

THE CONJUGATE GRADIENT PARTITIONED BLOCK FREQUENCY-DOMAIN ADAPTIVE FILTER FOR MULTICHANNEL ACOUSTIC ECHO CANCELLATION

Lino García¹, Jon A. Beracochea², Soledad Torres-Guijarro², and F. Javier Casajús-Quirós²

¹Escuela Superior Politécnica, Universidad Europea de Madrid
28670 Villaviciosa de Odón, Madrid, Spain

phone: + (34) 912115200, email: lino.garcia@uem.es

²Dpto. Señales, Sistemas y Radiocomunicaciones, Universidad Politécnica de Madrid
Ciudad Universitaria, 28040, Madrid, Spain

ABSTRACT

Multichannel acoustic cancellation problem requires working with extremely large impulse responses. Multirate adaptive schemes such as the partitioned block frequency-domain adaptive filter (PBFDAF) are good alternatives and are widely used in commercial echo cancellation systems nowadays. However, being a Least Mean Square (LMS) derived algorithm, the convergence speed may not be fast enough under some circumstances.

In this paper we propose a new scheme which combines frequency-domain adaptive filtering with conjugate gradient techniques in order to speed up the convergence time. The new algorithm (PBFDAF-CG) is developed and its behaviour is compared against previous PBFDAF schemes.

1. INTRODUCTION

The Multichannel Acoustic Echo Cancellation (MAEC) problem has to deal with very long adaptive filters in order to achieve good results. Depending on the environment and its reverberation time, the echo paths can be characterized by FIR filters with thousands of taps.

The so-called Partitioned Block Frequency-Domain Adaptive Filter (PBFDAF) [1] was developed to deal efficiently with such situations. The PBFDAF algorithm is a more efficient implementation of the Least Mean Square (LMS) algorithm in the frequency-domain. It reduces the computational burden and user-bounded delay. 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 to set 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 known frequency-domain adaptive filter (FDAF) for every section of the filter [2].

In general, the PBFDAF algorithm is widely used due to its good trade-off between speed, computational complexity and overall latency. However, when working with long acoustic impulse responses (AIR), 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.

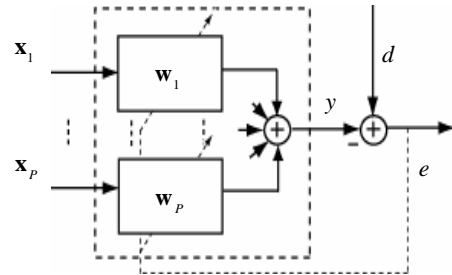


Figure 1 – Multichannel adaptive filtering

Figure 1 shows the working framework where $x_p[n]$, $p = 1, \dots, P$ (P is a number of channels), represents the input signals recorded by the microphones in the remote room, d the echo signal received after convolving with the AIR in the local room, y the output of adaptive filtering, and e the error signal we try to minimize. In typical scenarios, the filter input signals, are highly correlated which further reduces the overall convergence of the filter coefficients w_{pm} , $m = 0, \dots, L-1$ (L is the filter length).

This effect is particularly relevant when dealing with the MAEC problem as all the input signals \mathbf{x}_p come from a single source located in the remote room and the nearness between the microphones (especially when working with microphone arrays) can increase the coherence between channels.

In order to increase the convergence speed we propose a new algorithm which employs much more powerful Conjugate Gradient (CG) optimization techniques, but keeping the frequency block partition strategy to allow computationally realistic low latency situations. The paper is organized as follows: Section II reviews the Multichannel PBFDAF approach and its implementation. Section III develops the Multichannel Conjugate Gradient Partitioned Frequency Domain Adaptive Filter algorithm (PBFDAF-CG). Results of the new approach are presented in Section IV followed by conclusions.

