EUROPEAN ACADEMIC RESEARCH
Vol. V, Issue 2/ May 2017

Impact Factor: 3.4546 (UIF)
DRJI Value: 5.9 (B+)

ISSN 2286-4822
www.euacademic.org

New Approach of Performance Analysis of RLS Adaptive Filter Using a Block-wise Processing

HAMZÉ HAIDAR ALAEDDINE
HKS Laboratory, Electronics and Physics Dept.
Faculty of Sciences I, Lebanese University, Lebanon
ALAA A. GHAITH
HKS Laboratory, Electronics and Physics Dept.
Faculty of Sciences I, Lebanese University, Lebanon
MOSTAFA AL-SAHILI
HKS Laboratory, Electronics and Physics Dept.
Faculty of Sciences I, Lebanese University, Lebanon
MOUENES FADLALLAH
HKS Laboratory, Electronics and Physics Dept.
Faculty of Sciences I, Lebanese University, Lebanon

Abstract:

This paper presents an efficient implementation of adaptive filtering for echo cancelers for teleconference systems during single-talk situation. The problem of echo cancellation is recurrent for all modern communication systems. The general solution consists to estimate the room impulse response and reproduce the echo signal in order to subtract it from the received signal and then reduce its effect at the far-end user side and consequently improves the conversation quality. Several types of adaptive algorithms exist in the estimation process. The most popular echo canceler (EC) uses Least Mean Squares (LMS) or Recursive Least Squares (RLS) adaptive filter. In practice, the choice of an algorithm must be made bearing in mind some fundamental aspects:

- Faster convergence in the presence of large variations in the far-end speech: the acoustic echo cancellation system must provide an echo reduction of about 24 dB for delays lower than 25 ms and of about 40 dB for delays exceeding 25 ms.

- Reduced computational complexity algorithms.

The literature shows that the RLS algorithm is more efficient than the LMS algorithm. It allows to reach an echo reduction of 46 dB which is large enough than the required threshold. Moreover, it presents a faster convergence compared to the LMS algorithm during single-talk situation. However, the implementation of the RLS algorithm applied to acoustic echo cancellation system has shown the disadvantage of requiring a strong computational complexity. To reduce the computational cost of this algorithm, we propose a new version of the RLS algorithm based on the Fast Fourier Transform (FFT) and on the recursive operation over a block of N samples instead of one sample.

Simulation results show that the proposed algorithm has a faster adaptation convergence, during single-talk situation, and reduce considerably the hardware resources needed for the implementation and the execution time in terms of N.

Key words: Acoustic Echo Cancellation, Adaptive Filter, Adaptive algorithm, LMS Algorithm, RLS Algorithm.

1. INTRODUCTION

The ever-increasing demand for telecommunication systems is stimulating research efforts to develop efficient systems, in order to transmit an intelligible and good quality speech signal. These problems such as sound cancellation and noise reduction must be resolved without affecting the quality of the signal. The problem dealt consists this paper is that of acoustic echo control in teleconferences in case of simple speech. The signal emitted from a distant speaker by the loudspeaker is received after passing through the acoustic channel of the so-called local microphone and returned to this same speaker. When the acoustic echo is present in an inconvenient way, ie clearly distinct subjectively from its original signal, a specific treatment, called acoustic echo cancellation, must be implemented in order to preserve the quality of the communication.

Adaptive filtering is proved to be an effective tool for this task for several years: In fact, filtering is made adaptive if its parameters are modified according to a given criterion, as soon as a new value of the signal becomes available. Adaptation of the adaptive filter coefficients has been carried out for decades using the LMS (Least Mean Squares) algorithms and has also been performed with recursive type methods such as RLS (Recursive Least Squares).

The RLS algorithm makes it possible to obtain efficient results when identifying the acoustic channel, that is to say with a faster convergence rate compared to the LMS algorithm. On the other hand, its implementation is more complex and costly.

The choice of the optimization algorithm will be made according to two criteria, which are; The speed of convergence which will be the number of iterations necessary to converge close enough to the optimal solution and the computational complexity of the adaptive algorithm.

The main objective of this paper is to propose a new version of the adaptive filtering algorithm RLS by processing it in blocks of samples. This block processing method has the following advantages:

- It processes a block of samples at each iteration, which reduces the calculation execution time.
- It involves a series of convolution products which can be realized by the implementation of the fast Fourier transform.
- It allows an implementation on a fixed-point processor.
- Its convergence rate is fairly rapid.

On the other hand, this paper is organized as follows: Section 2 presents quickly the acoustic echo cancellation system and recalls the theory of adaptive filtering. It also recalls and compares the two different adaptation algorithms, LMS and RLS. Section 3 describes the method of processing the RLS algorithm by sample blocks and highlights the value of using this method in signal processing. It also presents an implementation using the FFT of this new algorithm. In the last section, numerical results of the realization of the RLS algorithm by blocks using the FFT are given.

2. ACOUSTIC ECHO

The acoustic echo is caused by the transmission of the signal emitted by the loudspeaker and received by the microphone: this transmission is composed of a direct path and multiple reflections captured by the microphone, and consequently return to the speaker who spoke, in a distant room, his own signal.

