On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation

Article *in* Signal Processing · June 1992

DOI: 10.1016/0165-1684(92)90077-A

CITATIONS
38

READS
1,353

2 authors, including:

José M Paez Borrallo
Tecnológico de Monterrey
124 PUBLICATIONS **711** CITATIONS

SEE PROFILE

Signal Processing 27 (1992) 301–315
Elsevier

On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation*

José M. Páez Borrallo and Mariano García Otero

Departamento Señales, Sistemas y Radiocomunicaciones, ETSIT, Universidad Politécnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain

Received 20 December 1991

Abstract. The growing demand of communication systems incorporating hands-free audioterminals has multiplied the efforts of developing efficient and reduced size acoustic echo cancellers. This work deals with the implementation of a wideband acoustic echo canceller using only a single DSP chip. The complete implementation takes advantage of two interesting features of a partitioned block frequency technique for the filtering operation: its good computational burden, and reduced and user-bounded delay. Essentially, this technique makes a sequential partition of the impulse response in the time domain prior to a frequency domain implementation of the filtering operation. Furthermore, this time segmentation allows to set up individual coefficient updating strategies concerning to different sections of the adaptive canceller, thus avoiding the need of disabling the adaption in the complete filter. The adaptive algorithm is based on the known frequency domain adaptive filter (FDAF) for every section of the filter, but the adaptation step computation procedure differs from other usual ones because it introduces an additional measure of the input spectral uniformity. Finally, a real time prototype is implemented with the help of a commercial floating-point DSP board. It is able to satisfy the needs of any wideband hands-free audioterminal working in reverberant environments with acoustic echoes of length up to 200 ms (≈3000 filter coefficients).

Zusammenfassung. Die wachsende Nachfrage nach Kommunikationssystemen mit Freisprech-Audioterminals hat zu vermehrten Anstrengungen bei der Entwicklung von leistungsfähigen und aufwandreduzierten akustischen Echokompensatoren geführt. Diese Arbeit behandelt die Implementierung eines breitbandigen akustischen Echokompensators unter Verwendung nur eines DSP-Chips. Die gesamte Implementierung nutzt zwei interessante Eigenschaften der Partitioned-Block-Frequency-Technik für die Filteroperation aus: den günstigen Rechenaufwand und die reduzierte und anwendungsangepaßte Verzögerung. Diese Technik besteht im wesentlichen aus einer sequentiellen Aufteilung der Impulsantwort im Zeitbereich vor der Implementierung der Filteroperation im Frequenzbereich. Ferner erlaubt diese Segmentierung im Zeitbereich individuelle Strategien für den Koeffizientenabgleich in den verschiedenen Abschnitten des adaptiven Kompensators, wodurch nicht mehr die Adaption im gesamten Filter unterbrochen werden muß. Der adaptive Algorithmus für die einzelnen Abschnitte des Filters beruht auf dem bekannten Frequency Domain Adaptive Filter (FDAF), jedoch unterscheidst sich die Berechnung der Adaptionsschrittweite vom üblichen Verfahren durch Einführung eines zusätzlichen Maßes für die spektrale Gleichförmigkeit des Eingangssignals. Mit Hilfe einer kommerziellen floating-point DSP-Karte wurde ein Echtzelt-Prototyp implementiert. Er erfüllt die Anforderungen von breitbandigen Freisprech-Audioterminals für hallige Umgebungen mit Längen der akustischen Echos bis zu 200 ms (≈3000 Filterkoeffizienten).

Résumé. La demande croissante de systèmes de communications incorporant des terminaux mains-libres a stimulé les efforts de développement d'annuleurs d'échos acoustiques efficaces et d'encombrement réduit. Le présent article décrit la mise en oeuvre d'un annuleur d'échos acoustiques à bande large sur un circuit processeur de signal unique. Deux propriétés intéressantes d'une technique fréquentielle à blocs partitionés sont mises à profit dans l'opération de filtrage: la bonne efficacité de calcul et le faible retard, que l'utilisateur peut limiter. Dans cette approche, on effectue une partition séquentielle de la réponse impulsionnelle du filtre avant l'implantation dans le domaine des fréquences. De plus, cette segmentation temporelle permet de mettre à jour les coefficients des différentes sections de l'annuleur adaptatif suivant des stratégies différentes, en évitant ainsi de bloquer l'adaptation de l'ensemble du filtre. L'algorithme est basé sur le principe du filtrage adaptatif dans le domaine des fréquences pour chaque section, mais le calcul du pas d'adaptation se fait en introduisant une mesure additionnelle de l'uniformité spectrale de l'entrée. Finalement, un prototype en temps réel a été réalisé à l'aide d'une carte processeur flottant du commerce. Il permet de satisfaire aux besoins de tout type de terminal mains-libre à bande large fonctionnant dans un environnement réverbérant avec des échos acoustiques de longueur pouvant atteindre 200 ms (≈3000 coefficients).

Keywords. Frequency domain adaptive filter, echo cancellation, hands-free, single DSP implementation.

* Work supported by ALCATEL SESA, C/Ramírez de Prado 5, 28045 Madrid, Spain.

1. Introduction

Nowadays, the high capacity and power of the most recent DSP hardware allows some engineering fields, specially those related to audio-frequency signals, to advance in a real and spectacular way. Acoustic echo control is a field that follows that growth as well, since, as is notoriously manifest, in the past two or three years the efforts to set up real-time and economically feasible acoustic echo cancellers have been multiplied [6, 11–13].

Acoustic echo control arises as a real need in hands-free telephony or teleconference communication systems. The presence of any acoustic echo of our voice in the incoming far-end signal produces an undesired effect in natural half-duplex communications and an important loss of intelligibility and quality in full-duplex ones. Whereas in half-duplex this problem may be simply solved by switching off the return channel, in full-duplex it has to remain open. To alleviate this problem, the present wide-band audioterminals try to incorporate, among other classical techniques like variable loss insertions, an additional solution based on the implementation of a parallel adaptive filter to the acoustic path. The goal is to provide an echo replica on its output to cancel locally the acoustic echo. However, due to the duration of a typical channel impulse response for a reverberant room (from 0.1 to 0.5 seconds), the adaptive filter length may even reach 8000 taps at a sampling rate of 16 kHz. Obviously, this fact makes difficult its real-time implementation when using typical filtering schemes.

The present work deals with the implementation of a complete acoustic echo canceller for wide-band hands-free telephony on a commercial prototyping board. It contains a single floating point DSP chip plus the signal acquisition part (A/D and D/A converters and filters). The board is specially suited to develop any DSP algorithm on an AT station (see Fig. 1). The basic connection between the board and the AT station to properly develop our system is depicted in Fig. 2. The objective is to obtain a relatively flexible cancellation scheme capable of being implemented on a single digital signal processor. Additionally it has to be able to provide an appropriate acoustic echo return loss (AERL), thus reducing the cost of 'domestic' hands-free audioterminals as much as possible.

