CONVEX COMBINATION OF AFFINE PROJECTION ALGORITHMS

Miguel Ferrer, Maria de Diego, Alberto Gonzalez, and Gema Piñero

Audio and Communications Signal Processing Group (GTAC)
Institute of Telecommunications and Multimedia Applications (iTEAM)
Universidad Politecnica de Valencia, 46022-Valencia, SPAIN
Phone: + (34) 963879763, email: mferrer@dcom.upv.es
http://www.iteam.upv.es

ABSTRACT

It is well known that performance of adaptive filters is mainly a compromise among computational complexity, speed of convergence and steady-state behavior. The affine projection (AP) algorithm offers a good convergence speed that increases with the projection order N and a computational complexity that can be reduced by applying different fast strategies. However, its steady-state mean square error (MSE) worsens when N grows. This work introduces the convex combination of two AP adaptive filters in order to improve the performance capabilities of the overall filter. The purpose of the convex AP approach is to improve the convergence performance of a single AP algorithm but not at the expense of an increase of the steady-state MSE. To achieve this we combine two AP filters with different projection orders, one of high N order that performs faster than other with a lower order but with a better MSE. Moreover, the computational cost of the AP combination scheme would be similar to that of the higher order AP filter working separately. Simulation results have validated the proposed approach.

1. INTRODUCTION

In adaptive filtering, affine projection (AP) algorithms are attractive mainly due to its fast convergence speed in comparison with LMS-type classical algorithms [1]. Moreover, this algorithm exhibits a convergence speed that increases with its projection order (N). However, this performance improvement involves two negative effects: firstly there is an increase of its computational complexity that could become too high for many applications, and secondly, the mean square error (MSE) at steady-state worsens when N grows [2]. The first issue can be suitably minimized by applying different fast strategies that lower the computational cost of the AP algorithm [3, 4], and even in case of real-time applications, by using hardware platforms powerful enough where the computational load is not constrained. Nevertheless, regarding the second issue, the MSE at steady-state is inherent to the adaptive algorithm and it can not be applied any strategy to improve it.

The aim of this paper is to develop a new adaptive filtering approach based on the parallel combination of two AP adaptive filters which supplements the work presently available in the literature. The purpose of the AP combination is to improve the convergence speed of a single AP algorithm but not at the expense of an increase in the steady-state MSE. The convex combination strategy presented in [5], whose basic idea is to combine two algorithms of complementary capabilities to get a global filter of improved performance, is being used. In this work we consider two AP algorithms with different projection orders, one of high N order that performs faster and other with a lower order but with a better steady-state MSE. The two adaptive filters are independently adapted using its own error signal and the outputs of both filters are combined by a mixing parameter that is adjusted in order to minimize the overall mean square error. It should also be noted that the rule to determine the optimal combination of the adaptive filters is not trivial and requires a great effort. Some works have been previously published regarding this issue for LMS type adaptive filters, see [6]. On the other hand, in case the two AP adaptive filters of the parallel scheme have the same number of filter coefficients, most of the computational load of the lower order algorithm is shared with the algorithm of higher order, so the computational cost of the parallel structure, which is contrary to what might think, is not much larger than the computational cost of the AP algorithm with higher order working individually.

This paper is organized as follows. Section 2 briefly reviews the adaptive controller using the AP algorithm. In Section 3 the convex combination of the AP algorithm is introduced. Section 4 presents some experimental results and finally, conclusions follow in Section 5.

Partially supported by: CICYT grant TEC2006-13883-C04-01.

2. AFFINE PROJECTION ALGORITHM

The affine projection algorithm is an adaptive filter whose objective is to iteratively estimate the adaptive filter weights in such a way that a function of the error signal $e[n]$ is minimized. This type of algorithms is based on multidimensional orthogonal projections on affine subspaces and was firstly introduced in [7]. Figure 1 shows the block diagram of an adaptive filter using the AP algorithm.