The impulse response of the echo path is in the form of a direct wave and a succession of waves reflected from the walls of a particular room. The long duration of the echo path is mainly due to the low speed of sound in the air. When the acoustic echo is present in an inconvenient way, it is necessary to remove it [Gilloire 1987].

2.1. Acoustic echo cancellation

Telecommunication systems use an acoustic echo canceller to cancel unwanted echoes resulting from coupling between a loudspeaker and a microphone. Generally, echo cancellation is performed by modeling the impulse response of the echo path by adaptive finite impulse response filtering and subtracting an echo estimate of the microphone signal [Kazuo 1990], [Farhang 1998]. A typical acoustic echo canceller has been illustrated in Figure (1).

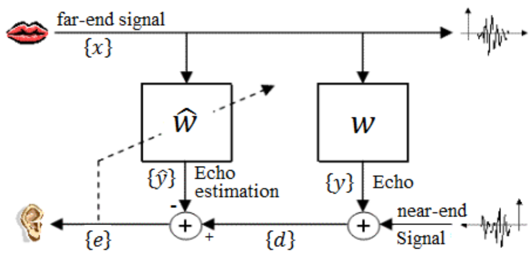

Figure 1. Principle of the acoustic echo cancellation system

The echo cancellation is performed by identifying the impulse response of the echo path by adaptive filtering represented by its coefficients, $\hat{w}_k = \left[\hat{w}(0)\ \hat{w}(1)\ldots\hat{w}(L-1)\right]^T$, where k represents the discrete time. Typically, the echo cancellation system uses two main adaptive algorithms which are: LMS (Least Mean Squares) and Recursive Least Squares (RLS).

2.2. Acoustic echo cancellation algorithms

2.2.1. Least Mean Square Algorithm (LMS)

The Least Mean Square algorithm (LMS) designed in 1959, is based on the gradient method which computes and updates recursively the weights. This gradient method consists of obtaining from a given vector $\hat{w}_k$, a vector $\hat{w}_{k+1}$ by incrementing of the vector $\hat{w}_k$ in the opposite direction of the gradient of the cost function ζ : [Kazuo 1990], [George 1999], [Paleologu 2010]:

$$\hat{w}_{k+1} = \hat{w}_k - \frac{1}{2}\mu\nabla\zeta \quad (1)$$

where μ is the adaptation step of the gradient algorithm which controls the convergence of the adaptive filter.

The cost function ζ, which will minimize the mean squared error (MSE), is defined by:

$$\zeta_k = E\left(e_k^2\right) \quad (2)$$

The error e_k is a quadratic form of the weights, and intuitively, the optimal solution is obtained by stepwise correcting the weighting vector in the direction of the minimum.

$$e_k = d_k - \hat{y}_k = d_k - x_k^T \hat{w}_k \tag{3}$$

- $x_k = [\mathrm{x}_k \ \mathrm{x}_{k-1} \ \ldots \mathrm{x}_{k-L+1}]^T$ is the input sequence.

- $\hat{y}_k$ is the echo estimation that can be obtained by filtering the input sequence x_k with the coefficients $\hat{w}_k$ of the filter as follow:

$$\hat{y}_k = \sum_{n=0}^{L-1} \hat{w}_k(n) \mathrm{x}_{k-n} = x_k^T \hat{w}_k \tag{4}$$

Adaptation of the coefficients of the adaptive filter $\hat{w}$ according to the LMS algorithm is given by:

$$\hat{w}_{k+1} = \hat{w}_k + \mu x_k e_k \tag{5}$$

The computational complexity of the LMS algorithm is known: each iteration consists $(2L + 1)$ multiplications and $(2L)$ additions. It can be seen that the LMS algorithm has an extremely low computational complexity. On the other hand, for non-stationary signals, it is difficult to follow the variations of the input signal by adapting the coefficients of the adaptive filter by this algorithm, which gives a slow convergence. To solve this problem, we use the RLS algorithm.

2.2.2. Recursive Least Squares Algorithm (RLS)

To compensate the problem of the non-stationarity of the signal, we use the Recursive Least Squares (RLS) algorithm. In this method, instead of minimizing a statistical criterion established on the error committed by estimating a signal $\{d\}$, we minimize, at each iteration k, the weighted sum of the squares of the errors committed from the initial moment. In this case the cost function ζ_k, is given by: [Ljung 1983], [Benesty 2011]

$$\zeta_k = \sum_{n=0}^{k} \left(d_n - \hat{y}_n\right)^2 \qquad (6)$$

The signal estimation $\{d\}$ using the least-squares method and an impulse response $\hat{w}_k$, is obtained when the cost function ζ_k is minimized.

By analogy with the LMS algorithm, the estimated impulse response of the RLS filter is therefore to be modified with each new iteration. To limit the number of computations, we use a recursive equation given by:

$$\hat{w}_{k+1} = \hat{w}_k + G_k e_k \qquad (7)$$

$$e_k = d_k - \hat{y}_k = d_k - x_k^T \hat{w}_k \qquad (8)$$

The column vector G_k, with size *L*, is called the Kalman gain, which can be defined as follows:

$$G_k = \frac{\lambda^{-1} P_{k-1} x_k}{1 + \lambda^{-1} x_k^T P_{k-1} x_k} \qquad (9)$$

- λ is a weighting factor that always takes a positive value: 0<< λ ≤1. This factor is also called forgetting factor because it is used to forget data that corresponds to a distant past.

