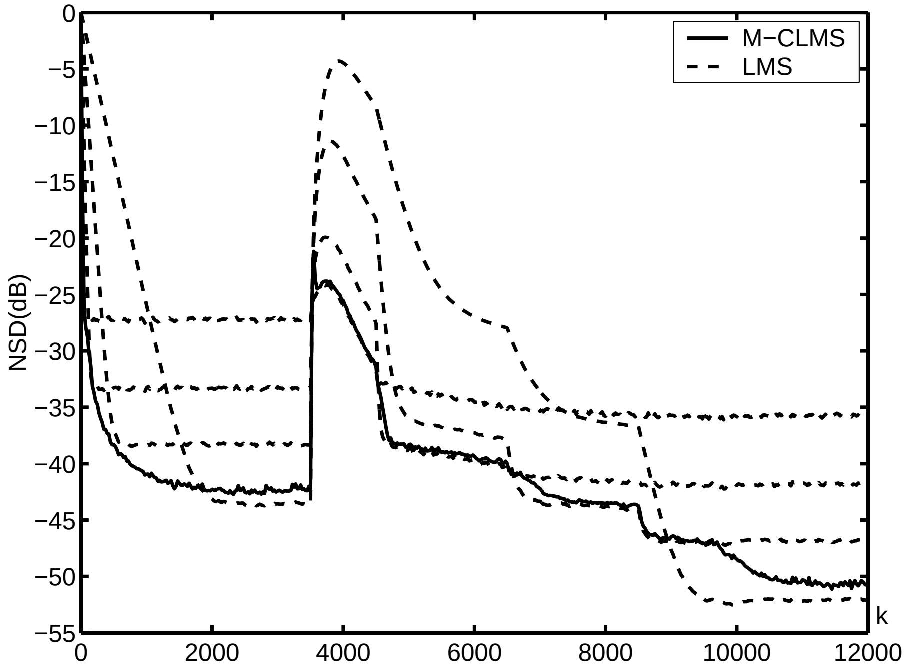
We have initialized the weights of all the LMS filters to the zero vector, and the $a_i[k]$ parameters are initially set to get $\lambda_1[0] = 0.95$ and $\lambda_2[0] = \lambda_3[0] = \lambda_4[0] = 0.05/3$.

The merit figure that will be used through our simulations is the Normalized Square Deviation (NSD) between the real and the estimated plant

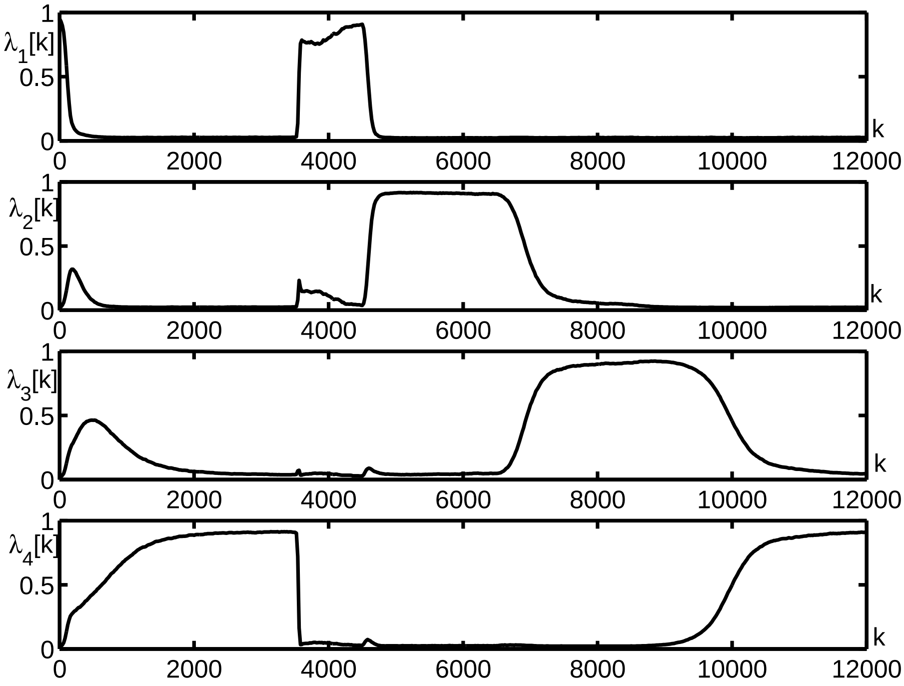
$$\mathrm{NSD}[k] = \frac{\|\mathbf{w}[k] - \hat{\mathbf{w}}[k]\|_2^2}{\|\mathbf{w}[k]\|_2^2} \qquad (13)$$

All the results are averaged over 1000 runs.