This algorithm uses N data vectors to update the filter coefficients and can be considered as a generalization of the Normalized LMS (NLMS) algorithm [1], which uses only one input signal vector. Regarding the NLMS algorithm, the update equation can be given by,

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu \mathbf{x}[n](\mathbf{x}^T[n]\mathbf{x}[n])^{-1}e[n] \quad (1)$$

being $\mathbf{w}[n]$ a vector formed by the L adaptive filter coefficients at the nth time instant, μ is a step-size parameter, and $\mathbf{x}[n]$ is a vector with the most recent L samples of the input signal $x[n]$,

$$\mathbf{x}[n] = (x[n], x[n-1], ..., x[n-L+1])^T. \quad (2)$$

On the other hand, the AP algorithm is characterized by the adaptation rule of

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu \mathbf{A}[n](\mathbf{A}^T[n]\mathbf{A}[n])^{-1}\mathbf{e}[n], \quad (3)$$

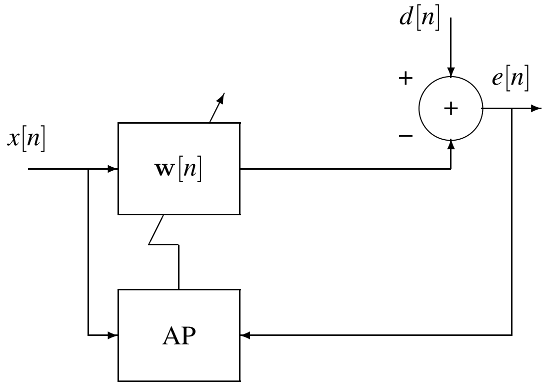

Figure 1: Block diagram of an adaptive system using the AP algorithm

where $\mathbf{A}[n]$ is a matrix of size $L \times N$ defined as

$$\mathbf{A}[n] = (\mathbf{x}[n], \mathbf{x}[n-1], \ldots \mathbf{x}[n-N+1]), \quad (4)$$

and the $N \times 1$ error vector $\mathbf{e}[n]$ is expressed as,

$$\mathbf{e}[n] = \mathbf{d}[n] - \mathbf{A}[n]\mathbf{w}[n] \quad (5)$$

$\mathbf{d}[n]$ being an $N \times 1$ vector composed of the desired signal samples.

Thus, N data vectors are used to update the L filter coefficients at each iteration as given in (3). This rule is the solution to an optimization problem constrained by the *minimum perturbation principle*, which implies a minimum variation of the weight vector while constraining the filter coefficients so that a replica of the desired signal $d[n]$ is generated by filtering the input signal $x[n]$ as shown in Fig. 1. The weight vector variation is defined as

$$\Delta\mathbf{w}[n+1] = \mathbf{w}[n+1] - \mathbf{w}[n] \quad (6)$$

and the objective cost function to minimize is given by

$$\|\Delta\mathbf{w}[n+1]\|^2 = \Delta\mathbf{w}^T[n+1]\Delta\mathbf{w}[n+1] \quad (7)$$

with the constraint

$$\mathbf{w}^T[n+1]\mathbf{x}[n-k] = d[n-k], \quad k = 0, 1, \ldots, N-1 \quad (8)$$

where the projection order N is lower than L.

In practical applications, possible instabilities due to the $(\mathbf{A}^T[n]\mathbf{A}[n])^{-1}$ matrix inversion can be avoided by adding the term $\delta\mathbf{I}$, where δ is a small positive constant and $\mathbf{I}$ is the $N \times N$ identity matrix. This technique is called regularization. Then we may rewrite (3) as

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu\mathbf{A}[n](\mathbf{A}^T[n]\mathbf{A}[n] + \delta\mathbf{I})^{-1}\mathbf{e}[n]. \quad (9)$$

The update equation in (9), verifies on the one hand that the computational cost grows with the projection order, and on the other hand, that the NLMS algorithm defined in (1) is a particular case of the AP algorithm for $N = 1$ and $\delta = 0$, although the NLMS also supports the regularization parameter [8]. Regarding the performance of both algorithms, the NLMS exhibits a worse convergence speed, but a smaller steady-state MSE, see more details in [2]. In Fig. 2 it can be easily seen how the speed of convergence increases with the projection order but also the steady-state MSE. Therefore it would be interesting to combine the steady-state performance of an AP algorithm with low N order (considering also the NLMS) with a fast AP algorithm for high N order. Next section describes the convex combination scheme that solves this problem.