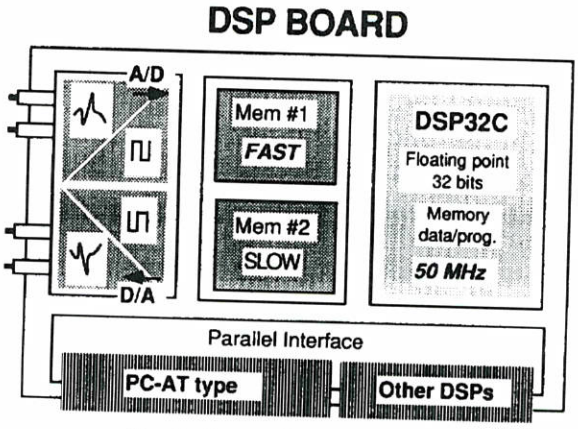
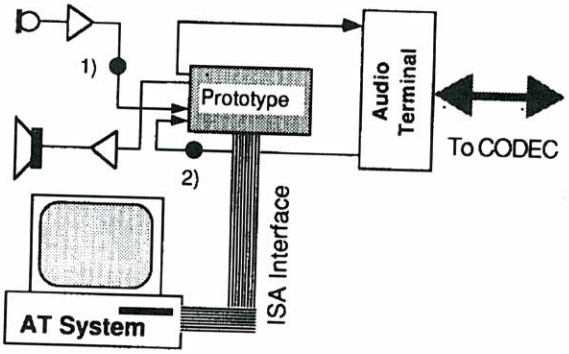


Fig. 1. Prototyping DSP board.



1) and 2):
Also data from internal disk system

Fig. 2. Hardware connection scheme.

The software developed for our canceller consists of the following main parts: (a) the adaptive filter that tries to get an echo replica to cancel the real one; (b) the adaptive algorithm that modifies the filter parameters to minimize a residual echo cost function; (c) the talking monitor that detects the voice activity periods in the transmitting and receiving lines to enable (or disable) other internal functions; (d) the adaptation step computation box that offers the best current adaptation step to the adaptive algorithm in each iteration; and (e)

the loss insertion control that reinforces the performance of the echo canceller by inserting predefined losses in the receiving and transmitting paths. Figure 3 shows a sketch of the system with the interconnections of these parts.

Focusing our attention on the main part of the system, that is, the adaptive filter, the number of arithmetic operations required for a time domain filtering process on a long FIR structure is so large that other filtering techniques (or non-transversal filter structures) have to be considered for a real-time implementation of such a filter. An evident alternative is the use of frequency domain techniques to realize the filtering operation [4]. But, in the case of a long filter, the unavoidable previous data gathering introduces long delays in the natural speech conversation that cannot be tolerated.

To reach a trade off between a real-time implementation on a single DSP chip and an admissible delay, we use a simplified and mixed version of the partitioned block frequency domain adaptive filter (PBFDAF) [2, 9, 10]. This procedure combines frequency and time transversal structures to implement rapidly the linear time convolution. It allows one to consider the total length (or nearly the total) of the channel impulse response without paying for a high computational burden or inadmissible delay in the filtering operation. This filtering technique makes use of appropriate data sectioning procedures, like, in our case, the *overlap-save* method, to carry out the linear convolution in the frequency domain.

The adaptation process makes use of a maskable, normalized and complex LMS algorithm, per frequency bin and filter section, with the spectral adaptation step matrix computed directly from the current data. However, due to the statistics of speech signals and the bad convergence properties of the LMS algorithm with highly colored, nonstationary signals and double-talk periods, it is also necessary to track the signal activity in the transmitting and receiving paths to have a control information that allows one to freeze the adaptation process within such inactivities or double talk periods. Additionally, to improve the adaptation step computation, we introduce in our system a measure that gives information about the spectral uniformity of the incoming signal. This spectral information is used, among other things, to compensate the existing deviations of the variance of the spectral estimator for variably colored signals. This consideration is important because of the intrinsic statistic differences between voiced and unvoiced sounds, or even the presence of a strongly colored noise. This spectral indication allows one to readjust some components of the adaptation step vector in a conservative way.

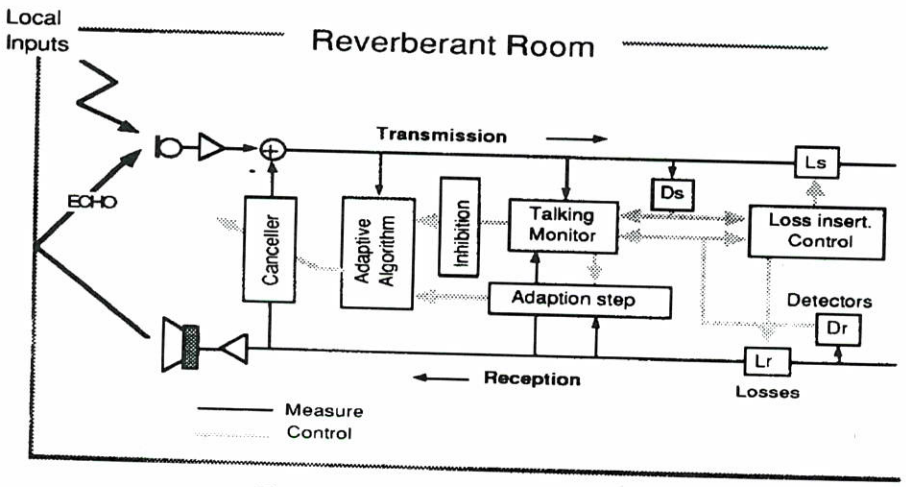


Fig. 3. Diagram of the complete system.

The signal activity information is also of interest to insert variable losses in the reception and transmission paths. This operation reinforces the echo cancellation effect in like half-duplex communication periods and decreases the risk of *howling* in the conversation loop. The total loss inserted in the loop is a fixed amount (12 dB), but the individual losses are shared between both paths depending on the current voice activities.

The masking or inhibition strategy in the adaptation process may be guided entirely by the activity monitoring results. However, because of any hardware or time limitation in the system, we can additionally set a rotative updating strategy, or even any other type, to adapt only those filter sections that we currently consider relevant because of their cancellation effect or the recent inhibition history.

Finally, the whole echo canceller is implemented on a commercial board that plugs into a PC-AT slot for software development purposes. It consists basically of the following parts: (a) an AT&T DSP32C processor at 50 MHz supporting 32-bit floating-point arithmetic; (b) two external memory banks (zero and two wait states); (c) two high precision (16 bits) analog I/O channels with selectable sample rate up to 140 kHz.

The developing and monitoring tools have been provided by AT&T and Loughborough Sound Images Ltd (LSI) [5, 8].