In Figure 3(a) we show the performance of the proposed M-CLMS filter against that of the μ_1, μ_2, μ_3 and μ_4 LMS filters. Figure 3(b) shows the evolution of the mixing coefficients as a function of k. We can see (looking at either the NSD or $\lambda_i[k]$ graphics) that, following the initial transition, M-CLMS



(a)



(b)

Fig. 3. Performance of the M-CLMS filter. (a) NSDs achieved by LMS filters with $\mu_1 = 0.1$, $\mu_2 = 0.03$, $\mu_3 = 0.01$ and $\mu_4 = 0.003$ are depicted using dashed lines. NSD of the combined M-CLMS filter with $\mu_a = 200$, $\alpha = 0.8$, $\beta = 0.8$, $\gamma = 0.03$ and $\rho = 0.5$ is represented using a solid line; (b) evolution of the mixing coefficients λ_1, λ_2, λ_3 and λ_4.

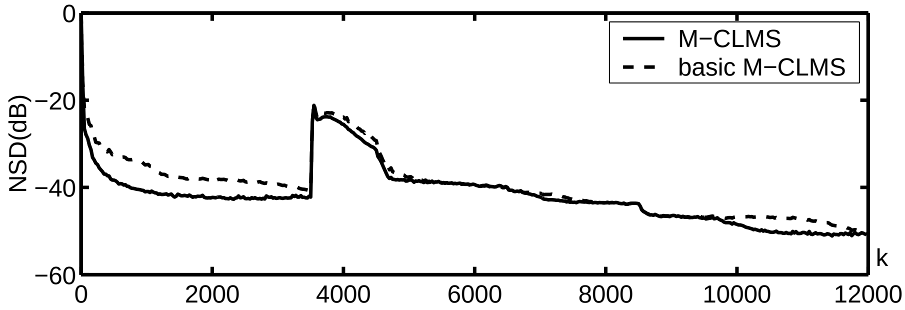


Fig. 4. Performance of the M-CLMS filter when used with and without speeding up procedures (solid and dashed lines, respectively).

softly changes from the fastest LMS filter to the slowest one, thus achieving the good initial convergence properties of the μ_1 LMS filter, and the low misadjustment error of the μ_4 filter.

Later, at the non stationary interval ($3500 < k < 8500$), the M-CLMS filter is, at each time, tuned with the LMS filter which is achieving a lower identification error. Thus, during $3500 < k < 4500$ it is the μ_1 LMS filter which performs best and, correspondingly, $\lambda_1[k] \approx 1$. During $4500 < k < 6500$ and $6500 < k < 8500$ the best filters are the μ_2 and μ_3 LMS filters, respectively, and we have $\lambda_2[k] \approx 1$ for $4500 < k < 6500$ and $\lambda_3[k] \approx 1$ for $6500 < k < 8500$. Finally, from $k = 8500$, the plant remains without changes and, as a consequence, $\lambda_4[k]$ evolves towards 1, reflecting the best performance of the μ_4 LMS filter in steady state situations.

Next, we illustrate the importance of the speeding up procedures that have been devised. In Fig. 4 we compare NSD evolution for M-CLMS when speeding up procedures are activated/deactivated. The use of these procedures leads to a lower NSD. This is specially clear following the initial transition, when the weight transfer procedure makes the combined filter converge to the misadjustment of the slowest LMS filter at $k \approx 1000$. However, a μ_4 LMS filter would not get that identification error before $k = 2000$ (see Fig. 3(a)). The effect of the momentum is well appreciated after $k = 10000$, when it speeds up the switching from $\mathbf{w}_{eq}[k] \approx \mathbf{w}_3[k]$ to $\mathbf{w}_{eq}[k] \approx \mathbf{w}_4[k]$.