- P_k is the inverse auto-correlation matrix of size (*L*×*L*), which can be calculated recursively as follows:

$$P_k = \lambda^{-1}\left(P_{k-1} - G_k\left(x_k^T P_{k-1}\right)\right) \qquad (10)$$

$P_0 = \delta^{-1}.I_L$, δ is a positive constant

I_L is an identity matrix of size (*L*×*L*)

The details of the calculations necessary to arrive at the formulation of the RLS algorithm which is represented by equations (8), (9) and (10) is given in the Appendix.

The computational complexity of the RLS algorithm is proportional to L^2, the number of additions/subtractions, multiplications/divisions implemented for the RLS algorithm at each iteration is respectively of the order $2L^2+1$, and $3L^2+7L$. Then the RLS algorithm requires many more operations than the LMS algorithm.

2.3. Performance comparison of LMS and RLS algorithms

Numerical simulations were made in the single talk situation to evaluate the performance of the two adaptive algorithms LMS and RLS using MatLab programming software.

The two algorithms are evaluated by a speech signal and an echo path impulse response respectively presented in the two figures (2-a) and (2-b). In our study, we are interested in the case of a simple speech (presence of a distant signal only).

The problem that arises is that of choosing an optimization algorithm. This choice will be determined by the convergence speed of the filter and the computational complexity. The optimal algorithm is one that satisfies these two criteria.

The attenuation of the echo can be measured by the evolution of the convergence N_m of the adapted filter given by the two algorithms LMS and RLS. The measurement is carried out in a conventional manner using the L successive samples, that is to say, at the index time k:

$$N_m = 10log_{10}\left[\frac{\|w-\hat{w}\|^2}{\|w\|_2}\right] \qquad (11)$$

Where w and $\hat{w}$ denote respectively the impulse response and the estimated impulse response of the echo path.

The results of the simulations are obtained by choosing the length of the adaptive filter, equal to 128 (the acoustic echo is therefore present for an overall transmission delay of 16 ms), the adaptation step $\mu = 0.05$, the forgetting factor $\lambda = 1$ and $\delta = 8$.

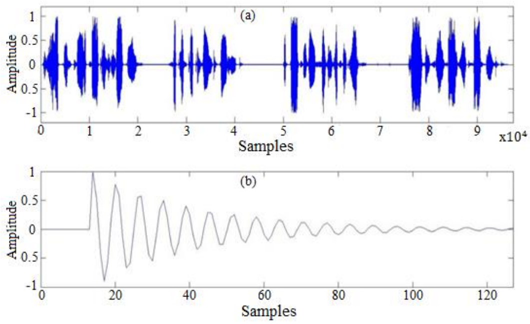

Figure 2. (a) Input Signal (b) The impulse response of the echo path

An echo cancellation system must provide echo attenuation of the order of 24 dB. The same system must be able to provide 40 dB echo attenuation for delays exceeding 25 ms. [Gilloire 1994] Figure (3) shows the evolution of the convergence of two adaptive algorithms, LMS and RLS as a function of the number of samples. This figure shows that the RLS (discontinuous line curve) algorithm has a faster convergence rate (-46 dB) than the LMS algorithm (-38.7 dB). The difference between the two convergence speeds can reach 7 dB.

The choice of the value of the coefficients is only an example to demonstrate the performance of the RLS algorithm with respect to the LMS algorithm. Any other choice of the coefficients can only show that the RLS algorithm remains the most efficient in terms of making the residual echo less audible at the output of the acoustic echo cancellation system.

The objective observations, such as the measurement of the convergence of the coefficients of the adaptive filter which remain insufficient to properly evaluate the acoustic echo cancellation system. For this purpose, a listening test was carried out to obtain a subjective assessment of the quality of listening to the residual echo issued from both echo cancellation systems, successively adapted by the LMS and the RLS. The echo cancellation system works well in the case of simple speech. We observed that the residual echo provided by the RLS algorithm becomes less audible compared to that provided by the LMS algorithm. On the other hand, the implementation of the RLS algorithm presented a significant global computational load compared to the LMS algorithm. Therefore, problems of execution time and complexity of computation always arise when we try to implement this algorithm in a real time processor. For these reasons, it is then necessary to reduce the costs of these operations to be processed. To remedy to this problem, a new technique to treat the RLS algorithm has been proposed in this article. This technique consists of processing the RLS algorithm by blocks of samples instead of processing it sample by sample.

