ON THE FIXED-POINT IMPLEMENTATION OF A SUBBAND ACOUSTIC ECHO CANCELER BASED ON A MODIFIED FAP ALGORITHM

Mohamed Ghanassi and Benoît Champagne

INRS-Télécommunications, Université du Québec
16 place du Commerce, Verdun, Québec, Canada, H3E 1H6

ABSTRACT

We investigate quantization effects in the fixed-point implementation of a subband AEC system based on a FAP-RLS algorithm. The latter uses a sliding window RLS (instead of FRLS) to compute the normalized residual echo vector. Subband decomposition is performed with modified uniform DFT filter banks, realized efficiently via the weighted overlap-add (WOA) technique for flexibility in oversampling. We characterize the main sources of errors in both the FAP-RLS and the DFT filter banks, and propose simple and effective solutions for stable operation of the subband AEC system. Our findings are supported experimentally.

1. INTRODUCTION

Acoustic echo cancellation (AEC) via subband adaptive filtering (combined with voice activity detection) is one of the most effective approach to the control of acoustic echos generated by hands-free terminals. In this technique, echo estimation and cancellation is realized via a set of parallel, independent adaptive filters operating on subband versions of the loudspeaker and microphone signals. Significant computational savings result from operating the subband adaptive filters at a reduced sampling rate.

Major sources of problem in designing an adaptive filter for AEC are the long duration of the echo path and the non-stationary nature of the excitation signal. As a result, instantaneous stochastic gradient algorithms, such as the NLMS, exhibit slow convergence. In recent years, several new adaptive filtering algorithms have thus been proposed for AEC applications, with the aim of achieving faster and signal-independent convergence while preserving the low complexity of NLMS. Among these, the fast affine projection (FAP) [1] is now receiving considerable attention.

In this paper, we investigate the quantization effects in the fixed-point implementation of a subband AEC system based on a FAP-RLS algorithm. The latter uses a sliding window RLS (instead of FRLS) to compute the normalized residual echo vector [2]. Subband decomposition is performed with oversampled, modified uniform DFT filter banks, realized efficiently via the weighted overlap-add (WOA) technique [3]. We characterize the main sources of errors in both the FAP-RLS and the DFT filter banks, and propose simple and effective solutions for stable operation of the subband AEC system. Our findings are supported experimentally.

2. BACKGROUND

2.1. Algorithm overview

In subband AEC, the loudspeaker signal $x(k)$ and the microphone signal $y(k)$ are split into K adjacent subbands by analysis filter banks. The subband signals at the filter outputs are downsampled by D, resulting in $x_i(m)$ and $y_i(m)$, where $i = 0, ..., K-1$ is the subband index and m is the time index at the reduced rate. In each subband, an adaptive filter with input $x_i(m)$ is used to cancel the echo component in $y_i(m)$, yielding a subband residual echo $e_i(m)$. Finally, the subband residuals are upsampled by D and combined into a single fullband residual $e(k)$ by a synthesis filter bank. As a result of subband downsampling, important computational savings of the order of D^2/K may be achieved. To avoid decimation aliasing in the subbands, oversampling (i.e. $D < K$) is now commonly used in practical implementations.

The subband AEC system under consideration here is made up of subband transversal filters adapted with separate FAP-RLS algorithms; subband analysis/synthesis is achieved via oversampled modified uniform DFT banks. Additional details follow.

A) FAP-RLS algorithm:

FAP is a kind of fast version of the affine projection algorithm (APA) [4]. Three crucial steps are invoked in its derivation from APA, namely [1]: (1) fast (approximate) update of the residual echo vector; (2) use, and fast adaptation, of an alternative filter weight vector; and (3) use of FRLS for the implicit computation of the inverse data covariance matrix of size $M \times M$ needed in APA, where M is the projection order. The complexity of FAP is roughly $2N + 20M$. When using FAP in the AEC context, significant improvements in convergence speed are observed, as compared to NLMS, with only small values of M, i.e. $M \ll N$.

