CONJUGATE GRADIENT TECHNIQUES FOR MULTICHANNEL ADAPTIVE FILTERING

Lino García Morales and Fernando Juan Berenguer Císcar
Escuela Superior Politécnica,Universidad Europea de Madrid, Tajo S/N, Villaviciosa de Odón, Madrid, Spain
lino.garcia@uem.es, fjuan.berenguer@uem.es

Keywords: Multichannel Adaptive Filtering, System Identification, Optimization Method, Conjugate Gradient, Partitioned Frequency-Domain Adaptive Filtering.

Abstract: The conjugate gradient is the most popular optimization method for solving large systems of linear equations. In a system identification problem, for example, where very large impulse response is involved, it is necessary to apply a particular strategy which diminishes the delay, while improving the convergence time. In this paper we propose a new scheme which combines frequency-domain adaptive filtering with a conjugate gradient technique in order to solve a high order multichannel adaptive filter, while being delayless and guaranteeing a very short convergence time.

1 INTRODUCTION

The multichannel adaptive filtering problem's solution depends on the correlation between the channels, the number of channels and the order and nature of the impulse responses involved in the system. The multichannel acoustic echo cancellation (MAEC) application, for example, can be seen as a system identification problem with extremely large impulse responses (depending on the environment and its reverberation time, the echo paths can be characterized by FIR filters with thousands of taps).

In these cases a multirate adaptive scheme such a partitioned block frequency-domain adaptive filter (PBFDAF) (Páez and García, 1992) is a good alternative and is widely used in commercial systems nowadays. However, the convergence speed may not be fast enough under certain circumstances.

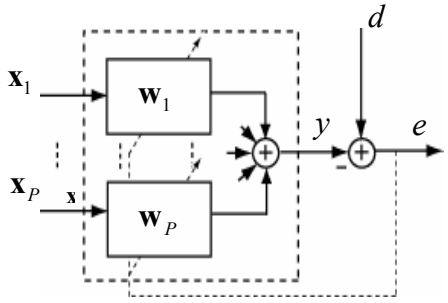

Figure 1: Multichannel Adaptive Filtering.

Figure 1 shows the working framework, where $\mathbf{x}_p$ represents the p channel input signal, d the desired signal, y the output of adaptive filter and e the error signal we try to minimize. In typical scenarios, the filter input signals $\mathbf{x}_p$, $p = 1,\ldots,P$ (where P is a number of channels), are highly correlated which further reduces the overall convergence of the adaptive filter coefficients w_{pm}, $m = 1,\ldots,L$ (L is the filter length),

$$y[n] = \sum_{p=1}^{P}\sum_{m=1}^{L} x_p[n-m]\,w_{pm} . \qquad (1)$$

The mean square error (MSE) minimization of the multichannel signal with respect to the filter coefficients is equivalent to the Wiener-Hopf equation

$$\mathbf{R}\mathbf{w} = \mathbf{r} . \qquad (2)$$

$\mathbf{R}$ represents the autocorrelation matrix and $\mathbf{r}$ the cross-correlation vector between the input and the desired signals. Both are a priori time-domain statistical unknown variables, although can be estimated iteratively from $\mathbf{x}$ and d.

$\mathbf{R} = E\left\{\mathbf{x}\mathbf{x}^H\right\}$ and $\mathbf{r} = E\left\{\mathbf{x}d^*\right\}$, with $\mathbf{x} = \left[\mathbf{x}_1^T \quad \dots \quad \mathbf{x}_P^T\right]^T$; $\mathbf{w} = \left[\mathbf{w}_1^T \quad \dots \quad \mathbf{w}_P^T\right]^T$ and $\mathbf{w}_p = \left[w_{p1} \quad \dots \quad w_{pL}\right]^T$. In the notation we are using a for scalar, $\mathbf{a}$ for vector and $\mathbf{A}$ for matrix; $\mathbf{a}$, $\mathbf{A}$ denotes vector and matrix respectively in a frequency-domain: $\mathbf{a} = \mathbf{F}\mathbf{a}$, $\mathbf{A} = \mathbf{F}\mathbf{A}$. $\mathbf{F}$ represents the discrete Fourier transform (DFT) matrix defined as $\mathbf{F}_{kl} = e^{-j2\pi kl/M}$, with $k,l = 0,\dots,M-1$, $j = \sqrt{-1}$ and $\mathbf{F}^{-1}$ as its inverse. Of course, in the final implementation, the DFT matrix is substituted by much more efficient fast Fourier transforms (FFT). Here $(.)^T$ denotes transpose operator and $(.)^H = \left((.)^T\right)^*$ the Hermitian operator (conjugate transpose).