We end this section by comparing the performance of M-CLMS with other previous proposals. In Figure 5(a) we have depicted M-CLMS and CLMS NSDs. CLMS has been used with μ_1 and μ_4 for the fast and slow LMS filters, and the same settings of M-CLMS for the speeding up procedures. While CLMS keeps the tracking capabilities and the precision of a μ_1 and a μ_4 LMS filters, respectively, it can be seen from the results in interval $4500 < k < 8500$, that incorporating extra component filters with intermediate adation steps can give some advantage at certain variation rates.

In Figure 5(b) we compare M-CLMS with an algorithm taken from [7]: Variable-Step LMS (VS-LMS). This algorithm increases/decreases the value of the adaption step by multiplying/dividing it by a constant factor (that we have fixed to $c = 1.1$ in our experiments), depending on the signs of each of the components of the gradient of the error with respect to the weights, along some of the last iterations (for a detailed description of the algorithm, please refer to [7]). It should be noted that VS-LMS uses a different adaption step for each weight in the adaptive filter, what could occasionally give it some advantage over M-CLMS.

VS-LMS requires the introduction of two parameters, m_0 and m_1, which must be selected a priori. According to [7], $m_0 = m_1$ is the natural choice when the algorithm is used in non stationary scenarios. In Fig. 5(b) we have depicted (dotted line) the NSD achieved for $m_0 = m_1 = 1$. In this case, M-CLMS outperforms VS-LMS during all the example. Selection of $m_0 < m_1$ favors the decreasing of the adaption step. Consequently, we can see that VS-LMS with $m_0 = 1$ and $m_1 = 2$ (dashed line) obtains a very good performance in stationary environments (see its initial convergence, and the low NSD after $k = 8500$). However, during $3500 < k < 8500$, when the plant is suffering changes, VS-LMS NSD is clearly higher than that of M-CLMS. We have carried out more experiments for other values of m_0 and m_1, arriving at the conclusion that the selection of these parameters implies itself a compromise of performance during varying/stationary intervals.

We believe that the advantage of using a convex combination of LMS filters over other algorithms that directly manage the value of the adaption step is that, rather than trying to guess an appropriate value for the adaption parameter, M-CLMS (and also CLMS) just has to decide which is the best filter from a set of candidates with different adaption steps.

V. CONCLUSIONS

In this paper we have presented the natural extension of the CLMS algorithm to the case in which we want to combine an arbitrary number of LMS filters. To respect the basic philosophy inherent to CLMS, we have included in the M-CLMS algorithm a *softmax* function to activate the mixing coefficients that are used to combine the outputs of the individual filters. Besides, a new procedure for transferring