The main disadvantage of FAP is its use of FRLS in step (3). Despite the low complexity of $10M$ for this step, as compared to $O(M^2)$ for a standard RLS, FRLS is quite sensitive to errors introduced by finite arithmetic, which rapidly lead to its instability. Another problem is the implementation overhead generally associated to FRLS type algorithms. Various derivatives of FAP have been proposed to further reduce its complexity and/or improve its numerical behavior. Our work here focus on a modified FAP algorithm, called FAP-RLS [2]. The latter is similar to FAP in its steps (1) and (2) above, but uses a standard sliding-window RLS in step (3). Since small values of M are used in subband AEC applications, typically ≤ 5, RLS is preferred over FRLS because of its simplicity and more robust numerical behavior.

The complex version of the FAP-RLS algorithm [1] is summarized in Table 1, where fullband notation is used to simplify the presentation. Here, $x(k)$ denotes the input signal, $y(k)$ is the desired signal, $\mathbf{x}(k) = [x(k), ..., x(k-N+1)]^t$, $\mathbf{s}(k) = [x(k), ..., x(k-M+1)]^t$, $\mathbf{R}(k)$ is a $M \times M$ sample covariance matrix, $\mathbf{r}(k)$ is a correlation vector of size $M-1$, $e(k)$ is the residual echo, $\boldsymbol{\epsilon}(k)$ is the normalized residual echo vector, $\hat{\mathbf{w}}(k)$ is the

Funding for this work was provided in part by Nortel Networks.

B. Champagne now with McGill University, Montréal, Canada

[1] For the real version, simply drop the complex conjugate symbol * and replace hermitian transpose ^{h} with plain transpose ^{t}.

$$\mathbf{a}(k) = \mathbf{R}^{-1}(k-1)\mathbf{s}^*(k)$$
$$\alpha(k) = [1 + \mathbf{s}^t(k)\mathbf{a}(k)]^{-1}$$
$$\mathbf{Q}^{-1}(k) = \mathbf{R}^{-1}(k-1) - \alpha(k)\mathbf{a}(k)\mathbf{a}^h(k)$$
$$\mathbf{b}(k) = \mathbf{Q}^{-1}(k)\mathbf{s}^*(k-N)$$
$$\beta(k) = [1 - \mathbf{s}^t(k-N)\mathbf{b}(k)]^{-1}$$
$$\mathbf{R}^{-1}(k) = \mathbf{Q}^{-1}(k) + \beta(k)\mathbf{b}(k)\mathbf{b}^h(k)$$
$$\boldsymbol{\epsilon}(k) = e^*(k)\mathbf{p}(k), \text{ where } \mathbf{p}(k) = \text{1st column of } \mathbf{R}^{-1}(k)$$
$$\mathbf{r}(k) = \mathbf{r}(k-1) + x(k)\overline{\mathbf{s}}^*(k-1) - x(k-N)\overline{\mathbf{s}}^*(k-N-1)$$
$$e(k) = y(k) - \hat{\mathbf{w}}^h(k-1)\mathbf{x}(k) - \mu\overline{\boldsymbol{\eta}}^h(k-1)\mathbf{r}(k)$$
$$\boldsymbol{\eta}(k) = \boldsymbol{\epsilon}(k) + [0, \overline{\boldsymbol{\eta}}(k-1)^t]^t$$
$$\hat{\mathbf{w}}(k) = \hat{\mathbf{w}}(k-1) + \mu\mathbf{x}(N-(M-1))\eta_{M-1}(k)$$

Table 1: FAP-RLS algorithm

alternative weight vector and μ is a relaxation factor close to 1. For initialization, we use $\mathbf{P}(0) = \mathbf{I}/\delta$ where δ is a regularization parameter, $\mathbf{r}(0) = \mathbf{0}$, $\boldsymbol{\eta}(0) = \mathbf{0}$, and $\hat{\mathbf{w}}(0) = \mathbf{0}$. The complexity of FAP-RLS is about $2N + 3M^2 + 6.5M$ complex multiplies plus 2 complex divisions per iteration.