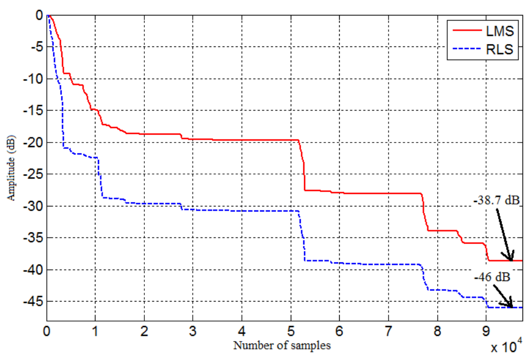

Figure 3. Convergence of filter coefficients by the two algorithms

3. Recursive Least Square by Block Structure

The main idea of our work is to propose a new and more efficient version of the RLS algorithm. In fact, instead of adapting the coefficients to each new sample, that meant that adapting the filter coefficients 16000 times per second for a sampling frequency $f_s = 16$ kHz. The proposed algorithm, "Block RLS (BRLS)", consists of adapting the filter coefficients, only each N samples, N>1 designating the size of the adaptation block. This implies a reduction of the execution time and a reduction of the calculation complexity by a factor N, at least for the adaptation part.

In this section, we describe the block adaptive filtering procedure. The mathematical formulation of this procedure is obtained in vector form by processing the *N* samples of the interval $[kN;kN+N-1]$ at each iteration, k denoting the index of the block ($k\in\mathbb{N}$).

Let $\hat{w}_k$ the adaptive filter of length *L* designed to estimate the acoustic feedback path and updated at each iteration k. The estimated echo using a block-processed algorithm is given by: [Clark 1981], [Alaeddine 2012]:

$$\tilde{y}_k=\begin{bmatrix} x_{kN} & x_{kN-1} & \cdots & x_{kN-L+1} \\ x_{kN+1} & x_{kN} & \cdots & x_{kN-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{kN+N-1} & x_{kN+N-2} & \cdots & x_{kN+N-L} \end{bmatrix}\begin{bmatrix} \hat{w}_k(0) \\ \hat{w}_k(1) \\ \vdots \\ \hat{w}_k(L-1) \end{bmatrix}=\tilde{R}_k.\hat{w}_k$$

$$=\left[\hat{y}_{kN} \quad \hat{y}_{kN+1}\ldots\hat{y}_{kN+N-1}\right]^T \qquad (12)$$

Where $\tilde{R}_k$ is the Toeplitz matrix of size ($N\times L$).

The principle of block processing is described by the figure (4). This figure shows that the mathematical formulation of the procedure for processing the RLS algorithm by blocks is obtained by replacing the input vector $x_k=[x_k \ x_{k-1} \ \ldots x_{k-L+1}]^T$ by the Toeplitz matrix in the adaptive equations of the filter. This algorithm will be called Toeplitz-BRLS.

The error vector (residual echo) of length N is given by:

$$\varepsilon_k=\left[e_{kN} \quad e_{kN+1}\ldots e_{kN+N-1}\right]^T \qquad (13)$$

where $\tilde{d}_k=\left[d_{kN} \quad d_{kN+1}\ldots d_{kN+N-1}\right]^T$ is the vector of the echo signal of length *N*.

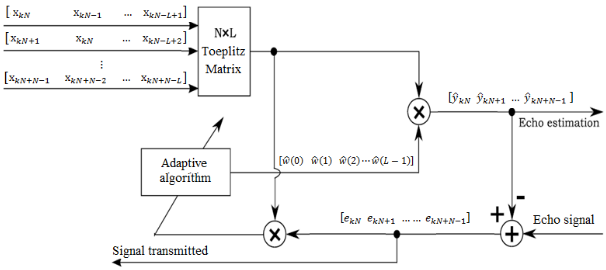

Figure 4. Adaptive filtering by block structure

From the equations (12 and 13), it can be seen that block processing allows us to calculate N elements at each iteration k, unlike sample processing.

The implementation of the Toeplitz-BRLS algorithm which is based on matrix computation is very complicated from a computational point of view. To remedy this problem, we propose to perform this calculation by the circular discrete convolution which can be made less expensive and does not modify the treatment to be carried out. The elements of the estimated echo signal, $\tilde{y}_k$, will be calculated so that the property of the circular convolution can be used as follows: [Khong 2005]

$$\hat{y}_{kN+m} = \sum_{i=0}^{L-1} \hat{w}_k(i)\, x_{kN+m-i} = \hat{w}_k(m) * x_k(m) \qquad (14)$$

where * represents the circular convolution and $0 \le m \le N-1$.

Similarly, the matrix product, $\tilde{R}_k^T \varepsilon_k$, in both equations (7) and (9) of the adaptive filter coefficient update can be calculated by the discrete convolution as follows:

$$\tilde{R}_k^T \varepsilon_k = \sum_{j=kN}^{kN+N-1} e_j . x_j = \phi_k \qquad (15)$$

$\phi_k = \tilde{R}_k^T \varepsilon_k$, corresponds to a cross-correlation whose coefficients ϕ_{ik} are given by:

$$\phi_k = \sum_{m=0}^{N-1} e_{kN+m}.\mathrm{x}_{kN+m-i}, \quad 0 \le i \le L-1 \tag{16}$$

If we put $n = kN + m,$ the ϕ_{ik} can be written in the form of a convolution in the following way:

$$\phi_{ik} = \sum_{n=kN}^{kN+N-1} e_n.x_{n-i} = e_n * x_{-i} \tag{17}$$

From the equations (14) and (17), it can be seen that the adaptive filtering by the proposed algorithm "Block-RLS (BRLS)" consists in calculating the circular convolution between the vectors $\hat{w}_k$ and $\tilde{x}_k$, of respective length N and $N+L-1$, and the vectors ε_k and $\tilde{x}_{-k}$, where $\tilde{x}_k$ being the sequence of the input signal defined by:

$$\tilde{x}_k = [\mathrm{x}_{kN-L+1} \quad \mathrm{x}_{kN-L+2} \ \dots \mathrm{x}_{kN+N-1}]^T \tag{18}$$

We can conclude that with our proposition, replacing the matrix calculation, by the convolution product, the computational complexity will be proportional to L instead of L^2.