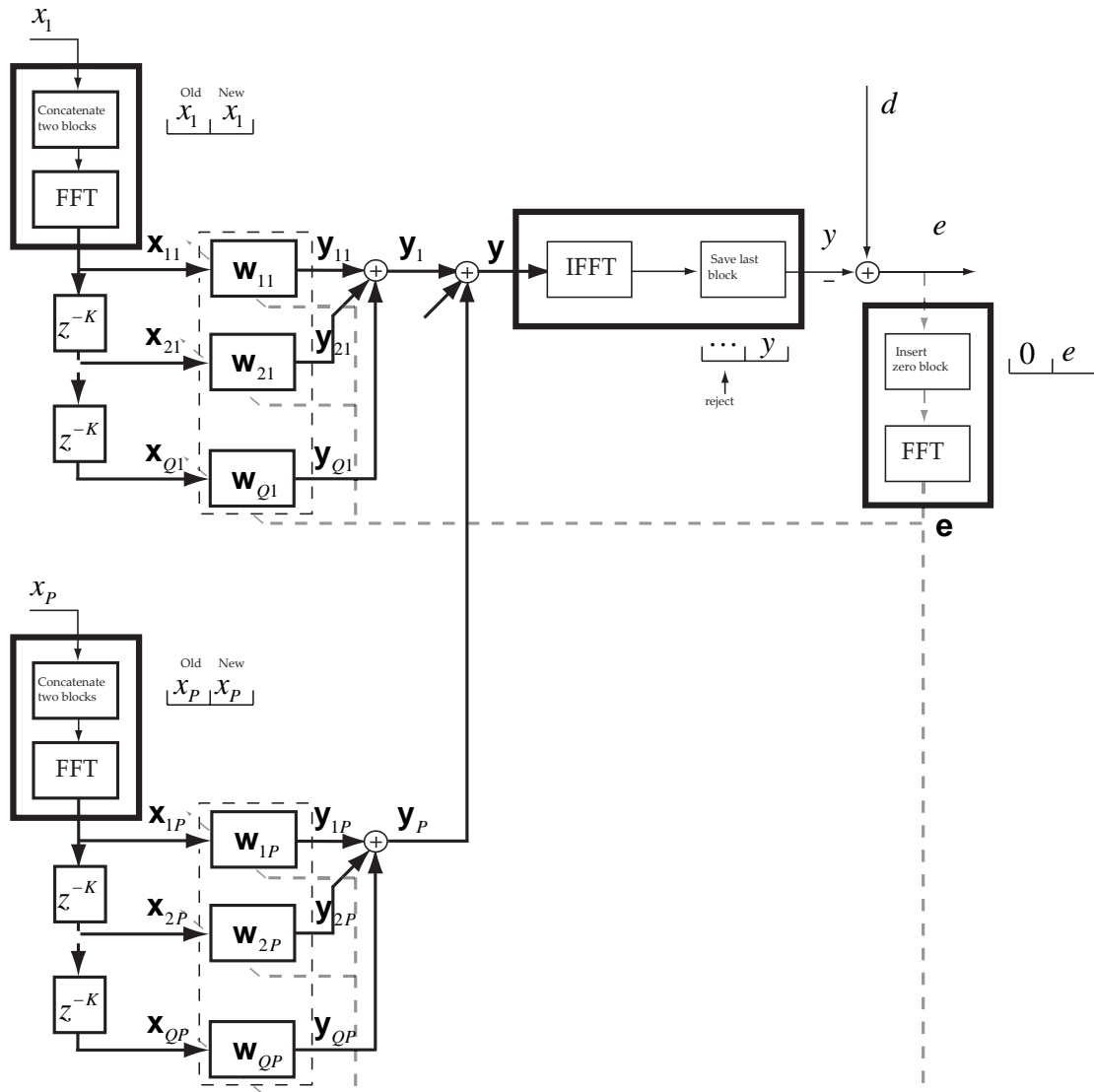


Figure 2 – Multichannel PBFDAF (Overlap-Save method)