B) Uniform DFT filter banks:

Subband analysis/synthesis is achieved via modified uniform DFT filter banks designed according to [3]. In the analysis bank, the input signal $x(k)$ is fed to a bank of K demodulators, whose outputs are low-pass filtered by $h(k)$ (ideal cut-off at π/K) and downsampled by D. The digital spectrum $[0, 2\pi]$ is thus divided into K uniform subbands of width $\Delta\omega = 2\pi/K$ and center frequency $i\Delta\omega$, $i = 0, ..., K-1$. In the synthesis bank, the input subband signals are upsampled by D, low-pass filtered by $g(k)$, modulated and summed. Aliasing distorsion is made acceptably small by using oversampling. Phase distorsion is eliminated by proper choice of the modulating functions and use of $g(k) = h(L-1-k)$. Finally, $h(k)$ is obtained as an FIR filter of length $L = n_b K$ by interpolation of a QMF to minimize amplitude distorsion.

An efficient weighted-overlap-add (WOA) approach is used for the implementation of the modified uniform DFT filter banks. This approach allows the use of arbitrary (integer) values of downsampling D, for optimum performance of the subband AEC system, i.e. trade-off between complexity reduction and subband aliasing. The main steps in the WOA implementation of the analysis bank for an arbitrary input signal $x(k)$ are summarized below: at discrete-time m (reduced rate)

1. Compute $y_m(r) = h(r)x(mD - r)$, $r = 0, 1, ..., L-1$.
2. Partition $\{y_m(r)\}_{r=0}^{L-1}$ into $n_b = L/K$ blocks of K samples.
3. Compute $\xi_m(t)$, $t = 0, 1, ..., K-1$, by adding the corresponding samples of each block.
4. Compute $x_m(t) = \xi_m((t - mD) \bmod K)$, $t = 0, 1, ..., K-1$
5. Compute $x_i(m)$, $i = 0, 1, ..., K-1$, as the K-point FFT of $\{\xi_m(t)\}_{t=0}^{K-1}$.

The corresponding steps in the WOA implementation of the synthesis bank are the dual of the above ones. The output is usually scaled by D to achieve unit gain of the overall system.

2.2. Scope of study

A) Objectives:

In the past, we have extensively studied the performance of the above subband FAP-RLS based AEC system in floating-point arithmetic, using an off-line C-language implementation on a computer worstation and a real-time assembler implementation on a TMS320C40 DSP from TI. Compared to a fullband NLMS-based AEC system, the subband FAP-RLS achieves much faster convergence at a fraction of the cost.

The main objective of this work is to develop a better understanding of quantization effects on the behavior of the subband FAP-RLS in a fixed-point implementation. This is judged important for minimizing implementation costs and avoiding undesirable numerical behaviors of the algorithm in applications. Both issues of numerical stability and precision are addressed.

B) Methodology:

We assume a b-bit fixed-point fractional representation extending from -1 to $1-\Delta$, where $\Delta = 2^{-(b-1)}$. Two's complement is used for representing negative numbers. The product x of two numbers must be quantized by either truncation or rounding: the error is denoted $q_x = Q(x) - x$, where $Q(.)$ is the quantization operator. Experimental results not reported here show a much faster error accumulation with truncation. Accordingly, rounding is assumed in our analysis and used in the DSP experiments. Quantization errors are modeled as independent random variables uniformly distributed within $\pm\Delta/2$; the corresponding variance is $\Sigma_q^2 = \Delta^2/12$.

Overflow may occur as the result of addition or division. In most commercial fixed-point DSPs, overflow is minimized by the use of accumulators with additional guard bits to the left of the MSB. Scaling (i.e. shift) of a quantity may still be necessary to avoid overflow during memory transfer. Due to lack of space, few details are provided on scaling, although it remains an essential aspect for optimal use of fixed-point resources.