2. System parts

2.1. *The partitioned block frequency filtering process*

Assuming a filter with a long impulse response $h(n)$, the filtering operation usually implies either a high computational burden if it implements the time-domain linear convolution or an intolerable delay because of the data gathering needed for a frequency domain implementation. Thus, both options make the implementation of the filtering operation in real time applications a difficult subject. Then, to reach an intermediate feasible and balanced solution, $h(n)$ can be sectioned in L adjacent, equal length and non-overlapping sections as

$$h(n)=\sum_{l=0}^{L-1} h_l(n), \tag{2.1}$$

where for every section, $h_l(n)=h(n)$ for $n=lN,\ldots,lN+N-1$ and zero elsewhere. This time partition of the filter can be seen, as opposite to the well known subband approach, as a bank of L parallel filters working with the full spectrum of the input signal (see Fig. 4). That is, the complete impulse response is partitioned in a transversal way and disposed as an equivalent parallel structure.

In fact, since the filter output $y(n)$ is mathematically obtained convolving $x(n)*h(n)$, we can get $y(n)$ as

$$\begin{aligned} y(n)&=x(n)*\sum_{l=0}^{L-1} h_l(n)=\sum_{l=0}^{L-1} x(n)*h_l(n)\\ &=\sum_{l=0}^{L-1} x(n-lN)*h_l(n+lN)=\sum_{l=0}^{L-1} y_l(n), \end{aligned} \tag{2.2}$$

that is, the sum of the outputs of L parallel N-tap filters with delayed inputs.

Now, using an appropriate data sectioning procedure, the L linear convolutions of the parallel filters can be independently carried out in the frequency domain with only a total delay of N samples, instead of the NL samples needed in a standard frequency domain implementation. See Fig. 5.

Here, the boxes S_{in} and S_{out} represent the necessary signal segmentation for the implementation of the fast convolution. These techniques need to enlarge the input vector to avoid undesired overlapping effects and to assure a mathematical equivalence with the time domain linear convolution. In our case, we will use the same length for the input vector and filter section: N samples. Thus, after sectioning the input signal in a $2N$ data block (every N samples) and taking its corresponding FFT, the resulting frequency block is stacked, in a transversal way (FIFO memory), at a rate of N samples. Within these blocks the filtering operation is performed directly in the frequency domain

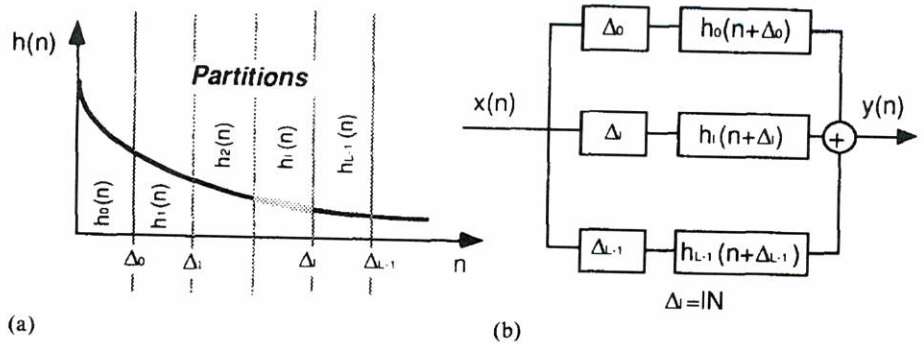


Fig. 4. (a) Impulse response partitions; (b) parallel implementation.

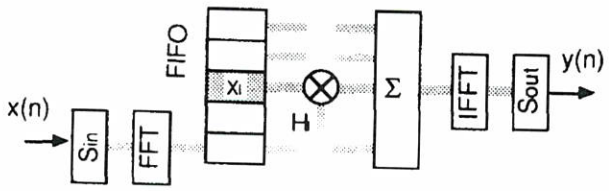


Fig. 5. Partitioned block frequency filtering schemes.

by simply multiplying all the frequency components of the current delayed FFT input by those of the corresponding FFT section of the filter. Finally, to obtain the equivalent time output, the $2N$ frequency bins of every section filter are summed up and fed to the IFFT processor and the output sectioner. Notice here that we have altered the order of the final two operations, sum and IFFT, since

$$\sum_{l=0}^{L-1} \mathrm{IFFT}[\boldsymbol{X}^l(j-l)\boldsymbol{H}^l] = \mathrm{IFFT}\left[\sum_{l=0}^{L-1} \boldsymbol{X}^l(j-l)\boldsymbol{H}^l\right],$$

thus saving $N-1$ FFT operations in the complete filtering process (j represents the time index and l the current section of the filter).

To conclude, if we use the standard time domain technique, there are LN^2 necessary operations to filter N real samples, whereas the proposed partioned block frequency technique needs only about $N(4L+3\log_2 2N)$ operations for the same process, that is, it results in a computational saving of $(1-(4L+3\log_2 2N)/LN)\times 100\%$ (>90% for $L>4$ and $N>128$). See Fig. 6.

2.2. *The adaptive algorithm*

Since the filtering operation is carried out in the frequency domain, we can exploit the fact of a multiband separation and perform directly here the adaptations. Also, because of the implicit input signal decorrelation provided by the FFTs, we can take advantage of the good convergence properties of the frequency domain adaptive filter (FDAF) when using an appropriate adaptation step vector. This technique uses a complex and normalized LMS algorithm for every considered frequency bin. Therefore, in a stationary situation, the global convergence can be made nearly independent of the

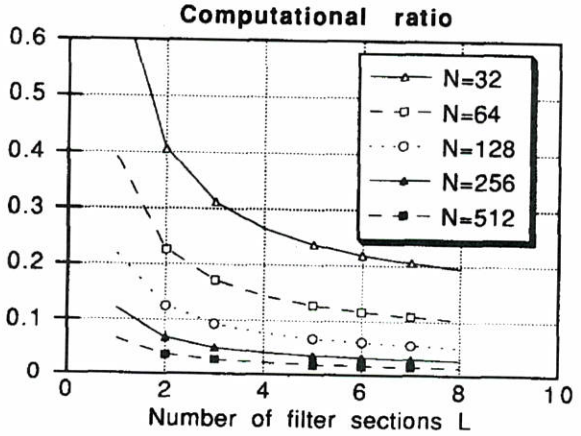


Fig. 6. Computational burden ratio between time domain linear convolution and partitioned block scheme.

input signal correlation (at least asymptotically speaking) by simple power normalization of each component of the adaptation step vector with respect to its associated frequency bin. In our case, we have adopted the following adaptive algorithm for an arbitrary filter section l ($l=0,\ldots,L-1$):

$$H_i^l(j+1)=H_i^l(j)+\mathrm{Inh}(l)\mu_i^l(j)E_i(j)X_i^*(j-l+1) \quad \text{for } i=l,\ldots,N+1, \tag{2.3}$$

where E_i is the corresponding error frequency bin (we only consider the first half plus one bins, $N+1$, by symmetry), the asterisk *, in the FFT of the input X_i, denotes complex conjugation and μ_i is the corresponding adaptation step for the frequency bin i. Here j indexes the current iteration time and i means frequency bin. The inhibition factor $\mathrm{Inh}(l)$ can take values 0 or 1, allowing one to freeze the adaptation process in the considered filter section. Also, the fact of using a partitioned block technique and therefore a small FFT size allows one to update the coefficients more often, thus enhancing its tracking capabilities.