The conjugate gradient (CG) method is efficient to obtain the solution to (2), however, a big delay is introduced (noted that the system order is $LP \times LP$). In order to reduce this convergence speed problem we propose a new algorithm which employs much more powerful CG optimization techniques, but keeping the frequency block partition strategy to allow computationally realistic low latency situations. The paper is organized as follows: Section 2 reviews the Multichannel PBFDAF approach and its implementation. Section 3 develops the Multichannel Conjugate Gradient Partitioned Frequency Domain Adaptive Filter algorithm (PBFDAF–CG). Results of the new approach are presented in Section 4 and 5 followed by conclusions.

2 PBFDAF

The PBFDAF was developed to deal efficiently with such situations. The PBFDAF is a more efficient implementation of Least Mean Square (LMS) algorithm in the frequency-domain. It reduces the computational burden and user-delay bounded. In general, the PBFDAF is widely used due to be good trade-off between speed, computational complexity and overall latency. However, when working with long impulse response, as the acoustic impulse responses (AIR) used in MAEC, the convergence properties provided by the algorithm may not be enough. Besides, the multichannel adaptive filter is structurally more difficult, in general, than the single channel case (Benesty and Huang, 2003).

This technique makes a sequential partition of the impulse response in the time-domain prior to a frequency-domain implementation of the filtering operation. This time segmentation allows setting up individual coefficient updating strategies concerning different sections of the adaptive canceller, thus avoiding the need for disabling the adaptation in the complete filter. The adaptive algorithm is based on the frequency-domain adaptive filter (FDAF) for every section of the filter (Shink, 1992).

The main idea of frequency-domain adaptive filter is to frequency transform the input signal in order to work with matrix multiplications instead of dealing with slow convolutions. The frequency-domain transform employs one or more DFTs and can be seen as a pre-processing block that generates decorrelated output signals.

In the more general FDAF case, the output of the filter in the time domain (1) can be seen as a direct frequency-domain translation of the block LMS (BLMS) algorithm. In the PBFDAF case, the filter is partitioned transversally in an equivalent structure. Partitioning $\mathbf{w}_p$ in Q segments (K length) we obtain

$$y[n] = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{m=0}^{K-1} x_p[n-qK-m]\, w_{p(qK+m)} \qquad (3)$$

Where the total filter length L, for each channel, is a multiple of the length of each segment $L = QK$, $K \le L$. Thus, using the appropriate data sectioning procedure, the Q linear convolutions (per channel) of the filter can be independently carried out in the frequency-domain with a total delay of K samples instead of the QK samples needed in standard FDAF implementations.

Figure 2 shows the block diagram of the algorithm using the overlap-save method. In the frequency domain with matrix notation, equation (3) can be expressed as

$$\mathbf{Y} = \mathbf{X} \otimes \mathbf{W}. \qquad (4)$$

Where $\mathbf{X} = \mathbf{F}\mathbf{X}$ represents a matrix of dimensions $M \times Q \times P$ which contains the Fourier transform of the Q partitions and P channels of the input signal matrix $\mathbf{X}$.