The adaptation of the coefficients of the adaptive filter by the proposed BRLS algorithm can be represented according to the equation:

$$\hat{w}_{k+1} = \hat{w}_k + \frac{\lambda^{-1} p_{k-1}}{1+\lambda^{-1}\tilde{x}_k^T p_{k-1}\tilde{x}_k}.\left(\tilde{x}_{-k} * \varepsilon_k\right) \tag{19}$$

Following our proposal, the estimated echo is given by:

$$\tilde{y}_k = \hat{w}_k * \tilde{x}_k \tag{20}$$

Figure (5) shows the advantage of using the block processing method to reduce the execution time and in a particular case where the length of the block is equal to $N = 2$.

This figure demonstrates the interest of our proposal at the execution time: Indeed, if the length of the block is equal to N=2, the block processing method allows us, for 3 iterations, to treat 6 new samples compared with the other method (Figure

5.a), which processes only 3 new samples for the same number of iterations. So we can see that our proposal to process the RLS algorithm by blocks of samples allows us to reduce the execution time as a function of the length *N* of the block.

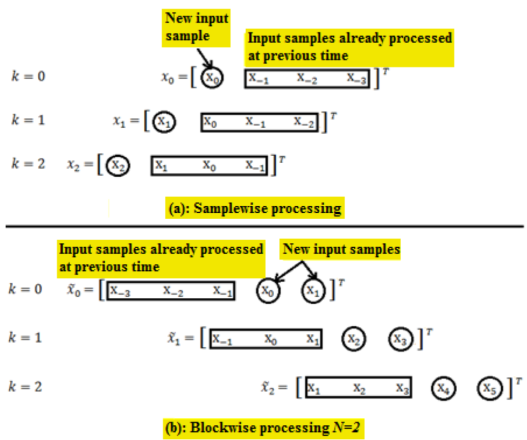

Figure 5. Representation of the input sequence by the two adaptation methods

A circular convolution in the time domain is equivalent to a term-to-term multiplication in the domain of the Fast Fourier Transform (FFT), which makes it possible to reduce the computational complexity and the execution time in the algorithms using the FFT.

3.1. Block-RLS by the Fast Fourier Transform (FFT-BRLS)

The circular convolution presented in the BRLS algorithm can be calculated in the FFT domain with reduced computational complexity.

In this way, we take advantage of the circular convolution properties of the discrete Fourier transform (DFT) and the speed of computation by the Fast Fourier Transform (FFT). The BRLS algorithm consists of calculating the circular convolution between vectors $\hat{w}_k$ and $\tilde{x}_k$, and vectors ε_k and $\tilde{x}_{-k}$.

Note that $\tilde{Y}_k$ is the product of the circular convolution between $\tilde{x}_k$ and $\hat{w}_k$:

$$\tilde{Y}_k = \left(\begin{bmatrix}\hat{w}_k^T & 0_{1\times(M-L)}\end{bmatrix}\right) * \left(\begin{bmatrix}\tilde{x}_k^T & 0_{1\times M-(N+L-1)}\end{bmatrix}\right)^T \quad (21)$$

$$= FFT^{-1}\left(FFT\left(\begin{bmatrix}\hat{w}_k^T & 0_{1\times(M-L)}\end{bmatrix}\right) \bullet FFT\left(\begin{bmatrix}\tilde{x}_k^T & 0_{1\times M-(N+L-1)}\end{bmatrix}\right)\right)^T$$

Where FFT and FFT^{-1} denote respectively the direct and inverse Fourier transforms. $0_{(i\times j)}$ is a null matrix of size $(i\times j)$. For the calculation of the FFT, it is necessary to choose M power of 2. In general, we choose $M = N + L - 1$ (in our case $N = L$) if this value is a power of 2, if not it is necessary to complete the vectors by a necessary samples of zero coefficients in order to give them a size equal to a power of 2, as shown in equation (21).

At each iteration k, the adaptation of the adaptive filter is then performed by calculating the circular convolution in the domain of the FFT following this expression:

$$\hat{w}_{k+1} = \hat{w}_k + \alpha.\Re\left(\left(\begin{bmatrix}\varepsilon_k^T & 0_{1\times(M-N)}\end{bmatrix}\right) * \left(\begin{bmatrix}\tilde{x}_{-k}^T & 0_{1\times M-(N+L-1)}\end{bmatrix}\right)\right)^T \quad (22)$$

$$= \hat{w}_k + \alpha.\Re\left(FFT^{-1}\left(FFT\left(\begin{bmatrix}\varepsilon_k^T & 0_{1\times(M-L)}\end{bmatrix}\right) \bullet FFT\left(\begin{bmatrix}\tilde{x}_{-k}^T & 0_{1\times M-(N+L-1)}\end{bmatrix}\right)\right)^T\right)$$

Where $\alpha = \dfrac{\lambda^{-1} p_{k-1}}{1+\lambda^{-1}\tilde{x}_k^T p_{k-1}\tilde{x}_k}$, and

$\tilde{x}_{-k} = [\mathrm{x}_{kN+N-1} \quad \mathrm{x}_{kN+N-2} \ \ldots \mathrm{x}_{kN-L+1}]^T$.