This algorithm would resemble the known FDAF algorithm [4] when $\mathrm{Inh}(l)\mu_i^l(j)=\text{constant}$ and the correcting term is the correct gradient projection $\mathrm{Prj}[E_i(ju)X_i^*(j-l+1)]$. As is shown in [4], the use of data sectioning in the filtering operation extends the original record lengths ($N\to N'$, $N'=2N$ in our case). It means that $2N$ frequency coefficients are necessary for the filter and, therefore, by using the IFFT operation, it yields the corresponding $2N$ time coefficients. Since every section of the filter is uniquely defined with its original first N coefficients, a zero setting mechanism has to be implemented to bring the remaining N coefficients to zero before the current adaptation is carried out. This is the task of the projection operation, which, in the case of using an overlap-save method for the data sectioning procedure, implements the three steps depicted in Fig. 7.

Nevertheless, as can be seen, this operation introduces the cost of two additional FFTs in the computational burden. To make the adaptive algorithm more computationally efficient, we do not impose the time weight constraint as the unconstrained frequency least mean squares [7] (UFLMS) does. This algorithm converges to the Wiener solution (under some favorable conditions), although it may certainly slow down the convergence. It ignores the projection of the gradient and uses directly the product **EX***. This simplification leads to a saving of two FFTs per block iteration. It means that the complete adaptive algorithm only makes use of three $2N$-point FFTs per block iteration. See Fig. 8.

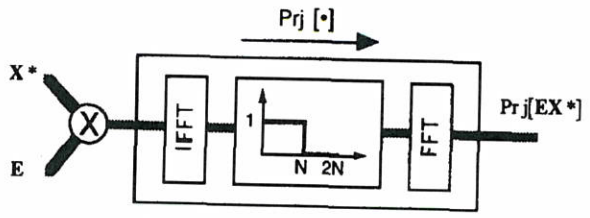


Fig. 7. The projection operation.

Here one can also observe a different segmentation for the current residual echo $e(n)$. This is built by replacing the N last terms of the previous residual echo by an N-zeros vector, that is, $e=[0,\ldots,0,e(j-N+1),\ldots,e(j)]^T$. This simplification in the segmentation saves N positions of memory and gives all the innovation information in the current new N-terms of the residual echo.

2.3. *The inhibition vector*

This binary vector plays the role of freezing the coefficient adaptation in a group of predetermined filter sections. The inhibition strategy can be set according to two different considerations. The first and most obvious one is that the adaptation process has to be disabled during local talking periods and in the absence of a driving signal (far end signal). The local talking would mask the right echo and the loss of driving signal would lead to a wrong step-size computation. The second consideration is mainly related to hardware limitations, that is, computational power and available memory. It means that, depending on how many sections (out of L) can be really updated in each

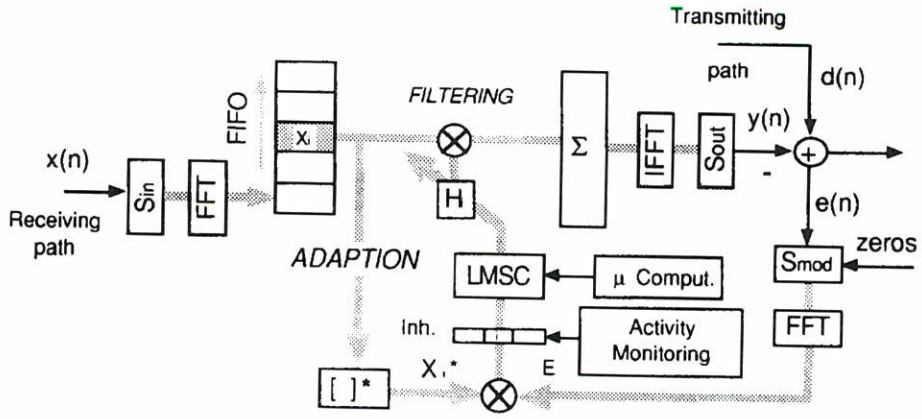


Fig. 8. Adaptive canceller block diagram.

iteration, we can select a rotative updating strategy, or any other, to adapt only those sections that we consider relevant in the total echo. This allows one, in some applications, to have an internal full echo path replica (2000, 4000 or 8000 taps) and to update only a part of it in each block iteration. For instance, assuming that the echo is time-invariant most of the time, we could update in every iteration the first filter sections (they concentrate more echo power) and just a reduced and selected group of the last ones. This strategy has to be always conditioned to the outputs of the activity monitor in order to adequate the inhibition actions to the current and dynamic situations of double talking and driving signal.

2.4. *The adaptation step computation*

The adaptation step vector is basically computed according to the inverse profile of the input spectral power [7]. Since (2.3) presents a transversal structure for each frequency bin, any individual adaptation step has to be bounded by $\mu_i^l(j) < 1/\mathrm{tr}(\mathrm{E}\{[\boldsymbol{X}_i^{\mathrm{T}}(j)\boldsymbol{X}_i^*(j)]\})$, $l=0,\ldots,L-1$. However, because we are implementing a block transversal filter and speech signals are nonstationary and highly colored, some modifications should be carried out in order to set up the on-line working adaptation step vector for each filter section.

First, we have considered the different spectral distributions and dynamic ranges of voiced and unvoiced sounds, or even an additive strong low-pass background noise, to bound properly the maximum allowable step-size for every frequency bin. That is, to avoid any possible misconvergence, we fix an upper bound for all the components of the step-size vector. It depends on a combined estimate of the total current power and whiteness of the spectral profile. It is computed as

$$\mu_{\max}(j) = \frac{c_1}{\mathrm{Sp}(j) \sum_{i=1}^{N+1} \sum_{l=0}^{L-1} |X_i(j-l)|^2}, \qquad (2.4)$$

where c_1 is a constant and $\mathrm{Sp}(j)$ is the spectral indicator. It is given by

$$\mathrm{Sp}(j) = [1-k_1^2(j)][1-k_2^2(j)], \qquad (2.5)$$

where k_1 and k_2 are the first two PARCORs computed directly from the current first block of signal. This correction factor avoids the bound being too small for those cases of sharp spectral distributions with dominant spectral maxima (or large spectral dynamic ranges). It tries to compensate the bound for the least relevant frequency bins.

In a second phase, we compute the $N+1$ spectral dependent components of the step-size vector and compare their values with the upper bound. Then, the initial vector for the step-size is given as

$$\mu_i(j)=\left[\frac{1}{1-c_2\log(\mathrm{Sp}(j))}\right]\min\left[\mu_{\max}(j),\frac{1}{P_i(j)}\right], \tag{2.6}$$

where we take for every bin i, $i=0,\ldots,N+1$, the minimum value between the bound and its corresponding inverse of spectral power (P_i). It avoids irrelevant frequency bands leading to an uncontrolled step-size and, therefore, a local divergence of the algorithm. The global factor $(0<1/\{1-c_2\log[\mathrm{Sp}(j)]\}<1)$ is again a spectral dependent weight that compensates the quality of spectral power estimates in the cases of colored spectral profiles. It is a conservative factor, since in those cases of spectral estimates with large deviations (peaks of the periodogram), the risk of a large step-size estimate also increases. Hence, this weight shrinks the complete step-size vector to avoid any possible local divergence. This factor is based on a monotonic function of Sp, a rough measure of the uniformity of the spectral distribution. Here, c_2 is a constant to adjust.

