

# A New Adaptive Filter Approach for Acoustic Echo Canceller in Teleconference Systems

*Hamze Haidar Alaeddine*

*Ali Beydoun*

HKS Laboratory, Electronics and Physics Dept.,  
Faculty of Sciences I, Lebanese University, Lebanon

Doi: 10.19044/esj.2018.v14n27p142 [URL:http://dx.doi.org/10.19044/esj.2018.v14n27p142](http://dx.doi.org/10.19044/esj.2018.v14n27p142)

---

## Abstract

A flexible Frequency domain Block Recursive Least Squares (FBRLS) algorithm using the Multi-Delay Filter (MDF) is presented throughout this paper. In term of performances, the MDF-FBRLS adaptive filter introduces smaller block delay and is usually faster and suitable for ideal time-varying system such as an acoustic echo in a teleconference room. The implementation of the FBRLS algorithm using MDF adaptive filter allows reducing the FFT size and consequently optimizing the hardware implementation that could be performed using standard DSP chips. These good performances are achieved by using smaller block size and updating frequently the weight vectors which will reduce the total execution time of the adaptive process. Simulation results show that the MDF-FBRLS algorithm is better than the FBRLS algorithm in terms of the total execution time and the efficiency of the computational complexity.

---

**Keywords:** Adaptive filter, Block RLS, MDF

## Introduction

Adaptive digital filters have become increasingly popular due to their intelligent nature of processing signals on one hand, and the emergence of a family of powerful digital signal processors has enhanced their tasks on the other hand (Farhang, 1998 ; Benesty, 2001). There are many adaptive algorithms available; each has its own merits and special applications. However, for applications such as an acoustic echo canceller in teleconference systems, which requires filter lengths of several hundreds or thousands of coefficients, the frequency domain block adaptive filter based on the Recursive Least Squares (FRLS) algorithm is considered to be most suitable (Stanciu, 2013 ; Clark, 1981). This is because the FRLS adaptive filter implements the block RLS (BRLS) algorithm efficiently by using the Fast

Fourier Transform (FFT) (Clark, 1981 ; Alaeddine 2012 ; Alaeddine 2017). In so doing, a significant reduction in computational load for the same adaptation performance is achieved. However, a few practical implementation problems of the FBRLS adaptive filter have hindered its applications. There are as follows:

**Inefficient Use of Hardware:** Most of the available FFT or DSP chips are designed and optimized for small FFT size, typically 256 points. To implement an acoustic echo canceller for a few thousands taps, several FFT chips are cascaded together with external memory to form a larger FFT configuration, which is inefficient and expensive.

**Long Block Delay:** Since the FBRLS algorithm implements block processing, for example, for the weight size  $M=1024$ , the first output of the adaptive filter needs to wait 128 ms for an 8 kHz sampling rate, after that the last output of the same block is processed. Such a long delay would make the echo more annoying.

**Large Quantization Error in FFT:** As the size of an FFT becomes larger, the number of multiplications and scaling increases. This causes extra quantization error.

With these limitations in mind, we present a more flexible frequency domain (MDF-FBRLS) algorithm. The performance of this algorithm is compared to the existing frequency domain FBRLS. It is found that by using a small FFT size and updating the weights more often, the MDF-FBRLS algorithm has a shorter block delay, and its computational complexity proved to be more efficient with smaller memory requirements.

The rest of the paper is organized as follows. In second section, we will introduce the general diagram of the MDF adaptive filter. Third Section, presents the proposed MDF-FBRLS algorithm. Then, simulation results and computational complexity estimation of the new algorithm are presented in section four and compared to those obtained with the FBRLS algorithm.