The fixed-point behavior of the subband FAP-RLS is investigated analytically and by computer simulations. To this end, we have developed several programs that enable us to control the number of bits used for the representation of numbers, as well as other related aspects. The 16-bit representation is particularly important for applications. In this particular case, we have developed a software application written in C that emulates the functionality of the TMS320C54 DSP from Texas Instruments [5].

3. FIXED-POINT IMPLEMNTATION OF FAP-RLS

In this section, we study the quantization effects introduced by fixed-point arithmetic on the behavior of the FAP-RLS. To simplify the presentation, fullband notation is assumed. However, in all experiments, the algorithm parameters are set to values that are representative of a subband AEC application. In particular, the filter length is set to $N = 128$ and only small values of M are used. The length of the acoustic echo path is also set to 128 and Gaussian white noise is added to the echo when appropriate.

3.1. Update of inverse covariance matrix $\mathbf{R}^{-1}(k)$

We first study the propagation of a single quantization error in the computation of $\mathbf{R}^{-1}(k)$ in Table 1. For simplicity, first consider the case $M = 1$. Assume that at time k_o, a quantization error $q(k_o)$ is introduced in $\mathbf{R}^{-1}(k_o)$. Using the equations in Table 1, one may show that the resulting error propagated by the algorithm at time $k \geq k_o$ is $q(k) = G(k)q(k_o)$ where

$$G(k) = \prod_{i=k_o+1}^{k} \left[1 + \mathbf{R}^{-1}(i-1)(|x(i)|^2 - |x(i-N)|^2)\right]^{-2} \quad (1)$$

The behavior of the gain $G(k)$ may be investigated numerically. In the case of a stationary Gaussian white noise process $x(k)$, the results show a stationary behavior of $G(k)$, i.e. no decay of

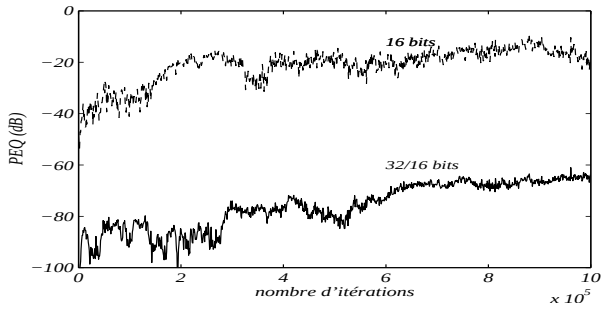

Figure 1: Error accumulation in $[\mathbf{R}^{-1}(k)]_{11}$ with 16-bit and 32/16 bit implementations

the quantization error with time. In practice, this behavior favors the accumulation of errors. In the case of a non-stationary signal, $G(k)$ is seen to increase at the beginning of an interval of small signal amplitude. Similar conclusions apply to the case $M > 1$.

Next, the accumulation effect of quantization errors on $\mathbf{R}^{-1}(k)$ is investigated via simulations. A stationary Gaussian white noise with variance $\sigma_x^2 = 0.025$ is used as input, $M = 4$ and $\delta = 12\sigma_x^2$. Fig. 1 (curve labeled 16 bits) shows the time evolution of the normalized power of the quantization error $q(k) = [\mathbf{R}_q^{-1}(k) - \mathbf{R}^{-1}(k)]_{11}$, where $\mathbf{R}_q^{-1}(k)$ is computed in fixed-point with 16-bit precision and $\mathbf{R}^{-1}(k)$ is computed in floating-point with 64-bit precision. It can be seen that the error increases with time. In other experiments with non-stationary signals, a net increase in the error power is observed at the beginning of an interval of small signal amplitude. In general, we observe a divergence of the FAP-RLS algorithm when the error level in Fig. 1 reaches about -10dB. Based on such results, we conclude that 16-bit precision is not sufficient for the computation of $\mathbf{R}^{-1}(k)$.