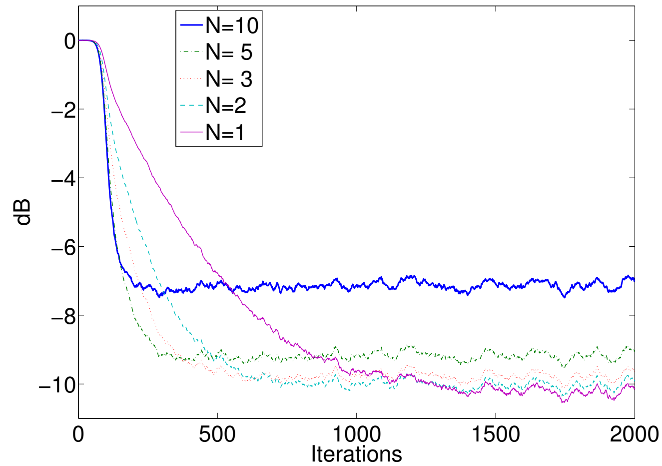

Figure 2: Convergence curves of the AP algorithm for different projection orders: $N = 1$, 2, 3, 5 and 10

3. CONVEX AP ALGORITHM

The aim of a convex combination approach is to improve the performance of adaptive schemes so that the overall performance is better than the performance of each algorithm separately or at least as good as the best individual algorithm [5]. In a classical adaptive system, the target is to minimize a cost function dependent on the desired signal $d[n]$ and on the input signal $x[n]$ that feeds the adaptive filter. In an AP convex combination scheme, rather than using a single AP adaptive filter we use two AP adaptive filters, each of them generating a different output. The algorithm that determines the coefficients of both filters (see $\mathbf{w}_1[n]$ and $\mathbf{w}_2[n]$ in Fig. 3) is given by the AP algorithm (9). Taking into account the output of both filters, $y_1[n]$ and $y_2[n]$ at time n, we obtain the output of the parallel filter as

$$y[n] = \lambda[n]y_1[n] + (1 - \lambda[n])y_2[n], \quad (10)$$

being $\lambda[n]$ a mixing parameter in the range $(0, 1)$ that controls the combination of the two filters at each iteration, and comes from

$$\lambda[n] = \frac{1}{1 + e^{-a[n]}}, \quad (11)$$

where $a[n]$ is updated in order to minimize the instantaneous square error of the overall filter, $J[n] = e[n]^2 = (d[n] - y[n])^2$, by using the gradient descent method [5]. Thus, the update

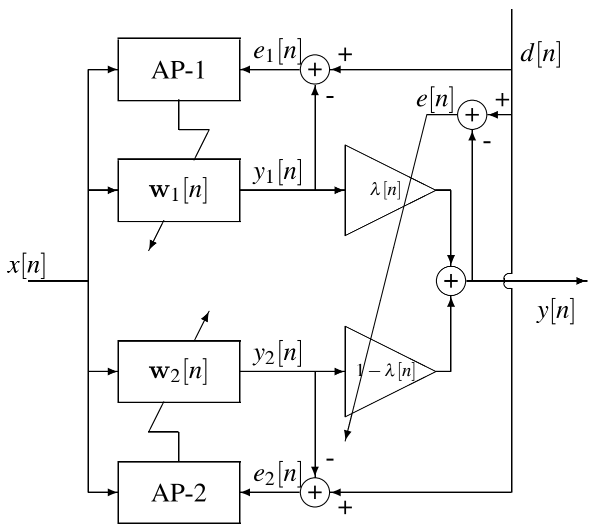

Figure 3: Block diagram of the convex AP scheme

equation for $a[n]$ is given by

$$\begin{aligned} a[n+1] &= a[n] - \tfrac{\mu_a}{2}\nabla J[n] \\ &= a[n] + \mu_a e[n](e_2[n] - e_1[n])\lambda[n](1-\lambda[n]), \end{aligned} \tag{12}$$