Additionally, we implement a monitoring function to track the evolution of the residual echo relative power. This measure is used to reduce conveniently the step-size vector during the convergence period. It allows the error misadjustment to converge to a smaller steady state value. Also, this function is oriented to give information on large and instantaneous residual echo variations, which allows one to reset the decreasing step-size mechanism. The monitor measures the current echo return loss enhancement (ERLE) defined by

$$\mathrm{erle}(j)=\frac{[\boldsymbol{d}(j)-\boldsymbol{y}(j)]^{\mathrm{T}}[\boldsymbol{d}(j)-\boldsymbol{y}(j)]}{\boldsymbol{d}^{\mathrm{T}}(j)\boldsymbol{d}(j)}, \tag{2.7}$$

where $\boldsymbol{d}$ and $\boldsymbol{y}$ are the current blocks for the local echo and its replica, respectively. This measure is conveniently smoothed and filtered using the following expressions:

$$S(j)=\frac{1}{1-c_3\log[\mathrm{erle}(j)]}, \tag{2.8}$$

$$F_\mu(j)=c_4F_\mu(j-1)+(1-c_4)S(j), \tag{2.9}$$

with c_3 and c_4 two more constants to adjust. Expression (2.8) uses the same monotonic smoothing function used in (2.6) with Sp. Figure 9 shows the shape of that monotonic function for different constant values.

Finally, we weight exponentially the computed step-size vector in the transversal (time) direction. This operation is necessary since any residual frequency echo $E_i(j)$ is composed of L transversal components, that is, from Fig. 8,

$$E_i(j)=\mathrm{FFT}_i\left\{S_{\mathrm{mod}}\left[\boldsymbol{d}(j)-S_{\mathrm{out}}\left(\underset{i=1,\ldots,2N}{\mathrm{IFFT}}\left[\sum_{l=0}^{L-1}H_i^l(j)X_i(j-l+1)\right]\right)\right]\right\},$$
$$i=1,,\ldots,N+1, \tag{2.10}$$

where S_{mod} and S_{out} are segmentation operators. Equivalently,

$$E_i(j)=D_i^*(j)-\sum_{l=0}^{L-1}H_i^{l*}(j)X_i^*(j-l+1)$$
$$=\sum_{l=0}^{L-1}E_i^{l*}(j),\quad i=1,\ldots,N+1, \tag{2.11}$$

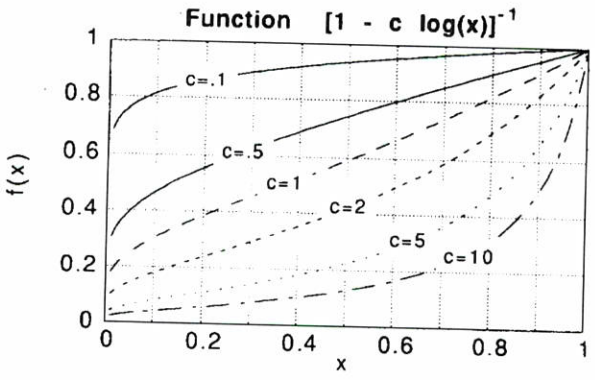


Fig. 9. Smoothing function for Sp and erle.

where the symbol • in the superindices of (2.11) concentrates the results of the sectioning and FFTing operations. Then, using (2.11) expression (2.3) is expanded as

$$H_i^l(j+1) = H_i^l(j) + \mathrm{Inh}(l)\mu_i^l(j)E_i^{l\bullet}(j) \times X_i^*(j-l+1) + \mathrm{Inh}(l)\mu_i^l(j)\left[\sum_{\substack{m=0 \\ m\neq l}} E_i^{m\bullet}(j)\right] \times X_i^*(j-l+1), \quad (2.12)$$

where the third element of the sum acts as a noise in the adaptation of any frequency bin of the considered filter section l. Regarding now any echo impulse response profile, the most energetic part is just concentrated on its beginning. It means that, for every frequency bin, the residual echo components, $E_i^{l\bullet}$, of the first filter sections are more relevant than those corresponding to the last ones, or in other words, the last sections of our adaptive filter suffer from a more significant source of noise than the first ones. To avoid this masking effect in the error reference signal, the step-size vector should be progressively attenuated in the time direction. The goal is to minimize the effect of large additive noises in the coefficient updating of the latter and the less significant filter sections. The time-weighting is carried out with the factor γ^l $(0<\gamma<1)$, where

$$\gamma = 0.8 + c_5 \log(\mathrm{Sp}), \quad 0<\gamma<1, \quad (2.13)$$

c_5 ($\ll 1$) is a constant and Sp is again the spectral indicator.

Then, considering the different points carried out in the computation of the adaptation step vector, this is finally formed using the following expression:

$$\mu_i^l(j) = \gamma^l(j)F_\mu(j)\mu_i(j), \quad i=1,\ldots,N+1,\ l=0,\ldots,L-1, \quad (2.14)$$

an individual component that depends on the considered frequency bin and filter section.

2.5. The talking activity monitor

To allow the adaptive algorithm to work properly, we must take care of two causes of misconvergence: the acoustic echo should not be contaminated with any large additive interference, especially local talking, and the adaptive filter has to be correctly driven, that is, some significant amount of signal has to be present. The first one affects directly the correct error reference for the adaptive algorithm and the second one affects the adaptation step computation in any NLMS-type algorithm.

To detect those events, we have implemented an activity talking box with the following two tasks: (a) detection of far-end signal activity and (b) detection of local signal activity and double-talking activity. To detect the signal activity on any single path, we use a mixed measure based both on the local total energy and on the spectral energy distribution. However, to discriminate the double-talking from a large echo, we use a combined measure based on the short-term correlations of the envelopes of the signals in the receiving and sending lines. This procedure seems to be very efficient and provides good scores for the probability of detection of double-talking periods, although in some cases it can lead to a moderate false alarm rate. Also, another measure based in the zero-crossings rate can be combined with the previous ones to improve the final performance in those cases of insignificant background low-pass noise. Finally, once the detectors deliver their results, the talking activity information is used to set adequately the inhibition vector and to control the insertion of variable losses in the sending and/or receiving lines. The maximum loss is limited to 12 dB.

Denoting $\boldsymbol{Ax}$ and $\boldsymbol{As}$ as two logical variables that represent the incoming signal and double-talking activities, respectively, the actions considered by the system depending on their values are depicted in Fig. 10 (the overbar means NOT operation).

In the sequel, we will describe in greater detail the two components of the talking monitor, namely the signal activity and the double-talking detectors.