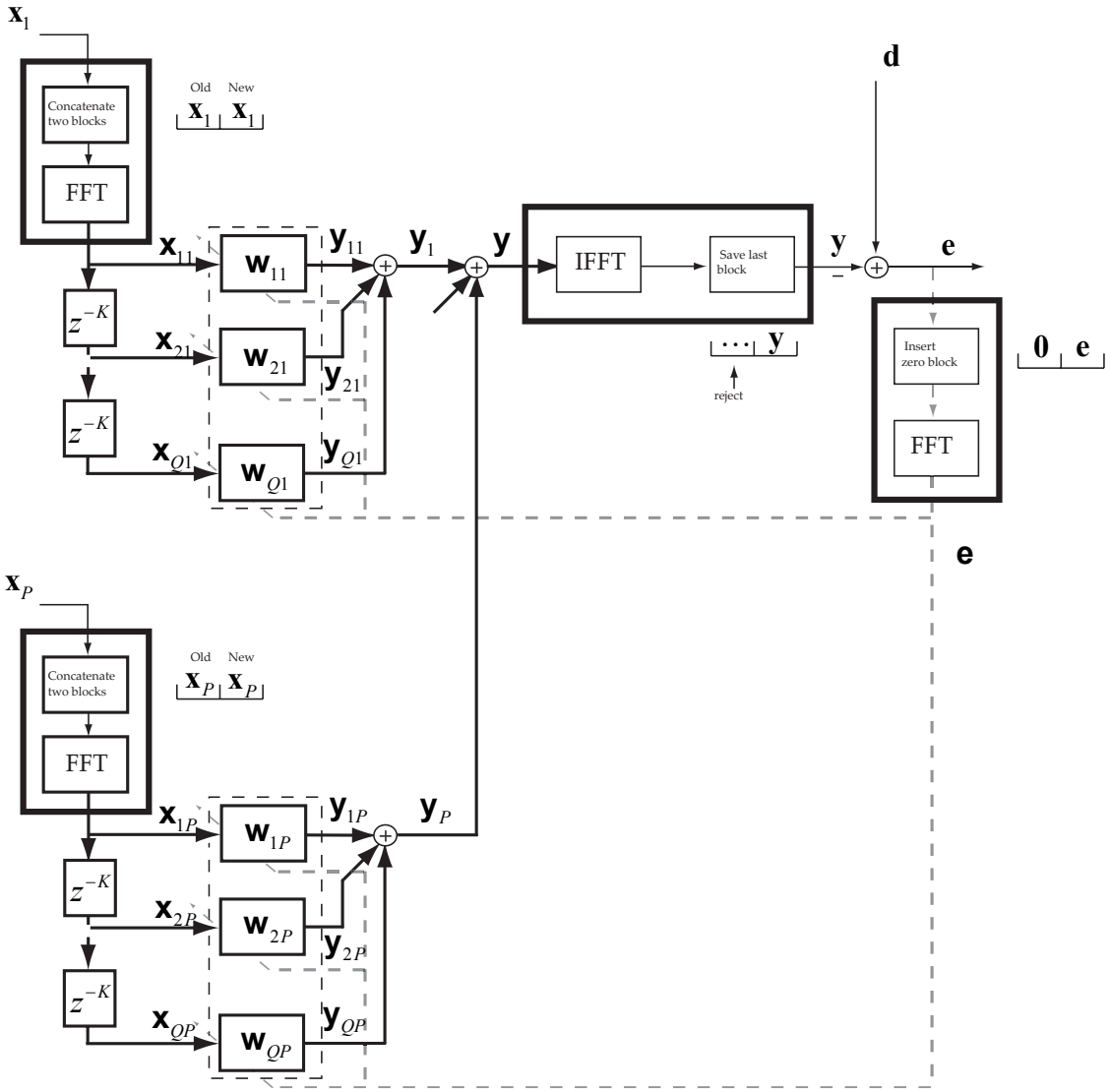

Figure 2: Multichannel PBFDAF (Overlap-Save method).

Being $\mathbf{X}$, $2K \times P$-dimensional (supposing 50% overlapping between the new block and the previous one).

It should be taken into account that the algorithm adapts every K samples. $\mathbf{W}$ represents the filter coefficient matrix adapted in the frequency-domain (also $M \times Q \times P$-dimensional) while the $\otimes$ operator multiplies each of the elements one by one; which in (4) represents a circular convolution.

The output vector $\mathbf{y}$ can be obtained as the double sum (rows) of the $\mathbf{Y}$ matrix. First we obtain a $M \times P$ matrix which contains the output of each channel in the frequency-domain $\mathbf{y}_p$, $p = 1, \ldots, P$, and secondly, adding all the outputs we obtain the output of the whole system $\mathbf{y}$. Finally, the output in the time-domain is obtained by using

$$\mathbf{y} = \text{last } K \text{ components of } \mathbf{F}^{-1}\mathbf{y}. \qquad (5)$$

Notice that the sums are performed prior to the time-domain translation. In this way we reduce $(P-1)(Q-1)$ FFTs in the complete filtering process. As in any adaptive system the error can be obtained as

$$\mathbf{e} = \mathbf{d} - \mathbf{y}, \qquad (6)$$
$$\mathbf{d} = \left[d[mK] \quad \ldots \quad d\left[(m+1)K - 1\right] \right]^T.$$

The error in the frequency-domain (for the actualization of the filter coefficients) can be obtained as

$$\mathbf{e} = \mathbf{F}\begin{bmatrix} \mathbf{0}_{K\times 1} \\ \mathbf{e} \end{bmatrix}. \quad (7)$$

As we can see, a block of K zeros is added to ensure a correct linear convolution implementation. In the same way, for the block gradient estimation, it is necessary to employ the same error vector in the frequency-domain for each partition q and channel p.