2. PARTITIONED BLOCK FREQUENCY-DOMAIN ADAPTIVE FILTERING

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 [2]. This is especially interesting when using long filters as in our case. The frequency domain transform employs one or more discrete Fourier transforms (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 can be seen as a direct frequency domain translation of the Block LMS (BLMS) algorithm

$$y[n] = \sum_{p=1}^P \sum_{m=0}^{L-1} x_p[n-m] w_{pm}. \quad (1)$$

In the PBFDAF case, the filter is partitioned transversally in an equivalent structure. Partitioning \mathbf{w}_p in Q segments (K - length) we obtain

$$y[n] = \sum_{p=1}^P \sum_{q=0}^{Q-1} \sum_{m=0}^{K-1} x_p[n-qK-m] w_{p(qK+m)}. \quad (2)$$

Where the total filter length L , for each channel, is a multiple of the length of each segment $L = QK$, $K \leq L$. 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 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{F} represents the 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 FFTs.

In the frequency domain with matricial notation, equation (2) can be expressed as

$$\mathbf{Y} = \mathbf{X} \otimes \mathbf{W}. \quad (3)$$

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} . 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 (3) 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 , $p = 1, \dots, P$, 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

$$\mathbf{y} = \text{last } K \text{ components of } \mathbf{F}^{-1}\mathbf{y}. \quad (4)$$

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

$$\mathbf{e} = \mathbf{d} - \mathbf{y}, \quad (5)$$

where $\mathbf{d} = [d[mK] \quad \dots \quad d[mK + K - 1]]^T$.

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

$$\mathbf{e} = \mathbf{F} \begin{bmatrix} \mathbf{0}_{K \times 1} \\ \mathbf{e} \end{bmatrix}. \quad (6)$$

As 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 generating an error matrix \mathbf{E} with dimensions $M \times Q \times P$ which contains replicas of the error vector, defined in (6), of dimensions P and Q ($\mathbf{E} \leftarrow \mathbf{e}$ in the notation). The actualization of the weights is performed as

$$\mathbf{W}[m+1] = \mathbf{W}[m] + 2\mu[m] \otimes \mathbf{G}[m]. \quad (7)$$

The instantaneous gradient is estimated as

$$\mathbf{G} = -\mathbf{X}^* \otimes \mathbf{E}. \quad (8)$$

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

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{G} \\ \mathbf{0}_{K \times Q \times P} \end{bmatrix}. \quad (9)$$

3. CONJUGATE GRADIENT PARTITIONED BLOCK FREQUENCY-DOMAIN ADAPTIVE FILTERING

Conjugate Gradient (CG) algorithm is a technique originally developed to minimize quadratic functions which was later adapted for the general case [3]. 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 autocorrelation matrix \mathbf{R} in frequency domain). The mean square error minimization of the multichannel signal with respect to the filter coefficients is equivalent to the Wiener-Hopf equation

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{r}. \quad (10)$$

Where $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^H\}$ represents the autocorrelation matrix

and $\mathbf{r} = E\{\mathbf{x}\mathbf{d}^*\}$ the cross-correlation vector in the time domain.

In each m -block iteration the conjugate gradient algorithm will iterate $k = 1, \dots, \min(N, K)$ times; where

N represent the memory of the gradient estimation, $N \leq K$. In a practical system the algorithm is stopped when it reaches a user-determined mean square error (MSE) level. To apply this conjugate gradient approach to the PBFDAF algorithm the weight actualization equation (7) 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}° solution as

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

Starting at any point \mathbf{w}_0 of the weighting space, we can define

$\mathbf{v}_0 = -\mathbf{g}_0$ being $\mathbf{g}_0 \leftarrow \bar{\mathbf{G}}_0$, $\bar{\mathbf{G}}_0 = \nabla(\mathbf{W}_0)$, $\mathbf{p}_0 \leftarrow \bar{\mathbf{P}}_0$, $\bar{\mathbf{P}}_0 = \nabla(\mathbf{W}_0 - \bar{\mathbf{G}}_0)$

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

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

$$\mathbf{g}_{k+1} \leftarrow \bar{\mathbf{G}}_{k+1}, \bar{\mathbf{G}}_{k+1} = \nabla(\mathbf{W}_{k+1}) \quad (15)$$

$$\mathbf{p}_{k+1} \leftarrow \bar{\mathbf{P}}_{k+1}, \bar{\mathbf{P}}_{k+1} = \nabla(\mathbf{W}_{k+1} - \bar{\mathbf{G}}_{k+1})$$

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

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

Where \mathbf{p}_k represents the gradient estimated in $\mathbf{w}_k - \mathbf{g}_k$.