CAUSES	ACTIONS
$\overline{Ax}$	Freeze Adap. step computations
$\overline{Ax}$ OR As	Freeze Adaptions
As OR $\overline{As}$	Insert Loss in Tx 1)
$\overline{Ax}$ OR As	Insert Loss in Rx

1) Different losses

Fig. 10. Actions derived from the talking activity analysis.

The signal activity detector

The signal activity detection statistic (SADS) is based on a measure of the local power of the incoming sequence, which is supposed to be higher when it contains a speech signal than when it is composed only by background noise. Because of the low-pass nature of the short-time spectrum of the speech, the detection signal-to-noise ratio is enhanced by considering only the spectral power in the frequency region that contains most of the speech information (that corresponds roughly to the band where the first three or four formants of the vocal tract are placed). A simple estimation of this power can be computed as

$$\text{SADS} = \sum_{j=N_1}^{N_2} |X(j)|^2, \tag{2.15}$$

where N_1 and N_2 are the first and last spectral lines to consider, and $\{X(j) | 1 \leqslant j \leqslant 2N\}$ is the FFT vector of a block of the incoming data.

The SADS is further compared against a threshold value to decide the presence or absence of activity. The threshold should be a function of the actual power of the noise in the line, and this can be preestimated during the set-up phase of the canceller.

The double-talking detector

Whenever there is activity both in the near and in the far-end paths of the system, the near-end signal $d(n)$ will be composed of a mixture of the echo of the received signal and the transmitted speech. As we mentioned earlier, the adaptation should be frozen in this situation, so as to avoid misconvergence of the parameters because of the reduced correlation of the near-end signal and the reference. The detection process in this case, however, cannot be based only on the power of $d(n)$, because an increment of the segmental energy of the near-end signal could be also produced by a sudden increase in the amplitude of the echo.

As an alternative indicator of double-talking, we suggest the use of a measure of the normalized instantaneous cross-correlation of the envelopes of the far-end reference $x(n)$ and the residual echo after cancellation $e(n)$. The so-called double-talking detection statistic (DTDS) is defined as

$$\text{DTDS} = \frac{E_x E_e}{E_x^2 + E_y^2}, \tag{2.16}$$

where we defined the quadratic envelopes of the input, the output and the error:

$$E_x = \sum_{j=1}^{N} x^2(j), \quad E_y = \sum_{j=1}^{N} y^2(j), \quad E_e = \sum_{j=1}^{N} e^2(j),$$

with $e(n) = d(n) - y(n) = d(n) - h(n) * x(n)$.

Notice that, *in the absence of local talk*, the following two hypotheses are reasonable:

- the power of the echo envelope is lower than that of the reference $x(n)$, i.e., the echo path *attenuates* the signal;
- the power of the residual echo is not greater than that of the uncancelled one and so we have the following inequalities:

$$0 \leqslant E_e \leqslant E_d \leqslant E_x,$$

which impose an upper bound to the value of the DTDS (taking into account that $E_y \geqslant 0$):

$$\text{DTDS} \leqslant 1 \quad \text{(without local talk)}.$$

During double-talking periods, however, the power of the envelope of $d(n)$ grows because of the presence of the local speech, and so does the power of the residual $e(n)$ because the adaptive system cannot cancel an interference without any reference

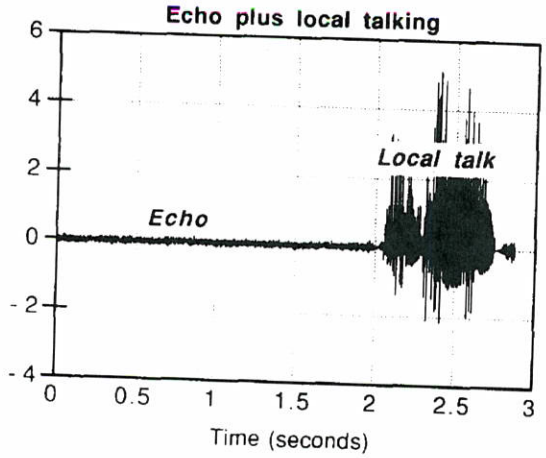


Fig. 11. Simulated echo plus local talking.

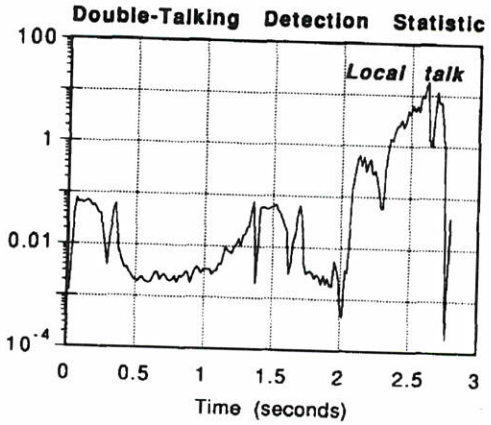


Fig. 12. Evolution of the DTDS for the signal of Fig. 11.

of it. On the other hand, the denominator in the expression for the DTDS remains essentially unchanged and, therefore, the value of the statistic increases by an amount roughly proportional to the segmental power of the local speech. The detection threshold, then, can be chosen so as to tolerate some amount of interference, freezing the adaptation process only when the gradient estimates would be too noisy to achieve a satisfactory convergence of the parameters to their optimal values.

For illustrative purposes, Fig. 11 shows an example of a simulated echo signal (see Section 4 for details about the generation method); at a certain point, a short segment of synthetic speech (25 dB over the echo) is added in order to simulate the sudden appearance of local talking.

The resulting evolution of the DTDS is shown in Fig. 12, where the increase in the value of the statistic is noticeable.

3. Implementation

3.1. Hardware and software support

The hardware prototype of the acoustic echo canceller is based on the general purpose DSP32C floating-point processor from AT&T running at 50 MHz. It is mounted on a prototyping board from LSI that is able to run on a PC-AT slot. The board is completed with two external memory banks, 8 Kword zero-wait and 32 Kword two-wait states, respectively, and two input/output 16-bit linear AD–DA channels. The canceller software is controlled by a monitor program, coded in C, that uses some standard primitive functions provided by LSI. Among others, the monitor program has facilities for displaying and recording intermediate variables and also for submitting the logical data input and outputs to internal disk or physical analog lines. The computational routines of the echo canceller are directly written in the own DSP32C macroassembler with exception of those ones concerning the initialization and control parts which are written in C. The computational part of the code is located in internal RAM and the control and initialization one in slow external RAM since they are less frequently called. The input, output and stacked data are located in fast external RAM, whereas, because of the lack of this type of fast external memory, the $2L(N+1)$ frequency coefficients have to reside in slow external memory.