This can be achieved by generating an error matrix $\mathbf{E}$ with dimensions $M \times Q \times P$ which contains replicas of the error vector, defined in (7), of dimensions P and Q ($\mathbf{E} \leftarrow \mathbf{e}$ in the notation). The actualization of the weights is performed as

$$\mathbf{W}[m+1] = \mathbf{W}[m] + 2\mu[m] \otimes \mathbf{G}[m]. \quad (8)$$

The instantaneous gradient is estimated as

$$\mathbf{G} = -\mathbf{X}^{*} \otimes \mathbf{E}. \quad (9)$$

This is the unconstrained version of the algorithm which saves two FFTs from the computational burden at the cost of decreasing the convergence speed. As we are trying to improve specifically this parameter we have implemented the constrained version which basically makes a gradient projection. The gradient matrix is transformed into the time-domain and is transformed back into the frequency-domain using only the first K elements of $\mathbf{G}$ as

$$\mathbf{G} = \mathbf{F}\begin{bmatrix} \mathbf{G} \\ \mathbf{0}_{K\times Q\times P} \end{bmatrix}. \quad (10)$$

3 PBFDAF–CG

CG algorithm is a technique originally developed to minimize quadratic functions, as (2), which was later adapted for the general case (Luenberger, 1984). Its main advantage is its speed as it converges in a finite number of steps. In the first iteration it starts estimating the gradient, as in the steepest descent (SD) method, and from there it builds successive directions that create a set of mutually conjugate vectors with respect to the positively defined Hessian (in our case, the auto-correlation matrix $\mathbf{R}$ in the frequency-domain).

In each *m*-block iteration the conjugate gradient algorithm will iterate $k = 1, \ldots, \min(N, K)$ times; where N represent the memory of the gradient estimation, $N \le K$. In a practical system the algorithm is stopped when it reaches a user-determined MSE level. To apply this conjugate gradient approach to the PBFDAF algorithm the weight actualization equation (8) must be modified as

$$\mathbf{w}[m+1] = \mathbf{w}[m] + \alpha \mathbf{v}[m]. \quad (11)$$

Where $\mathbf{w}$ is the coefficient vector of dimension $MQP \times 1$ which results from rearranging matrix $\mathbf{W}$ (in the notation $\mathbf{w} \leftarrow \mathbf{W}$). $\mathbf{v}$ is a finite $\mathbf{R}$-conjugated vector set which satisfies $\mathbf{v}_i^H \mathbf{R} \mathbf{v}_j = 0, \forall i \neq j$. The $\mathbf{R}$-conjugacy property is useful as the linear independency of the conjugate vector set allows expanding the $\mathbf{w}^{\bullet}$ solution as

$$\mathbf{w}^{\bullet} = \alpha_0 \mathbf{v}_0 + \cdots + \alpha_k \mathbf{v}_k = \sum_{k=0}^{K-1} \alpha_k \mathbf{v}_k. \quad (12)$$

Starting at any point $\mathbf{w}_0$ of the weighting space, we can define $\mathbf{v}_0 = -\mathbf{g}_0$ being $\mathbf{g}_0 \leftarrow \bar{\mathbf{G}}_0$, $\bar{\mathbf{G}}_0 = \nabla(\mathbf{W}_0)$, $\mathbf{p}_0 \leftarrow \bar{\mathbf{P}}_0$, $\bar{\mathbf{P}}_0 = \nabla(\mathbf{W}_0 - \bar{\mathbf{G}}_0)$.