For that, it is necessary to evaluate $\mathbf{Y} = \mathbf{X} \otimes (\mathbf{W} - \bar{\mathbf{G}})$, (4),

(5), (6) and (8). In order to be able to generate nonzero direction vectors which are conjugate to the initial negative gradient vector, a gradient estimation is necessary [4]. This gradient estimation is obtained by averaging the instantaneous gradient estimates over N past values. The ∇ operator is an averaging gradient estimate with the current weights and N past inputs \mathbf{X} and \mathbf{d} ,

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

This alternative approach does not require knowing neither the Hessian nor the employment of a linear search [4]. 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.

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

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

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

The constant β_k is chosen to provide \mathbf{R} -conjugacy for the vector \mathbf{v}_k with respect to the previous direction vectors $\mathbf{v}_{k-1}, \mathbf{v}_{k-2}, \dots, \mathbf{v}_0$. 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.

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	$((N(A + 1) + 1) + 1)2 + 1)(k + 1)$	$(13O + 2)k$	$2CN(k + 1)$

Table 1 – Computational Cost Comparative ($O = PQM$)

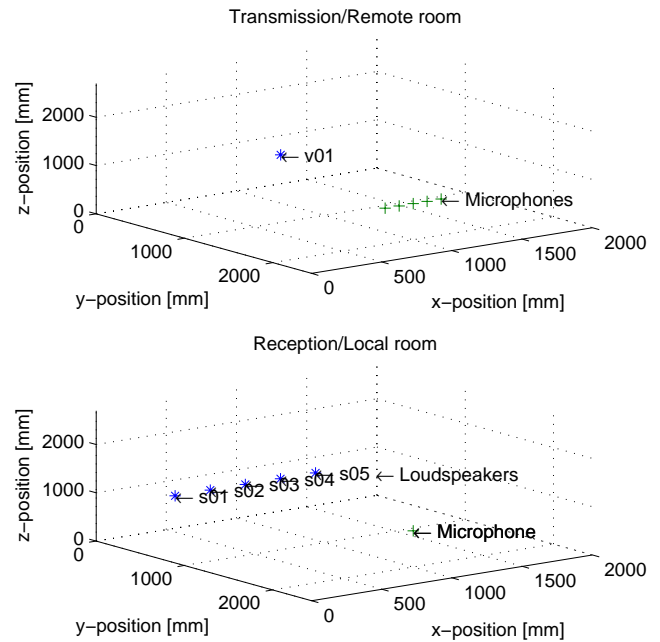


Figure 3 – Working environment for the tests ($P = 5$)

For one iteration ($k = 1$), the computational cost of the PBFDAF-CG is 40 times higher than that of the PBFDAF.

5. SIMULATION EXAMPLES

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 [5] 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 0° , azimuth -90° , aperture beam 180°]) and the microphones ([0° 90° 180°]) 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. 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.