3.2. System settings

The internal working parameters may be selected by filling out a configuration file that will be read prior to the initialization phase. There exist two

subsets of parameters: one depending on the application specifications and the other related to the designer's requirements. In our system, the same period for wide-band speech is 62.5 μs and the block size used is $N=256$ samples. Therefore, each block iteration takes place every 16 ms. In the current realization, we have used 8 filter sections, that is, 2048 taps or equivalently an echo path replica of 128 ms at that sampling rate. This selection is mainly imposed for testing purposes. The complete cancellation process is carried out at a rate of 16 ms, while, at the same time, the next two input blocks, far-end and local signals, are being acquired (internal disk or analog lines) and the last processed block is being sent out. It means that the total delay is corresponding to two blocks, that is, 32 ms. Once the initialization process is over, the main part of the program consists of an infinite loop, where the most important tasks, previously mentioned, are executed in a sequential way. A simple flowchart of this loop is shown in Fig. 13. The three $2N$-FFTs plus the $L(N+1)$ frequency complex multiplications and additions for the filtering operation take about $4+2.5$ ms (here is included the echo cancellation and additional loss insertions), the coefficients updating process takes about 2 ms, the activity monitor takes less than 1.5 ms, the step-size computation about 1 ms and the program overhead and acquisition interruption services about 2 ms. That is, an overall execution time of about 13 ms for the whole program in each block iteration, leaving around 3 ms free for other operations. It means that it is still possible to increase the filter length to 3072 taps (192 ms of echo path) before exceeding the iteration period. Additionally, it is important to point out that we have not used any other strategy for the inhibition vector than that derived from the talking activity results.

4. Results and conclusions

4.1. Controlled test

Up to the present date, we have not tested the prototype in all real thinkable situations. However, we can give some interesting results for controlled tests when using prestored wide-band records. The first two are white and colored stationary noises, whereas the third one is a synthetic speech record (called artificial voice) from the CCITT recommendation P.50 [3] (using a custom-built software simulator). We also use a simulated echo path, for a small size room, of 4096 taps (see Fig. 14). The echo signal is the result of filtering the synthetic signal with the simulated echo. The echo path is also simulated by using our own software written with the aid of MATLAB functions. It implements the known image method for rectangular small

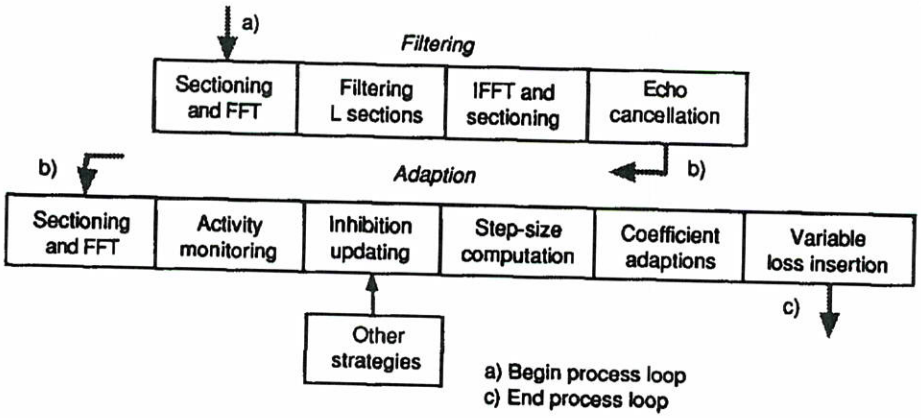


Fig. 13. General flowchart.

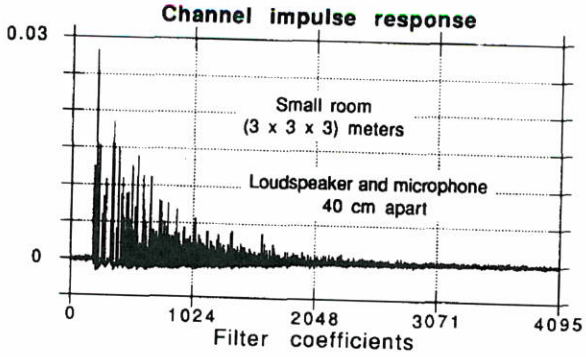


Fig. 14. 4096 taps of synthetic echo.

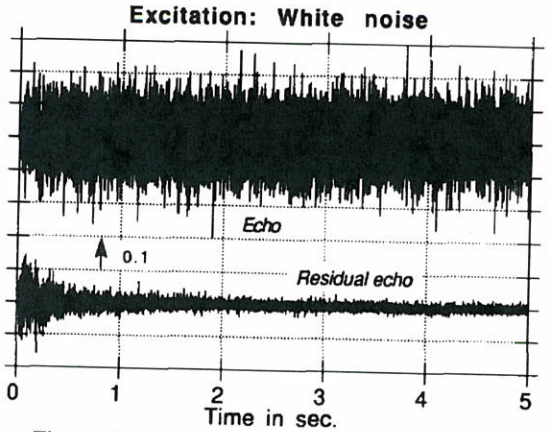


Fig. 15. Result for synthetic white noise as input.

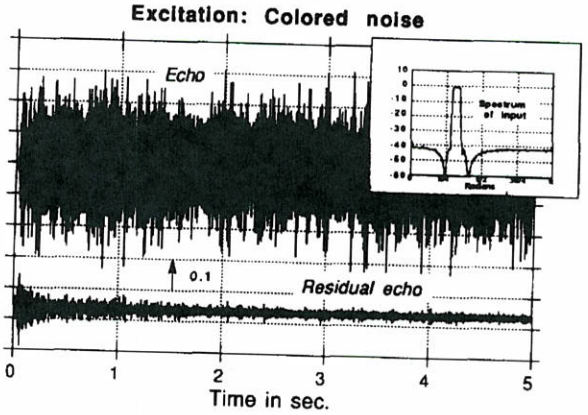


Fig. 16. Result for synthetic colored noise as input.

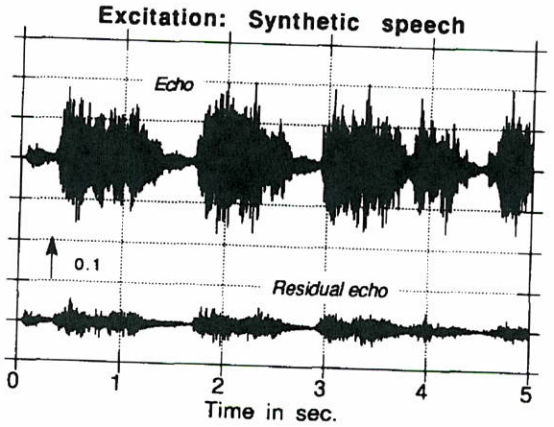


Fig. 17. Result for synthetic wide-band male speech as input.

rooms described in [1]. It provides the basic successive delays, conveniently damped, due to the multiple reflections in the surfaces of the room. Furthermore, it also has the facility of including the 3D radiation diagrams of the electroacoustic transducers (microphone and loudspeaker) in any 3D arbitrary location of the room with any desired axis direction (in Fig. 14 they are located 40 cm apart with their axes in orthogonal planes). It does not include the electrical responses of the electroacoustic elements (this is the reason why Fig. 14 only shows positive values). Both tests run without the addition of local speech or relevant local noise activities. In the final test, we use 5 seconds of male speech buried in a −25 dB white ground noise for the incoming signal.