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{v}_k \quad (13)$$

$$\alpha_k = \frac{\mathbf{g}_k^H \mathbf{v}_k}{\mathbf{v}_k^H (\mathbf{g}_k - \mathbf{p}_k)} \quad (14)$$

$$\begin{aligned} \mathbf{g}_{k+1} \leftarrow \bar{\mathbf{G}}_{k+1},\ \bar{\mathbf{G}}_{k+1} = \nabla(\mathbf{W}_{k+1}) \\ \mathbf{p}_{k+1} \leftarrow \bar{\mathbf{P}}_{k+1},\ \bar{\mathbf{P}}_{k+1} = \nabla(\mathbf{W}_{k+1} - \bar{\mathbf{G}}_{k+1}) \end{aligned} \quad (15)$$

$$\mathbf{v}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{v}_k \quad (16)$$

$$\beta_k^{HS} = \frac{\mathbf{g}_{k+1}^H (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{v}_k^H (\mathbf{g}_{k+1} - \mathbf{g}_k)} \quad (17)$$

Where $\mathbf{p}_k$ represents the gradient estimated in $\mathbf{w}_k - \mathbf{g}_k$. For that, it is necessary to evaluate $\mathbf{Y} = \mathbf{X} \otimes \left(\mathbf{W} - \bar{\mathbf{G}} \right)$, (5), (6), (7) and (9). In order to be able to generate nonzero direction vectors which are conjugate to the initial negative gradient vector, a gradient estimation is necessary (Boray and Srinath, 1992). This gradient estimation is obtained by averaging the instantaneous gradient estimates over N past values. The ∇ operator is an averaging gradient estimation with the current weights and N past inputs $\mathbf{X}$ and $\mathbf{d}$,

$$\bar{\mathbf{G}}_k = \nabla \left(\mathbf{W}_k \right) = \frac{2}{N} \sum_{n=0}^{N-1} \mathbf{G}_{k-n} \Big|_{\mathbf{W}_k, \mathbf{X}_{k-n}, \mathbf{d}_{k-n}} . \quad (18)$$

This alternative approach does not require knowing neither the Hessian nor the employment of a linear search. Notice that all the operations (13-17) are vector operations that keep the computational complexity low. The equation (17) is known as the Hestenes-Stiefel method but there are different approaches for calculating β_k: Fletcher-Reeves (19), Polar-Ribière (20) and Dai-Yuan (21) methods.

$$\beta_k^{FR} = \frac{\mathbf{g}_{k+1}^H \mathbf{g}_{k+1}}{\mathbf{g}_k^H \mathbf{g}_k} \quad (19)$$

$$\beta_k^{PR} = \frac{\mathbf{g}_{k+1}^H \left(\mathbf{g}_{k+1} - \mathbf{g}_k \right)}{\mathbf{g}_k^H \mathbf{g}_k} \quad (20)$$

$$\beta_k^{DY} = \frac{\mathbf{g}_{k+1}^H \mathbf{g}_{k+1}}{\mathbf{v}_k^H \left(\mathbf{g}_{k+1} - \mathbf{g}_k \right)} \quad (21)$$

The constant β_k is chosen to provide $\mathbf{R}$-conjugacy for the vector $\mathbf{v}_k$ with respect to the previous direction vectors $\mathbf{v}_{k-1}, \ldots, \mathbf{v}_0$. Instability occurs whenever β_k exceeds unity.

In this approach, the successive directions are not guaranteed to be conjugate to each other, even when one uses the exact value of the gradient at each iteration. To ensure the algorithm stability the gradient can be initialized forcing $\beta_k = 1$ in (16) when $\beta_k > 1$.

4 COMPUTATIONAL COST

Table 1 shows a comparative analysis for both algorithms in terms of operations number (multiplications, sums) clustered by functionality. Note that constants A, B and C, in the PBFDAF computational burden estimation, are used as reference for the number of operations in PBFDAF–CG. For one iteration (k = 1), the computational cost of the PBFDAF–CG is 40 times higher than the PBFDAF.

5 SIMULATION EXAMPLES

MAEC application is a good example of complex system identification because has to deal with very long adaptive filters in order to achieve good results. The scenario employed in our tests simulates two small chambers imitating a typical teleconference environment. Following an acoustic opening approach, both chambers can be acoustically connected by means of linear arrays of microphones and loudspeakers. Details of this configuration follow. Room dimensions are [2000 2440 2700] mm.

The impulse responses are calculated using the image method (Allen and Berkley, 1979) with an expected reverberation time of 70ms (reflection coefficients [0.8 0.8; 0.5 0.5; 0.6 0.6]). The speech source, microphones and loudspeakers are situated as in Figure 3. In the emitting room, the source is located in [1000 500 1000] and the microphones in [{800 900 1000 1100 1200} 2000 750]. Notice that the microphone separation is only 10 cm, which would be a worse case scenario that provides with highly correlated signals. In the reception room the loudspeakers are situated in [{500 750 1000 1250 1500} 100 750] and the microphone in [1000 2000 750].