being

$$\frac{d\lambda[n]}{da[n]} = \lambda[n](1-\lambda[n]). \tag{13}$$

Moreover $e_1[n]$ and $e_2[n]$ are the output error signals of the component filters, and μ_a is a step-size parameter that control changes in $a[n]$ from one iteration to the next. Its correct adjustment depends on some characteristics of the system such as the step sizes of adaptive filters or the input signal. On the other hand, the update of $a[n]$ at each iteration tries to extract the best properties of combined filters. The rule to determine the optimum combination is not trivial and different schemes have been previously proposed (see e.g. [6]).

It is clear that this AP combination approach involves a computational burden higher than the classic adaptive filter, but in general this complexity is not twice the complexity of the simple AP algorithm because the parallel combination usually involves an AP algorithm of low order (with a small computational load). Moreover, in the typical case in which both algorithms have the same number of filter coefficients, as we assume in this work, they could share the data buffers for the desired and reference signals, further reducing memory and complexity requirements of the combination. It should also be noted that the implementation of the AP algorithm implies a matrix inversion (see (9)) that turns out to be the main computational burden of the algorithm. Nevertheless some iterative strategies [9] already proposed, such as the Levinson-Durbin recursion [10], could be applied in order to reduce the complexity of the convex AP algorithm. The applied iterative method efficiently provides the inverse of a matrix of N order from the inverse matrix of $N-1$ order. Therefore computation of matrix inversions for the AP algorithm with low order does not involve any additional cost since it is embedded in the calculation of the inverse matrix of the higher order AP algorithm. Table 1 evaluates the complexity of the considered algorithms. It illustrates the number of multiplications required by the convex AP and the standard AP component filters for the general case and also for a typical case ($L = 45$, $N_1 = 1$ and $N_2 = 10$). It can be seen that the computational complexity of the convex AP algorithm for $N_1 = 1$ and $N_2 = 10$ is much higher than that of single AP algorithm for $N_1 = 1$, but only slightly more complex than single AP with $N_2 = 10$.

From what has been said, it can be deduced that the convex AP algorithm reduces the transient period before the algorithm converges and achieves a good steady-state MSE. To obtain this we combine the better convergence performance of a high order AP algorithm with the better steady-state performance of an AP algorithm with a low N order. In case of rapid transitions in the desired signal, the convex AP combination should follow changes in the signal without worsening its steady-state performance.

4. SIMULATION RESULTS

In order to test the performance of the proposed parallel combination of AP algorithms as an alternative to the commonly applied single AP algorithms some simulation results have been obtained. Two qualities have been considered: convergence speed and the steady-state MSE. Convergence is defined as the ratio between the instantaneous estimated power of the error signal and the instantaneous estimated power of the desired signal, expressed in decibels. Several simulations have been carried out using a uniform random signal of zero mean and variance $\sigma_x^2 = 1$ to produce the input signal $x[n]$. The desired signal has been generated by filtering the input signal through a moving average (MA) filter of order 50 whose coefficients have been randomly chosen. In order to better estimates the signal power, the algorithms were run 1000 times, each with 10000 iterations. Two AP algorithms with the same step sizes ($\mu_1 = \mu_2 = 0.001$), and adaptive filters of 45 coefficients have been used as the filter components. Moreover, μ_a was set to 1.

Fig. 4 illustrates the convergence curves of the convex AP approach (combining an AP algorithm for $N = 10$ and the AP for $N = 1$ (NLMS)) and comparing it with the convergence curves of the two single AP algorithms independently working. As can be seen, the combination approach presents the fast convergence of the rapid AP algorithm and the low steady-state MSE of the AP algorithm for $N = 1$.