To extend the stable life of FAP-RLS, we increase the number of bits used in the computation of $\mathbf{R}^{-1}(k)$. Specifically, all the variables entering this computation are represented with 32 bits (i.e. two 16-bit words), except for $1 + \mathbf{s}^t(k)\mathbf{a}(k)$ and $1 - \mathbf{s}^t(k-N)\mathbf{b}(k)$ in Table 1, which are represented with 16 bits (to avoid costly 32-bit divisions). The evolution of the normalized error power is shown in Fig.1 (curve labeled 32/16). The level of error is now much lower than in the previous case; but the computation of $\mathbf{R}^{-1}(k)$ requires about 2.5 more cpu cycles. For small M, this is quite acceptable.

3.2. Update of correlation vector $\mathbf{r}(k)$

Another potential source of error accumulation in FAP algorithms is in the computation of the correlation vector $\mathbf{r}(k)$ in Table 1. To avoid such accumulation, $\mathbf{r}(k)$ should be computed as follows

$$\mathbf{r}(k) = \mathbf{r}(k-1) + Q[x(k)\overline{\mathbf{s}}(k)] - Q[x(k-N)\overline{\mathbf{s}}(k-N)] \quad (2)$$

where $Q(.)$ denotes quantization by rounding. This approach introduces a small quantization error, but no accumulation of the latter, as verified experimentally.

3.3. Filtering and adaptation

The computation of $e(k)$ and $\hat{\mathbf{w}}(k)$ in Table 1 is similar in nature to the NLMS algorithm, while the computation of $\boldsymbol{\eta}(k)$ does not permit error accumulation. However, these operations use $\mathbf{R}^{-1}(k)$; We have verified that accumulation errors in this quantity may lead to instability of the FAP-RLS algorithm.

To investigate this behavior of the FAP-RLS, we have used a CSS as input signal $x(k)$. With 16-bit precision, we observe important instabilities of FAP-RLS; these are the result of error accumulation in $\mathbf{R}^{-1}(k)$, and occur principally during silence periods. Reinitialization (see below) is not always effective to overcome this problem. However, when the FAP-RLS is implemented in 16-bit and the computation of $\mathbf{R}^{-1}(k)$ in 32/16-bit as described above, a robust performance is observed and the algorithm remains operational over extremely long periods of time.

To avoid potential problems at the beginning of a silence period in the case of speech signals, we have implemented and tested a reinitialization mechanism in which $\mathbf{R}^{-1}(k)$ is reset to $\mathbf{I}/\delta$ after detection of a silence period (i.e. signal power below predefined threshold) of duration N samples. In this case, previous signal samples $x(k-i)$, $i = 0, ..., N-1$, and vectors $\boldsymbol{\eta}(k)$ and $\mathbf{r}(k)$ are also reset to zero. With this mechanism, instability were never observed in our experiments with the 32/16-bit implementation. This approach is simple, does not require significant extra computations, and has no effects on the observed convergence behavior of the FAP-RLS.

3.4. Algorithm precision

We finally investigate the effects of the number b of representation bits on the transient and steady-state [2] behaviors of the algorithm. In the experiments, the background noise power is set to zero. Results (not shown due to lack of space) indicate that the quantization errors have no effects on the initial convergence speed. In the steady-state regime, a decrease in the residual error level of roughly 7dB to 9dB per additional bit is observed. With $b = 16$ bits, the residual error level would remain below that of the background noise usually present in applications.

4. FIXED-POINT WOA IMPLEMENTATION

Since, the WOA technique makes use of the DFT, we first investigate and compare the effects of quantization on a direct computation of the DFT and on the FFT algorithm. Quantization effects in the analysis steps and synthesis steps of the WOA method are then studied separately. Since all the operations involved are stable, only the precision aspect is considered.

4.1. DFT versus FFT?

The K-point DFT of a sequence $x(k)$, $k = 0, .., K-1$ is defined by $X(n) = \sum_{k=0}^{K-1} x(k)W_K^{nk}$, $n = 0, ..., K-1$, where $W_k = e^{-j2\pi/K}$. In practice, instead of using the above definition (i.e. direct method), the DFT is evaluated via the FFT algorithm. However, sor small values of K, the computational complexity of the FFT does not largely exceed that of the direct method. Thus, a study of quantization effects on both approaches deserves to be done.