$\Re$ is a matrix of size $(L \times M)$ which makes it possible to eliminate the first $(M-L)$ vector components introduced during the inverse transformation. It is defined by:

$$\Re = \left[0_{L\times(M-L)} \quad I_L \right] \tag{23}$$

I_L is the identity matrix of size $(L \times L)$.

By computing, at each iteration k, the inverse autocorrelation matrix according given the equation:

$$P_k = \lambda^{-1}\left(P_{k-1} - G_k \left[\tilde{x}_k^T . P_{k-1} \right] \right) \tag{24}$$

$$P_0 = \delta^{-1}.I_M$$

Where I_M is the identity matrix of size $(M \times M)$.

By using of the FFT, the calculation of the error and the update of the coefficients of the block filter has become cheaper than in the case of temporal BRLS.

4. RESULTS OF SIMULATIONS OF THE FFT IMPLEMENTATION OF THE PROPOSED BRLS ALGORITHM

The BRLS algorithm that we proposed was developed to allow its implementation using the Fast Fourier Transform (FFT). Indeed, an implementation of this algorithm, based on the cyclic convolution property, will then lead to the use of the FFT of length $M = 256$ for an impulse response of $L = 128$ coefficients and for a sample block of length $N = 64$.

Several simulations have been carried out to evaluate the performance of the BRLS algorithm. For these tests, the FFT-BRLS algorithm is compared with the other algorithms, RLS and Toeplitz-BRLS.

The values of the various parameters used in the simulation are given by: $\lambda = 1$ et $\delta = 8$. $N = 64$ is the block size of the input sequence.

The simulations are performed using the same input signal and the same echo path impulse response used later (Figure 2).

4.1. Convergence of adaptive filter coefficients

One of the performance criteria used in this simulation is the convergence, N_m, of adaptive filter coefficients:

$$N_m = 10log_{10}\left[\frac{\|w-\hat{w}\|^2}{\|w\|2}\right] \qquad (25)$$

Where w and $\hat{w}$ denote respectively the impulse response and the estimated impulse response of the echo path.

This criterion of convergence of the coefficients of the adaptive filter is illustrated in figure (6). This figure shows that the two algorithms, FFT-BRLS (dashed linear curve) and Toeplitz-BRLS (continuous linear curve) have the same result in terms of the convergence rate of the coefficients of the adaptive filter and consequently in terms of having the same residual echo at the output of the acoustic echo cancellation system. Moreover, the simulations do not reveal any noted difference in terms of convergence between the two adaptation methods (sample processing represented by the RLS algorithm and block processing). The difference between the two methods lies rather in the complexity of computation and the execution time.

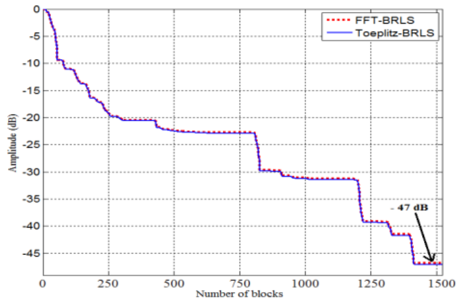

Figure 6. Convergence of the adaptive filter coefficients using the sample block processing method

4.2. Calculation Complexity

This section is devoted to the study of the computational complexity of three adaptive algorithms: RLS, Toeplitz-BRLS and FFT-BRLS. We will present the number of operations (additions, multiplications, subtractions and divisions) necessary to implement the three algorithms in question.

As mentioned in paragraph 2.2.2, the number of additions/subtractions and multiplications/divisions implemented for the RLS algorithm at each iteration k, is respectively of the order $2L^2+1$, and $3L^2+7L$. Consequently, for processing the echo signal of our study, which is composed of **97408 samples**, this algorithm needs **97408 iterations**.

The FFT-BRLS algorithm requires two convolution products as shown in the two equations (21) and (22). The computation of each convolution product requires the computation of three transforms, two direct (FFT) and one inverse $\left(FFT^{-1}\right)$. The number of additions and multiplications implemented for a FFT or for one FFT^{-1} is of the order of $M\log_2 M$ and $\frac{M}{2}\log_2 M$, where M is the size of the transform.

The figure (7) provides a comparison in terms of the number of additions and multiplications necessary for the implementation of the two adaptation methods, RLS and FFT-BRLS as a function of N. This comparison is carried out from a sequence of length $M = 256$.