The directivity patterns of the loudspeakers ([elevation 0º, azimuth -90º, aperture beam 180º]) and the microphones ([0º 90º 180º]) are modified so that they are face to face. We are considering $P = 5$ channels as it is a realistic situation for home applications; enough for obtaining good spatial localization and significantly more complex than the stereo case.

Table 1: Computational Cost Comparative ($O = PQM$).

Alg.\Op.	Gradient Estimation and Convolution	Updating	Constrained Version
PBFDAF	$A = (P+2)O\log_2 O + P(Q(M+1)+1) + K + O$	$B = 9O$	$C = 2O\log_2 O$
PBFDAF–CG	$(((N(A+1)+1)+1)2+1)(k+1)$	$(13O+2)k$	$2CN(k+1)$

The source is a male speech recorded in an anechoic chamber at a sampling rate of 16 kHz and the background noise in the local room has a power of -40 dB of SNR.

Figure 4 shows the constrained PBFDAF algorithm behaviour. For equation (8) we are using a power normalizing expression as

$$\mu[m] = \frac{\mu}{\mathbf{U}[m] + \delta}, \qquad (22)$$

$$\mathbf{U}[m] = (1-\lambda)\mathbf{U}[m-1] + \lambda|\mathbf{X}|^2. \qquad (23)$$

Where $\mu[m]$ is a matrix of dimensions $M \times Q \times P$, μ is the step size, λ is an averaging factor, and δ is a constant to avoid stability problems. In our case $\mu = 0.025$, $\lambda = 0.25$ and $\delta = 0.5$.

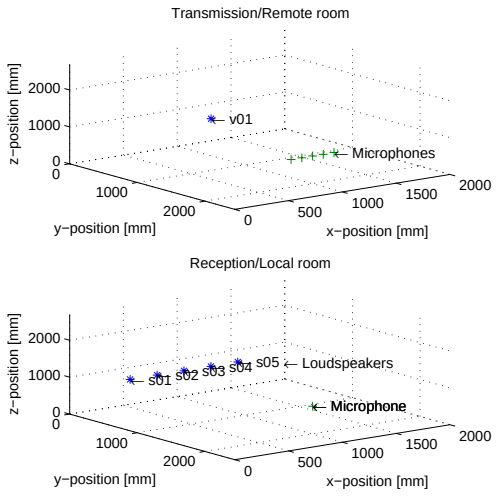

Figure 3: Working environment for the tests.

Figure 5 shows the result of using the PBFDAF-CG algorithm with the Hestenes-Stiefel method where the difference in convergence can be observed. A maximum of $N = \lfloor \sqrt{K} \rfloor$ or when MSE below -45 dB is employed.

For both algorithms we use $Q = 8$ partitions, $L = 1024$ taps, $K = L/Q = 128$ taps for each partition. The length of the FFTs is $M = 2K = 256$. Working with sample rate of 16 kHz means 8 ms of latency (although a delayless approach already has been studied) (Bendel and Burshtein, 2001). Again in both cases the algorithm uses the overlap-save method (50% overlapping).

The upper part of the figures show the echo signal d (black) and the residual error e (grey).

The centre shows the MSE (dB) and the lower picture the misalignment (also in dB) obtained as $\varepsilon = \|\mathbf{h} - \mathbf{w}\| / \|\mathbf{h}\|$, being $\mathbf{h}$ the unknown impulse response and $\mathbf{w} = \left[\mathbf{w}_1^T \ \ldots \ \mathbf{w}_P^T\right]^T$ the estimation.

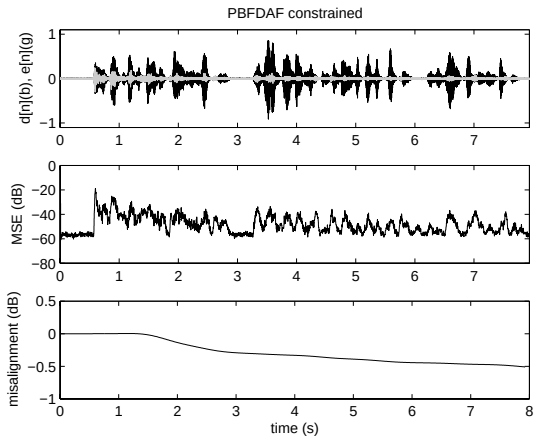

Figure 4: PBFDAF Constrained.