To evaluate the signal to quantization noise ratio (SQNR), we assume a statistical model of the quantization error as described previously. Quantization of a complex number introduces an error $q = q_r + jq_i$ with variance $2\Sigma^2$. For simplicity, the signal $x(k)$ (either real or complex) is modeled as a white noise sequence with variance σ_x^2. We assume that an error q_x of variance Σ_x^2 has been introduced in the quantization of $x(k)$. We also assume that the fixed-point DSP under consideration has at least $\log_2 K$ guard bits beyond the MSB in its accumulators, so that scaling is only necessary when the final result is being transferred to memory. To avoid overflow during the transfer, the result of the computation is in

[2] Before error accumulation in $\mathbf{R}^{-1}(k)$ becomes important.

effect scaled down (shifted towards LSB) by a factor $1/K$ (since $|X(n)| \leq \sum_{k=0}^{K-1} |x(k)| < K$).

For the direct method, a straightforward calculation of the SQNR yields the following result:

$$SQNR \simeq \sigma_x^2/(2K\Sigma^2 + \Sigma_x^2) \quad (3)$$

For the FFT, a more elaborate calculation, involving a recursion over the butterfly index, yields

$$SQNR \simeq \sigma_x^2/(4K\Sigma^2 + \Sigma_x^2) \quad (4)$$

According to (3)-(4), the direct method presents a small advantage of at most 3dB with respect to the FFT. As an example, for a real process $x(k)$ with $\sigma_x^2 = 0.1$, $\Sigma_x = \Sigma$, $K = 16$ and $b = 16$ bits, the computed SQNR is about 76dB for the direct method and 73dB for the FFT. These figures perfectly match the values measured experimentally with the TMS320C54 simulator.

The small loss in SQNR with FFT is compensated by its faster operation: for $K = 16$, the FFT is 1.6 times faster than the direct method for real data and 3.4 times for complex data. Based on these observations, FFT remains advantageous in our application.

4.2. Analysis bank

We next study quantization effects in the analysis steps of the WOA implementation. The signal $x(k)$ is modeled as a real, white noise sequence with variance σ_x^2. We assume that quantization errors $q_x(k)$ and $q_h(k)$ are present in $x(k)$ and the low-pass filter $h(k)$ (also real), with variance $\Sigma_h^2 = \Sigma_x^2 = \Sigma^2$. Considering the introduction of quantization errors at the various steps in the analysis portion of the WOA method, the propagation of these errors in subsequent steps, and using results obtained in the above study of the FFT algorithm, one may show that the SQNR in each branch $x_i(m)$ at the output of the analysis bank is approximately given by

$$SQNR \simeq \|\mathbf{h}\|^2 \sigma_x^2/[(L + 4K^2)\Sigma^2] \quad (5)$$

where $\mathbf{h} = [h(0), h(1), ..., h(L-1)]^t$ is a vector containing the coefficients of the analysis prototype filter. The above formula is valid provided $\|\mathbf{h}\|^2/L \ll 1$ and $\sum_{s=0}^{n_b-1} h(t+sK)^2 \simeq n_b \|\mathbf{h}\|^2/L$. Since $|h(k)| < 1$ and L is usually of the order of 10^2 of more, these two conditions are usually satisfied.

As an example, consider a real white noise process $x(k)$, with variance $\sigma_x^2 = 0.1$, going through the analysis bank in Design Example 2 in [3], for which $K = 16$, $D = 12$ and $L = 128$. In this case, the above formula yields an SQNR of about 60dB. This result is in perfect agreement with fixed-point experiments on the TMS320C54 simulator. Such a value of SQNR is well below the background noise level in typical subband AEC applications. Accordingly, we conclude that quantization errors in the analysis banks do not represent a significant impairment.