Performance of AP combination scheme has also been studied if there are changes of either the plant or the input signal. Thus the experiment has been carried out using the same configuration as before, but the variance of the input signal and the plant were varied during the experiment. Changes were performed every 10000 samples and the algorithms were run during 40000 iterations. That means the algorithms should converge four times. As it can be seen in Fig. 5 the convex AP approach behaves as well as the best component filter. In the transient periods it follows the faster AP algorithm and once it has reached the steady-state follows

Algorithm	Multiplies per iteration	Typical case
AP filter ($N_1 = 1$)	$2L + N_1^2(L+1) + 2N_1L + O(N_1^3/2)$	226.5
AP filter ($N_2 = 10$)	$2L + N_2^2(L+1) + 2N_2L + O(N_2^3/2)$	6090
Convex AP	$4L + N_1^2(L+1) + N_2^2(L+1) + (N_1+N_2)L + O(N_2^3/2) + L(N_1+N_2) + 5$	6321

Table 1: Number of multiplications per iteration of the AP component filters and of their convex AP combination approach for the general case and also for a typical case ($L = 45$, $N_1 = 1$ and $N_2 = 10$).

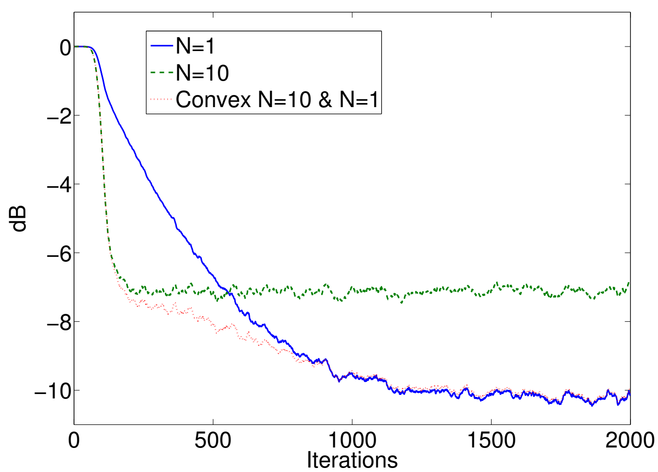

Figure 4: Convergence curves of the AP component filters (for $N = 10$ and $N = 1$) and of their convex AP combination approach.

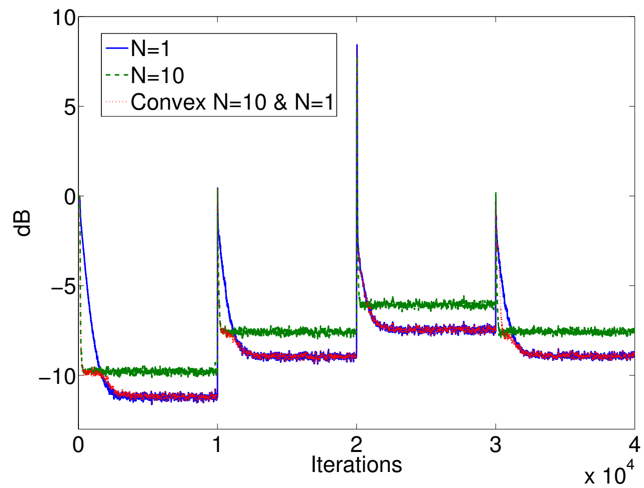

Figure 5: Convergence curves of the AP component filters (for $N = 10$ and $N = 1$) and of their convex AP combination approach for non-stationary condition.

the slow AP algorithm.

In order to improve even more the steady-state performance of the convex AP approach, we can think of combining a high order AP filter with an LMS adaptive filter, which exhibits a lower steady-state MSE. Thus, this hybrid convex approach reduces even more the final MSE but with a moderate convergence speed. Fig. 6 shows the convergence curves of the hybrid convex approach (AP filter for $N = 10$ and LMS filter), the convex AP approach (two AP filters for $N = 10$ and $N = 1$), and the different single component filters (LMS, AP for $N = 10$ and $N = 1$).