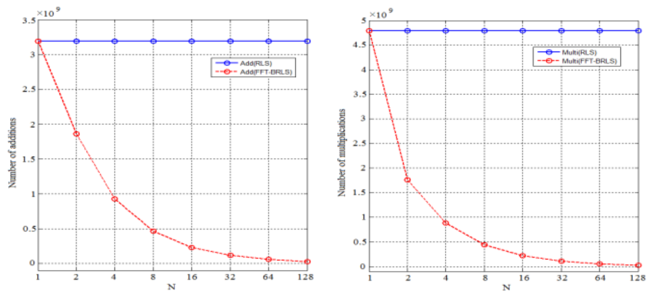

Figure 7. Computation complexity of two RLS and FFT-BRLS algorithms as a function of *N* and for *L* = 128

The curves show that, as N increases, the number of calculations decreases for the FFT-BRLS algorithm, and the more the benefit of FFT implementation will be marked. If N = 1, this corresponds to the RLS algorithm. In this particular case where N = 1, the FFT-BRLS algorithm has the same number of operations as that of the RLS algorithm.

4.3. Execution Time

The performance of the block-processed RLS algorithm is intended to compare this time the execution time of both the RLS and the FFT-BRLS algorithms to highlight the advantage of using the block processing method using the Fast Fourier transform (FFT). In this comparison, the length of the blocks varies between N=1 (this is the particular case that corresponds to the RLS algorithm) and N=1024. The figure (8) shows an example of this comparison.

This figure shows that the execution time of the FFT-BRLS algorithm depends on the values of N. It also shows that the reduction of the execution time is observed when N increases. On the other hand, the execution time represented by the RLS algorithm is generally the same, is equal to **t = 129.15 seconds** in order to process **97408 samples** (number of samples of the echo signal).

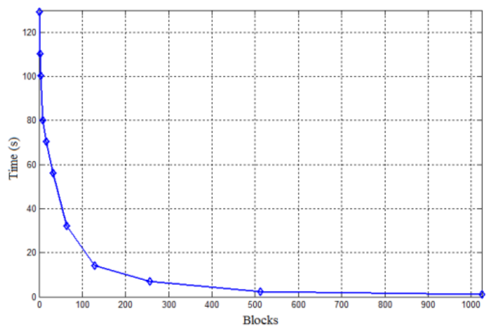

Figure 8. Execution time of the FFT-BRLS algorithm as a function of *N*

5. CONCLUSION

The work carried out and described in this article provides the first elements to evaluate the possibility of developing an acoustic echo cancellation system. First, we have presented the origin of the acoustic echo and in what situation this acoustic echo is present in an annoying way. The two adaptive algorithms, LMS and RLS, used in the acoustic echo cancellation system and their comparison at the level of the convergence of their coefficients have been described. This comparison led us to conclude that the RLS algorithm has a better acoustic echo attenuation compared to the LMS algorithm. On the other hand, the implementation of the RLS algorithm presents a significant global computational complexity compared to the LMS algorithm. To remedy this problem, we proposed to treat it in blocks of samples instead of treating it sample by sample. This process involves a series of convolution products that can be realized with reduced computational complexity by implementing the Fast Fourier Transform (FFT).

The results of the simulations reveal no noted difference in terms of convergence between the two adaptation methods (sample processing represented by the RLS algorithm and block processing). The only difference between the two methods resides in the computational complexity and the execution time, which was reduced by the new FFT-BRLS algorithm.

REFERENCES

1. [Gilloire 1987] Gilloire A., and Julien J.P. (1987). L'acoustique des salles dans les télécommunications. L'écho des recherches, n° 127, p. 43-54.
2. [Kazuo 1990] Kazuo M, Shigeyuki U, and Fumio A (1990). Echo Cancellation and Applications. IEEE Communications Magazine, p.49-55.
3. [Farhang 1998] Farhang-Boroujeny B (1998). Adaptive Filters. Theory and Apllications. Wiley & Sons, ISBN 978-0-471-98337-8.
4. [George 1999] George-Othon G, Kostas B, and Sergios T (1999). Efficient Least Squares Adaptive Algorithms for FIR Transversal Filtering. IEEE Signal Processing Magazine, p. 13-41.
5. [Paleologu 2010] Paleologu C, Benesty J, and Ciochina S (2010). Sparse Adaptive Filters for Echo Cancellation. Morgan & Claypool Publishers,
6. [Ljung 1983] Ljung L and Söderström T (1983). Theory and Practice of Recursive Identification. M.I.T.Press.
7. [Benesty 2011] Benesty J, Paleologu. C, and Ciochina S (2011). Regularization of the RLS algorithm. IEICE Trans. Fundamentals, vol. E94-A, p. 1628-1629.

8. [Gilloire 1994] Gilloire A, and Vetterli M (1994). Performance evaluation of acoustic echo controls: required values and measurement procedures. Annales des Télécommunications. vol 49, n°7, p. 368-372.
9. [Clark 1981] Clark G.A, Mitra S.K, and Parker S.R (1981). Block implementation of adaptive digital filters. IEEE Trans. on Circuits and Systems. vol. CAS-28, p. 585-592.
10. [Alaeddine 2012] Alaeddine H, Bazzi O, Alaeddine A, Mohanna Y and Burel G (2012). Fast convolution using generalized sliding Fermat number transform with application to digital filtering. Journal of The Institute of Electronics, Information and Communication Engineers (IEICE), E95-A (6), p.1007-1017.
11. [Khong 2005] Khong A, Benesty J, and Naylor P.A (2005). An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation, in Proc. EUSIPCO Actes du colloque Sémio 2005, Antalya, Turkey.