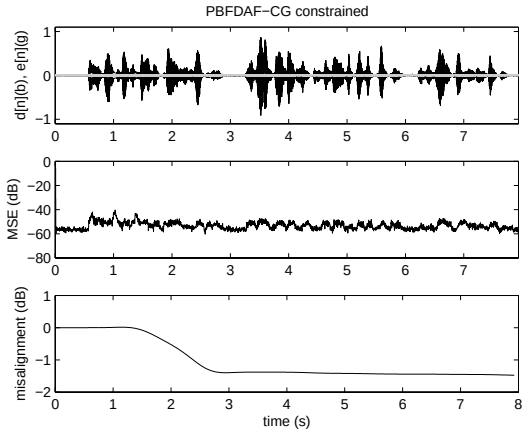

Figure 5: PBFDAF–CG Constrained.

The speech input signal to MAEC application is an inappropriate perturbation signal due to a nonstationary character. The speech waveform contains segments of voiced (quasi-periodic) sounds, such as "e," unvoiced or fricative (noiselike) sounds, such as "g," and silence.

Besides it is possible a double-talk situations (when the speech of the at least two talkers arrives simultaneously at the canceller) that made identification much more problematic than it might appear at first glance.

A much more conditioned application is an adaptive multichannel measure of impulse response. In this case, it is possible to select the best perturbation signal, with the appropriate SNR, for system identification and adapt until the error signal falls below a MSE setting threshold.

The maximum length sequences (MLS) are pseudorandom binary signals which autocorrelation function is approximately an impulse.

In an industrial case it is probably the most convenient method to use because it is simple and allows system identification without perturbing the system operation or stopping the plant (Aguado and Martínez, 2003). In this case it is necessary superimpose the perturbation signal to the input system with a power enough to identify the system while guaranty the optimal functioning.

6 CONCLUSIONS

The PBFDAF algorithm is widely used in multichannel adaptive filtering applications such as MAEC commercial systems with good results (in general for stereo case).

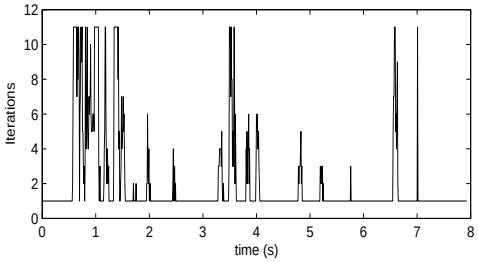

Figure 6: PBFDAF–CG iterations versus time.

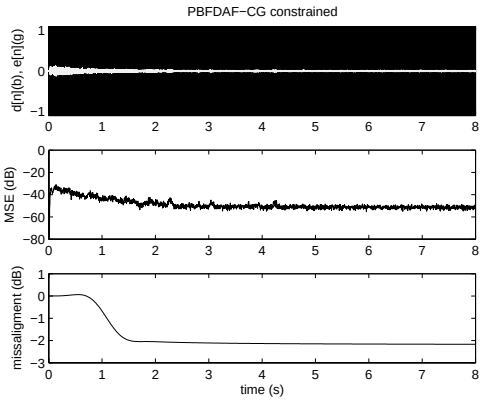

Figure 7: PBFDAF–CG Constrained (MLS).

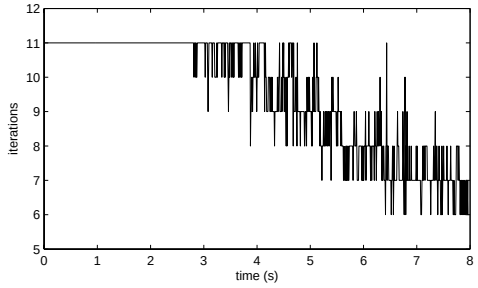

Figure 8: PBFDAF–CG iterations versus time (MLS).

However, especially when working in multichannel, high reverberation environments (like teleconference) its convergence may not be fast enough. In this article we have presented a novel algorithm: PFDAF–CG; based on the same structure, but using much more powerful CG techniques to speed up the convergence time and improve the MSE and misalignment performance.