5. CONCLUSIONS

This work presents a new approach that improves the performance of AP adaptive filters. The applied scheme, previously introduced for LMS filters, uses two AP adaptive filters that are independently adapted using its own error signal and mixes their outputs to improve the performance of the overall filter. The purpose of the AP combination scheme is to obtain an AP adaptive filter with fast convergence speed (using an AP adaptive filter with high projection order) and small steady-state MSE (being the other component filter an AP algorithm with low projection order). This new scheme is especially suitable when trying to achieve high speeds but without sacrificing steady-state MSE performance. Moreover, it has been shown that, as an alternative to the convex AP combination scheme, a hybrid convex approach using a high order AP algorithm and an LMS algorithm can be considered, reducing even more the MSE but at the expense of a moderate loss in convergence speed.

Simulation results in stationary and non-stationary conditions have validated the convex AP scheme proposed. So, in case of rapid transitions in the desired signal, the convex AP combination follow changes in the signal without worsening its steady-state performance.

Further research suggests the implementation of convex AP schemes in real-time applications that require good convergence performance such as sound reproduction or control.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Ed., Upper Saddle River, NJ, fourth edition, 2002.

[2] H. Shin and A. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. on*

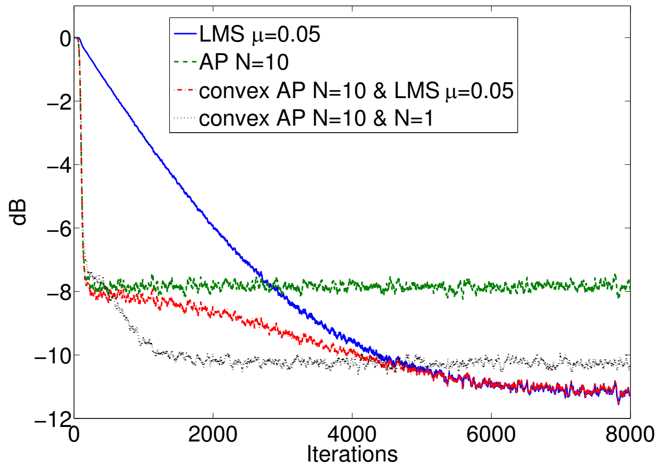

Figure 6: Convergence curves of the AP algorithms (for $N = 10$ and $N = 1$), of the LMS algorithm for $\mu = 0.05$ and of their convex combination approaches: convex AP algorithm and hybrid convex approach.

Signal Processing, vol. 52, no. 1, pp. 90–102, January 2004.

[3] S.L. Gay and S. Tavathia, "The fast affine projection algorithm," *ICASSP95, Proc. of the International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 3023–3026, May 1995, Detroit, MI.

[4] M. Ferrer, A. Gonzalez, M. de Diego, and G. Piñero, "Fast affine projection algorithms for filtered-x multichannel active noise control," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 16, no. 8, pp. 1396–1408, November 2008.

[5] J. Arenas-Garcia, A.R. Figueiras-Vidal, and A.H. Sayed, "Mean square performance of a convex combination of two adaptive filters," *IEEE Trans. on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[6] L.A. Azpicueta-Ruiz, A.R. Figueiras-Vidal, and J. Arenas-Garcia, "A normalized adaptation scheme for the convex combination of two adaptive filters," *ICASSP08, Proc. of Int. Conf. on Acoustics, Speech and Signal Process.*, pp. 3301–3304, 2008.

[7] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Proc. of the Electronics and Communications in Japan*, vol. J67-A, pp. 126–132, February 1984.

[8] Y. Choi and H. Shin, "Robust regularization for normalized LMS algorithms," *IEEE Trans. on Circuits and Systems*, vol. 53, no. 8, pp. 627–631, 2006.

[9] M. Ferrer, M. de Diego, A. Gonzalez, and G. Piñero, "Efficient implementation of the affine projection algorithm for active noise control aplications," *Proc. of the 12th European Signal Processing Conf.*, September 2004.

[10] P.M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, 1993.