APPENDIX: RECURSIVE LEAST SQUARES (RLS) ALGORITHM

We will develop a recursive algorithm which, using the coefficients of the filter at the instant $k-1,$ will estimate these coefficients at the instant using the new available data.

Problem Statement

Our objective is to estimate the parameters $\hat{w}_k$ using the following least squares criterion:

$$\zeta_k = \sum_{n=0}^{k} \lambda^{k-n} \left(d_n - \hat{w}_k^T x_n\right)^2 \tag{A.1}$$

With λ is a weighting factor that always takes a positive value: 0 $\langle\langle \lambda \leq 1.$ This factor is also called forgetting factor because it is used to forget data that corresponds to a distant past. The particular case $\lambda = 1$ corresponds to an infinite memory.

The normal equations

The problem is to determine the vector of the coefficients $\hat{w}_k$ which minimizes ζ_k. The solution is obtained by calculating the derivatives of the cost function ζ_k and by equating them to zero:

$$\sum_{n=0}^{k} \lambda^{k-n} \left[d_n - \hat{w}_k^T x_n \right] x_n = 0_{L\times 1} \qquad \text{(A.2)}$$

Let:

$$\sum_{n=0}^{k} \lambda^{k-n} d_n \; x_n = \sum_{n=0}^{k} \lambda^{k-n} \left[x_n \, x_n^T \right] \hat{w}_k \qquad \text{(A.3)}$$

It then comes:

$$Q_k = \Re_k \hat{w}_k \qquad \text{(A.4)}$$

$$\hat{w}_k = \Re_k^{-1} Q_k \qquad \text{(A.5)}$$

With:

$$\Re_k = \sum_{n=0}^{k} \lambda^{k-n} x_n \, x_n^T \qquad \text{(A.6)}$$

$$Q_k = \sum_{n=0}^{k} \lambda^{k-n} \, d_n \, x_n \qquad \text{(A.7)}$$

The above equations can be computed recursively

$$\Re_k = \lambda \Re_{k-1} + x_k \, x_k^T \qquad \text{(A.8)}$$

We have also:

$$Q_k = \lambda Q_{k-1} + d_k \, x_k \qquad \text{(A.9)}$$

RLS algorithm

It is recalled that the correlation matrix is:

$$\Re_k = \lambda \Re_{k-1} + x_k \, x_k^T \qquad \text{(A.10)}$$

The inversion lemma can be used to compute the inverse of $\Re_k$, let:

$A = \Re_k$, $B^{-1} = \lambda \Re_{k-1}$, $C = x_k$ and $D = 1$

Thus we obtain the following recursive equation for the inverse of the correlation matrix:

$$\Re_k^{-1} = \lambda^{-1}\,\Re_{k-1}^{-1} - \frac{\lambda^{-2}\,\Re_{k-1}^{-1}\,x_k\,x_k^T\,\Re_{k-1}^{-1}}{1+\lambda^{-1}\,x_k^T\,\Re_{k-1}^{-1}\,x_k} \qquad \text{(A.11)}$$

Let:

$$P_k = \Re_k^{-1} \quad \text{and} \quad G_k = \frac{\lambda^{-1}\,P_{k-1}\,x_k}{1+\lambda^{-1}\,x_k^T\,P_{k-1}\,x_k} \qquad \text{(A.12)}$$

Then the recursive equation for P_k can again be written:

$$P_k = \lambda^{-1}P_{k-1} - \lambda^{-1}\,G_k\,x_k^T\,P_{k-1} \qquad \text{(A.13)}$$

The vector G_k is called **Kalman gain**, which can be rearranged:

$$G_k = P_k\,x_k \qquad \text{(A.14)}$$

Now, we need to compute the filter coefficients recursively. We have:

$$\hat{w}_k = \Re_k^{-1}\,Q_k = P_k\,Q_k = \lambda\,P_k\,Q_{k-1} + P_k\,x_k\,d_k \qquad \text{(A.15)}$$

By replacing P_k in the first term of the preceding expression by equation (A.13), we obtain:

$$\begin{aligned}\hat{w}_k &= P_{k-1}\,Q_{k-1} - G_k\,x_k^T\,P_{k-1}\,Q_{k-1} + P_k\,x_k\,d_k \\ &= \hat{w}_{k-1} + G_k\left(d_k - x_k^T\,\hat{w}_{k-1}\right)\end{aligned} \qquad \text{(A.16)}$$

Finally, we have:

$$\hat{w}_k = \hat{w}_{k-1} + G_k\,e_k \qquad \text{(A.17)}$$

Where

$$e_k = d_k - x_k^T\,\hat{w}_{k-1} \qquad \text{(A.18)}$$

is the a priori error that is different from the posterior error:

$$e_k = d_k - x_k^T\,\hat{w}_k \qquad \text{(A.19)}$$