As shown in the results, the proposed algorithm improves a MSE and misalignment performance, and converges a lot faster than its counterpart while keeping the computational convergence relatively low, because all the operations are performed between vectors in the frequency-domain. We are working on better gradient estimation methods in order to reduce computational cost. Besides, it is possible to arrive to a compromise between

complexity and speed modifying the maximum number of iterations.

Figure 6 shows the PBFDAF–CG iterations versus time. The total number of iterations for this experiment is 992 for PBFDAF and 1927 for PBFDAF–CG (80 times higher computational cost).

Figure 7 shows the result of PBFDAF–CG with MLS source (identical settings) and Figure 8 the iterations versus time. Notice that more uniform MSE convergence and best misalignment. The computational cost decrease while time the increases. A better performance is possible increasing the SNR and diminishing the MSE level threshold.

REFERENCES

Aguado, A., Martínez, M., 2003. *Identificación y Control Adaptativo*, Prentice Hall.

Allen, J.B., Berkley, D.A., 1979. Image method for efficiently simulating small-room acoustics. In *J.A.S.A.*, 65:943-950.

Bendel, Y., Burshtein, D., 2001. Delayless Frequency Domain Acoustic Echo Cancelation. In *IEEE Transactions on Speech and Audio Processing*. 9(5):589-587.

Benesty, J., Huang, Y. (Eds.), 2003. *Adaptive Signal Processing: Applications to Real-World Problems*, Springer.

Boray, G., Srinath, M.D., 1992. Conjugate Gradient Techniques for Adaptive Filtering. In *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Application*. 39(1):1-10.

Luenberger, D.G., 1984. *Introduction to Linear and Nonlinear Programming*, MA: Addison-Wesley, Reading, Mass.

Shink, J., 1992. Frequency-Domain and Multirate Adaptive Filtering. In *IEEE Signal Processing Magazine*. 9(1):15-37.

Páez Borrallo, J., García Otero, M., 1992. On the implementation of a partitioned block frequency-domain adaptive filter (PBFDAF) for long acoustic echo cancellation. In *Signal Processing*. 27:301-315.

APPENDIX

The "conjugacy" relation $\mathbf{v}_i^H \mathbf{R} \mathbf{v}_j = 0, \forall i \neq j$ means that two vectors, $\mathbf{v}_i$ and $\mathbf{v}_j$, are orthogonal with respect to any symmetric positive matrix $\mathbf{R}$. This can be looked upon as a generalization of the orthogonality, for which $\mathbf{R}$ is the unity matrix. The best way to visualize the working of conjugate directions is by comparing the space we are working in with a "stretched" space.

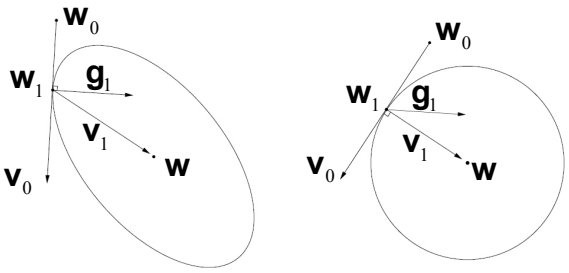

Figure 9: Optimality of CG method.

The SD methods are slow due to the successive gradient orthogonality that results of minimize the recursive updating equation (8) respect to $\mu[m]$. The movement toward a minimum has the zigzag form. The left part in Figure 9 shows the quadratic function contours in a real space (for $\mathbf{r} \neq \mathbf{0}$ in (2) are elliptical). Any pair of vectors that appear perpendicular in this space would be orthogonal. The right part shows the same drawing in a space that is stretched along the eigenvector axes so that the elliptical contours from the left part become circular. Any pair of vectors that appear to be perpendicular in this space is in fact $\mathbf{R}$-orthogonal. The search for a minimum of the quadratic function starts at $\mathbf{w}_0$, and takes a step in the direction $\mathbf{v}_0$ and stops at the point $\mathbf{w}_1$. This is a minimum point along that direction, determined in the same way for SD method. While the SD method would search in the direction $\mathbf{g}_1$, the CG method would chose $\mathbf{v}_1$. In this stretched space, the direction $\mathbf{v}_0$ appears to be a tangent to the now circular contours at the point $\mathbf{w}_1$. Since the next search direction $\mathbf{v}_1$ is constrained to be $\mathbf{R}$-orthogonal to the previous, they will appear perpendicular in this modified space. Hence, $\mathbf{v}_1$ will take us directly to the minimum point of the quadratic function (2nd order in the example).