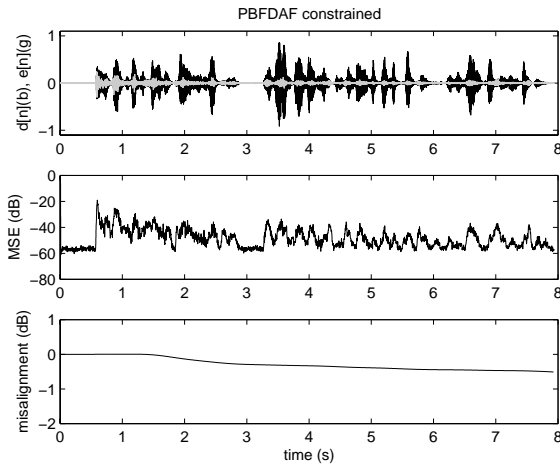


Figure 4 – Constrained PBFDAF

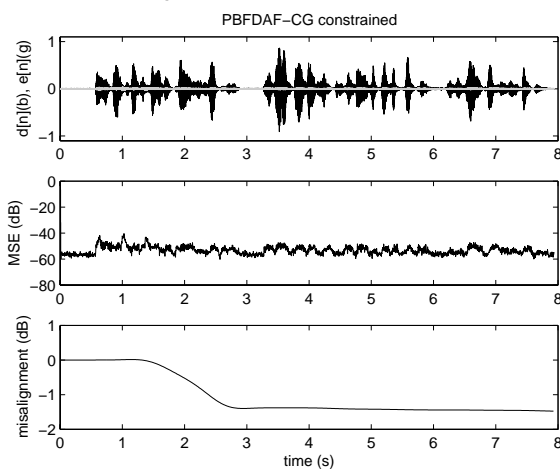


Figure 5 – Constrained PBFDAF-CG

Figure 4 shows the constrained PBFDAF algorithm behaviour. For equation (6) we are using a power normalizing expression as

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

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

Where $\mu[m]$ is a matrix of dimensions $M \times Q \times P$, μ is the step size, β an averaging factor, and δ a constant to avoid stability problems. In our case $\mu = 0,025$, $\beta = 0,25$ and $\delta = 0,5$.

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).

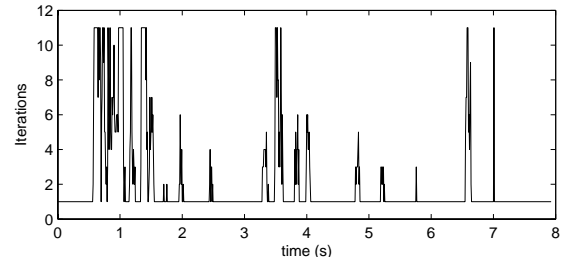


Figure 6 – PBFDAF-CG Iterations

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 Mean Square Error (dB) and the lower picture the misalignment (also in dB) obtained as $\varepsilon = \frac{\|\mathbf{h} - \mathbf{w}\|}{\|\mathbf{h}\|}$, being \mathbf{h} the unknown impulse response and $\mathbf{w} = [\mathbf{w}_1^T \dots \mathbf{w}_p^T]$ the estimation. Figure 6 shows the k PBFDAF-CG iterations versus time. The total number of iterations for this experiment is 992 for PBFDAF and 1927 for PBFDAF (80 times higher computational cost).

6. CONCLUSIONS

The PBFDAF algorithm is widely used in commercial AEC systems with good results. 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 based the same structure, but using much more powerful Conjugate Gradient techniques to speed up the convergence time and improve the mean square error and misalignment performance. As shown in the results, the proposed algorithm converges a lot faster than its counterpart while keeping the computational burden relatively low, as all the operations are performed between vectors in the frequency domain, although 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.

REFERENCES

- [1] J. Páez Borrillo, and M. García Otero, "On the implementation of a partitioned block frequency-domain adaptive filter (PBFDAF) for long acoustic echo cancellation," *Signal Processing*, 27:301-315, June 1992.
- [2] J. Shink, "Frequency-Domain and Multirate Adaptive Filtering," *IEEE Signal Processing Magazine*, 9(1):15-37, January 1992.
- [3] D. G. Luenberger, *Introduction to Linear and Nonlinear-Programming*. MA: Addison-Wesley, Reading, Mass., 1984.
- [4] G. Boray, and M.D. Srinath, "Conjugate Gradient Techniques for Adaptive Filtering," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 39(1): 1-10, January 1992.
- [5] J.B. Allen, and D.A. Berkley, "Image method for efficiently simulating small-room acoustics," *J.A.S.A.*, 65:943-950, Apr 1979.