4.3. Synthesis bank

Here, the input is a set of subband signals, say $\{x_i(m)\}$, obtained by passing a real signal $x(k)$ with variance σ_x^2 through the WOA implemented analysis bank. As a result, the quantization error in $x_i(m)$ has variance $\Sigma_{x_i}^2 = (4 + L/K^2)\Sigma^2$. Although the synthesis uses inverse FFT instead of FFT, the variance formulae obtained for FFT may be easily modified. Following a similar approach as the one taken in Section 4.2, the SQNR at the output of the synthesis bank, denoted $\hat{x}(k)$ is obtained as

$$SQNR \simeq K\sigma_x^2/[D\|\mathbf{h}\|^2(L + 4K^2)\Sigma^2] \quad (6)$$

For the same example as in Section 4.2, (6) gives an SQNR of about 61dB, in agreement with fixed-point experiments. For this example, we find $\Sigma_{\hat{x}_i}^2 \simeq 4.5\Sigma^2$ after the analysis and $\Sigma_{\hat{x}}^2 \simeq 1000\Sigma^2$ after the synthesis. Thus, quantization errors introduced in the synthesis bank are much more important than those introduced in the analysis bank. We also find that the reconstruction error at the output of the synthesis bank is about -40dB, as a result of non-ideal selective property of the prototype filter $h(k)$. Our main conclusion here is that in a 16-bit fixed-point WOA implementation of the uniform DFT filter banks with practical values of L, quantization errors are usually small compared to the reconstruction error and may be neglected.

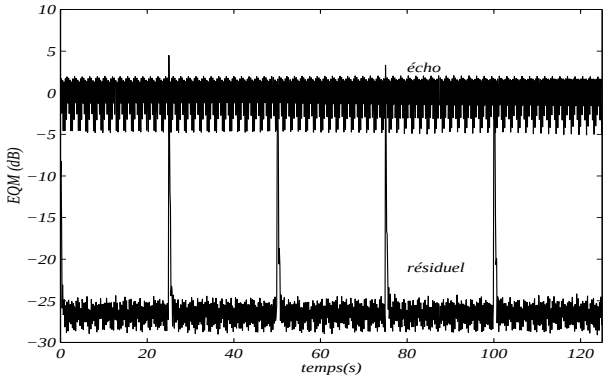

Figure 2: Short-term power of residual echo in fixed-point implementation of subband FAP-RLS.

5. EVALUATION OF COMPLETE ALGORITHM

Finally, we study the behavior of the complete subband FAP-RLS algorithm when implemented in fixed-point.

The parameters of the filter banks are chosen as follows: $K = 16$, $D = 12$, $L = 128$, $h(k)$ as in Design Example 2 in [3]. For the subband FAP-RLS algorithms, we use $N = 1000/12 = 83$ and $M = 4$ in all subbands (the values of μ and δ differ slightly). All the algorithms are implemented in 16-bit fixed-point using the TMS320C54 simulator. The 32/16-bit technique is used for the computation of $\mathbf{R}^{-1}(k)$. For the evaluation, a CSS sequence sampled at $F_s = 8$kHz is used as the input $x(k)$ (loudspeaker). The microphone signal $y(k)$ is obtained by convolving $x(k)$ with a room response, whose polarity is reversed every 25 seconds to simulate time-varying conditions; background noise at -30dB is added to the result. The short-term power of the (fullband) residual echo is shown in Fig. 2. Not that reinitialization has not been used here, although it may be included in practice for safer operation.

6. REFERENCES

[1] S. L. Gay *et al.*, "The fast affine projection algorithm," ICASSP-95, p. 3023.

[2] Q. G. Liu *et al.*, "On the use of a modified FAP algorithm in subbands for acoustic echo cancellation," *Proc. 7th IEEE DSP Workshop*, Norway, 1996, p. 354.

[3] Q. G. Liu *et al.*, "Simple design of oversampled uniform DFT filter banks with applications to subband acoustic echo cancellation," Submitted to *Signal Processing*, Nov. 1998 (See also, *Proc. IWAENC-97*, p. 132.)

[4] K. Ozeki *et al.*, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Elec. and Comm. in Japan*, vol. 67-A, no. 5, 1984, p. 19.

[5] Texas Instruments, "TMS320C54x DSP Reference Set," 1998.