In our realization, when using only an adaptive filter of length 2048 taps, the minimum ERLE to be reached for this echo path in a stationary white noise case is around −21 dB (here we do not consider the insertion of losses in the transmitting path), since the remaining tail (the last 2048 taps) of the impulse response behaves as an additive interference in the echo signal.

Consider now the three results of Figs. 15, 16 and 17. In the three tests we have run the program in the prototype with prestored driving and echo signals without altering any internal parameter. The first two plots show the original and residual echoes when we drive the room with 5 seconds of white and colored stationary noises. In these cases, the final and stationary measured ERLEs are

around −19.5 and −19 dB, respectively. The third plot shows the result for the synthetic wide-band speech record. Here, the measured final ERLE is around −18.4 dB. Notice that the differences during convergence are not too large. In the three cases they reach their steady state within the first two seconds and remain around their vicinities. For the first case (white noise), the spectral monitor gave a measure around the unity. It was around 0.08 in the second case of colored noise. In these two stationary situations, the deviation from the theoretical spectrum for the profile of the adaptation step vector was around 6 to 10 dB. However, in the third simulation (where voiced, unvoiced sounds and silences were possible in the input record), the greater variations in the dynamic ranges of the signal forced the adaptation step vector to vary from 10 to 35 dB. Also here, the spectral indicator moved frequently away from unity ($1 \rightarrow 0$) and, therefore, it had a tendency to attenuate the adaptation step forcing a slightly slow convergence.

These results show the good convergence properties of the system when driven with synthetic inputs. The stationary cases offer well behaved echo decreasing profiles whereas the residual echo profile of the nonstationary example tracks also quite well its corresponding level changes. Nevertheless, the synthetic record provided by the P.50 recommendation simulator has a relative slow speed of articulation and syllabic cadency in the simulated sounds. It could lead to argue that the degree of nonstationarity of such synthetic record is sufficiently small compared with the algorithm iteration rate, thus offering similar convergence results to the stationary, white noise test. To study this point in more depth, we have run another controlled simulation using real male speech in a low-pass fan noise around 20 dB below. This signal was digitally recorded (16 bits) in a non-empty office ($4 \times 5 \times 3$ meters) at a sampling rate of 16 Khz with a directional microphone to avoid any possible reverberance of the room. Afterwards, in order to have the same echo-channel reference in all the tests, the record was filtered with the same simulated channel as the previous ones to obtain its corresponding local echo. The result of the complete cancellation process for the first 5 seconds is displayed in Fig. 18. Here you can notice that even in the situation of an appreciably greater degree of nonstationarity the convergence properties remain nearly unalterable. In this case, we reached a final ERLE of −18.7 dB. It shows, at least from a practical point of view, that the proposed scheme is sufficiently robust against the usual nonstationarities found in real speech signals.

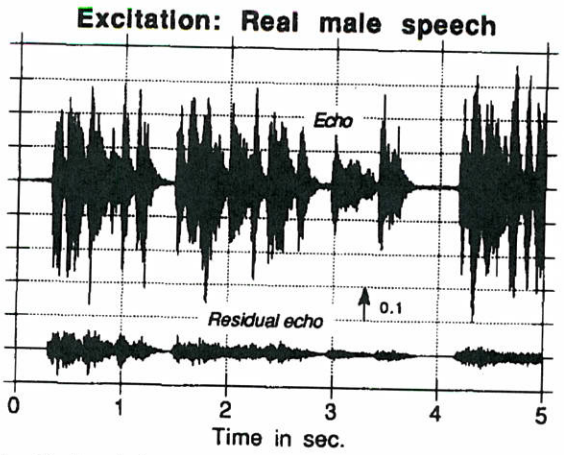


Fig. 18. Result for real wide-band male speech plus fan noise as input.

4.2. Conclusions

In this work we have presented a preliminary realization of an acoustic echo canceller implemented with only one floating point digital signal processor, the DSP32C from AT&T. We use a mixed structure for the filtering operation which causes a substantial computational reduction without paying for an intolerable delay in the data gathering. This technique allows to save a large portion of time in the filtering operation and offers the possibility of a frequency domain implementation of the adaptive algorithm. The scheme is flexible enough to allow the setting of echo path replicas of different lengths, since an existing inhibition vector, conveniently managed, might update only a reduced and predetermined number of filter sections out of the total. Furthermore, the scheme

introduces a spectral measure to improve the management of the local computed adaptation step vector avoiding any possible occasional divergence. It also allows us to alleviate the effect of strong nonstationary spectral transitions between voiced and unvoiced sounds, since the loss of excitation for a specific frequency band during a prolonged period of time causes, in some tested situations, a momentary misconvergence of the ERLE factor. This spectral measure also compensates the degradation of the periodogram spectral estimates when sharp and nonuniform spectrum are present. Finally, the scheme makes use of a standard technique for the double-talking discrimination. It is based on a statistic of the short-time correlations (a block length) of envelopes between the signals on the sending and receiving paths.

The complete system is now running with the aid of an AT-station since it is the only communication path with our lab's prototype.

Acknowledgments

Special thanks to engineers Ramón Nistal Alvarez and Antonio Martínez Marrón for their many helpful suggestions and engagement in the specification, developing and coding stages of the final prototype.

References

[1] J.B. Allen and D.A. Berkley, "Image method for efficiently simulating small-room acoustics", *J. Acoust. Soc. Amer.*, Vol. 65, April 1979, pp. 943–950.

[2] M.G. Amin et al., "A generalized noise canceller using orthogonal transformation", *Proc. EUSIPCO-88*, Grenoble, France, September 1988, pp. 419–422.

[3] CCITT, *Blue Book*, Vol. V, Rec. P.50, Melbourne, 1988.

[4] G.A. Clark et al., "A unified approach to time and frequency domain realization of FIR adaptive digital filters", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-31, October 1983, pp. 1073–1083.

[5] DSP32C PC System Board, User Manuel, Loughborough Sound Images Ltd., issue 2.01, November 1990.

[6] J.P. Jullien et al., "Acoustic echo controller for wideband hands-free telephony", *Proc. EUSIPCO-90*, Barcelona, Spain, September 1990, pp. 1983–1986.

[7] D. Mansour et al., "Unconstrained frequency domain adaptive filter", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-30, October 82, pp. 726–734.

[8] Software Development Tools for *WE*® DSP32C Digital Signal Processor, AT&T, 1988.

[9] P.C.W. Sommen, "On the convergence properties of a Partitioned Block Frequency Domain Adaptive Filter", *Proc. EUSIPCO-90*, Barcelona, Spain, September 1990, pp. 201–204.

[10] J.S. Soo et al., "Multidelay block frequency domain adaptive filter", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-38, No. 2, February 1990.

[11] H. Yasukawa et al., *Electron. Lett.*, August 1988, pp. 1039–1040.

[12] *First Workshop on Acoustic Echo Control*, Berlin, 1989.

[13] *Second Workshop on Acoustic Echo Control*, L'Aquila, 1991.