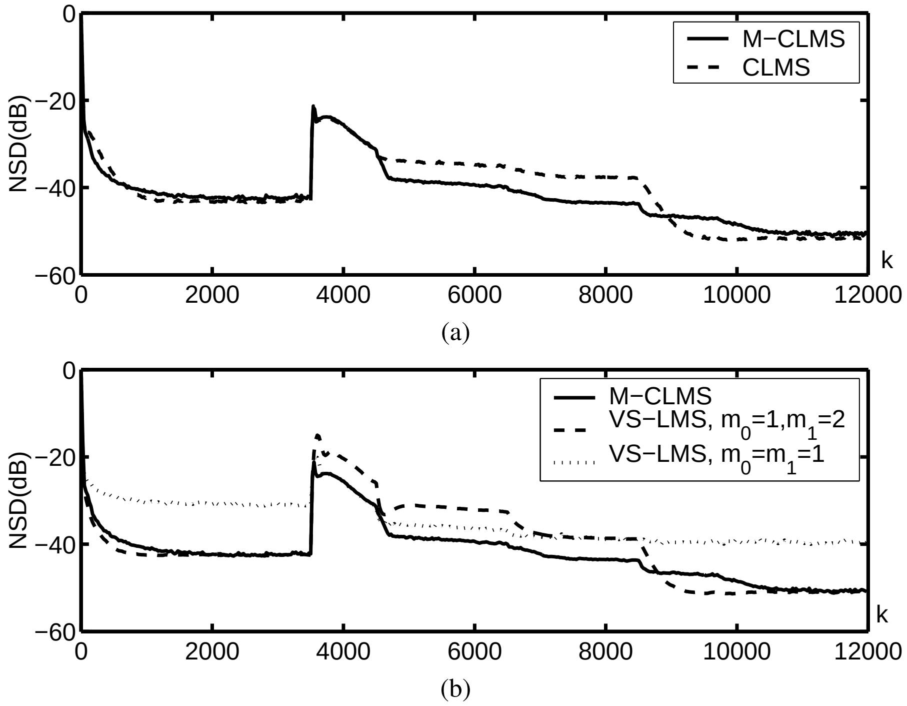


Fig. 5. Performance of the M-CLMS filter against that of CLMS and VS-LMS. (a) Comparison of performances of the M-CLMS and the CLMS filters (solid and dashed lines, respectively) with the same settings. (b) NSDs achieved by the M-CLMS filter (solid line) and VS-LMS filters for $m_0 = m_1 = 1$ (dotted line) and for $m_0 = 1$ and $m_1 = 2$ (dashed line).

weights across filters has been designed, to get a better practical performance.

Although the CLMS filter itself serves to combine high speed with low steady-state error, some computer simulations have given evidence that the inclusion in the scheme of a higher number of filters, operating with a wider range of adaption steps, provides a better performance when tracking changes that occur in the plant at different speeds. M-CLMS performance has also shown to be superior to that of previous techniques.

The use of CLMS and M-CLMS type algorithms in other practical applications represents a logical direction for further research. In addition to this, the proposed algorithm can be further refined. We are currently working on schemes that make use of the values of the mixing parameters to efficiently manage the adaption steps in a 3 filter configuration. Finally, it is also possible to combine different algorithms for adaptive filtering in a similar manner.

REFERENCES

[1] B. Widrow and M. E. Hoff, "Adaptive Swithching Circuits," *Wescon Conv. Record*, pp. 96–140, 1960.

[2] S. Haykin, *Adaptive Filter Theory (4th. ed.)*, Prentice-Hall, Upple Saddle River, NJ, 2002.

[3] E. Walach and B. Widrow, "The Least Mean Fourth (LMF) Algorithm and Its Family," *IEEE Trans. Information Theory*, vol. IT-30, no. 2, pp. 275–283, 1984.

[4] J.A. Chambers, O. Tanrikulu, and A.G. Constantinides, "Least Mean Mixed-Norm Adaptive Filter," *Electronic Letters*, vol. 30, no. 19, pp. 1574–1575, 1994.

[5] D. I. Pazaitis and A. G. Constantinides, "LMS - F Algorithm," *Electronic Letters*, vol. 31, no. 17, pp. 1423–1424, 1995.

[6] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "Adaptive Combination of Transversal LMS Filters for Plant Identification," submitted to *IEEE Trans. on Signal Processing*, 2002.

[7] R. W. Harris, D. M. Chabries, and F. A. Bishop, "Variable Step (VS) Adaptive Filter Algorithm," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp. 305–316, 1986.

[8] M. Martínez-Ramón, J. Arenas-García, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "An Adaptive Combination of Adaptive Filters for Plant Identification," in *Proc. of the 14th Intl. Conf. on Digital Signal Proc.*, Santorini, Greece, July 2002, pp. 1195–1198.

[9] A. C. Singer and M. Feder, "Universal Linear Prediction by Model Order Weighting," *IEEE Trans. on Signal Proc.*, vol. 47, no. 10, pp. 2685–2700, 1999.

[10] S. S. Kozat and A. C. Singer, "Multi-stage Adaptive Signal Processing Algorithms," in *Proc. of the 2000 IEEE Sensor Array and Multichannel Signal Processing Workshop*, Cambridge, MA, March 2000, pp. 380–384.