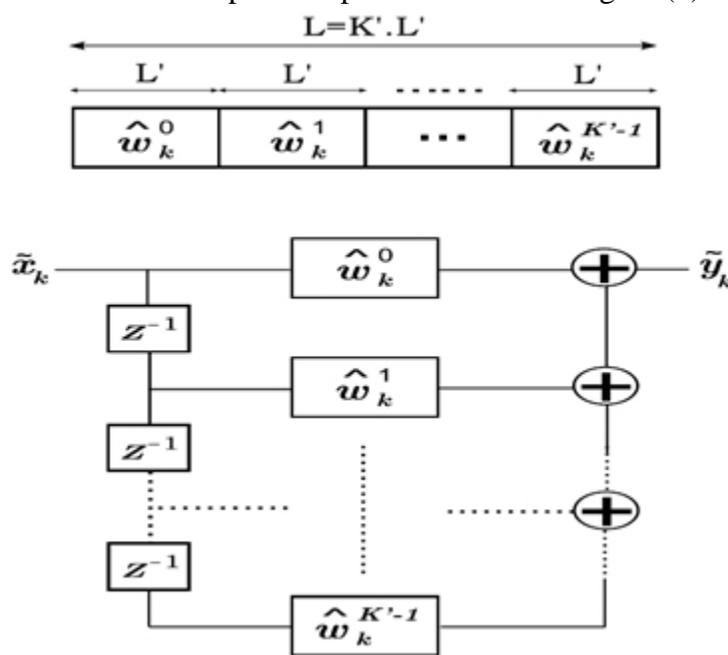
### The MDF adaptive filter

The MDF adaptive filter is initially developed to implement the FBLMS algorithm (Soo, 1990). The MDF algorithm is basically a block frequency domain adaptive filtering procedure, which consists of segmenting the filter's impulse response  $\hat{W}_k$  of length  $L$  in  $K'$ , ( $K' \in \mathbb{N}$ ), short successive

segments  $\hat{w}_k^{k'}$  of length  $L'$  ( $L = K'.L'$ ). We define then  $K'$  temporarily sub-vectors  $\hat{w}_k^{k'}$  of size as:

$$\hat{w}_k = \begin{bmatrix} \hat{w}_k^0 \\ \vdots \\ \hat{w}_k^{K'-1} \end{bmatrix} \tag{1}$$

The segmentation of the impulse response is shown in Figure (1).



**Figure 1.** Decomposition of the impulse response  $\hat{w}_k$  into  $K'$  vectors  $\hat{w}_k^{k'}$

The  $k^{th}$  segment  $\hat{w}_k^{k'}$  of the impulse response  $\hat{w}_k$  is given by the following expression:

$$\hat{w}_k^{k'} = [\hat{w}_k(k'L') \quad \hat{w}_k(k'L'+1) \dots \hat{w}_k((k'+1)L'-1)]^T \tag{2}$$

The first step of this algorithm is to convert the most recent overlapped input samples to the frequency domain via FFT as (Clark, 1981 ; Soo 1990 ; Buchner 2006):

$$\tilde{X}_k^{k'} = FFT(\tilde{X}_k^{k'}) = FFT\left(\left[\tilde{x}_k^{k'T} \mid 0_{1 \times (M'-(N+L'-1))}\right]^T\right) \tag{3}$$

$$\tilde{\mathbf{x}}_k^{k'} = [\mathbf{x}_{kN-k'L'-L'+1} \cdots \mathbf{x}_{kN-k'L'+N-1}]^T, \quad k' = 0, 1, \dots, K'-1$$

The output  $\tilde{y}_k$  and the error vector  $\mathbf{e}_k$  can be expressed as:

$$\tilde{y}_k = \text{last } N \text{ terms of } \left\{ FFT^{-1} \left( \sum_{k'=0}^{K'-1} (\tilde{\mathbf{x}}_k^{k'} \bullet \tilde{\mathbf{w}}_k^{k'}) \right) \right\} \quad (4)$$

$$\tilde{\mathbf{e}}_k = FFT \left( \left[ (\tilde{d}_k - \tilde{y}_k)^T \mid \mathbf{0}_{1 \times (M'-N)} \right] \right) \quad (5)$$

Where the operator  $\bullet$  denotes the term by term multiplication.  $\tilde{\mathbf{w}}_k^{k'}$  is the  $k^{th}$  weight vector and  $\tilde{d}_k$  is the desired vector. The updated weight equations based on the MDF-FBLMS algorithm are given by:

$$\hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \mu_B \cdot \tilde{\theta}_k^{k'}, \quad 0 \leq k' \leq K'-1 \quad (6)$$

$$\tilde{\theta}_k^{k'} = \text{last } L' \text{ terms of } \left\{ FFT^{-1} (\tilde{\mathbf{x}}_{-k}^{k'} \bullet \tilde{\mathbf{e}}_k^{k'}) \right\} \quad (7)$$

Where  $\mu_B$  is the block step size and  $\tilde{\mathbf{x}}_{-k}^{k'}$  is defined by:

$$\tilde{\mathbf{x}}_{-k}^{k'} = FFT \left( \left[ \tilde{\mathbf{x}}_{-k}^{k'T} \mid \mathbf{0}_{1 \times (M'-(N+L'-1))} \right]^T \right) \quad (8)$$

$$\tilde{\mathbf{x}}_{-k}^{k'} = [\mathbf{x}_{kN-k'L'+N-1} \cdots \mathbf{x}_{kN-k'L'-L'+1}]^T \quad (9)$$

Note that the input/output operations of the MDF adaptive filter is identical to the FBLMS, except now the FFT is  $M'$ -points long. The FBLMS adaptive filter can, in fact, be regarded as the special case of MDF with  $K'=1$ .

Figure (2) illustrates the block diagram of the MDF-FBLMS implementation showing the different steps of the weight update using a typical example. It is important to note that the calculation is performed in parallel using  $K'$  sub-vectors which reduce the FFT size and consequently the execution time.

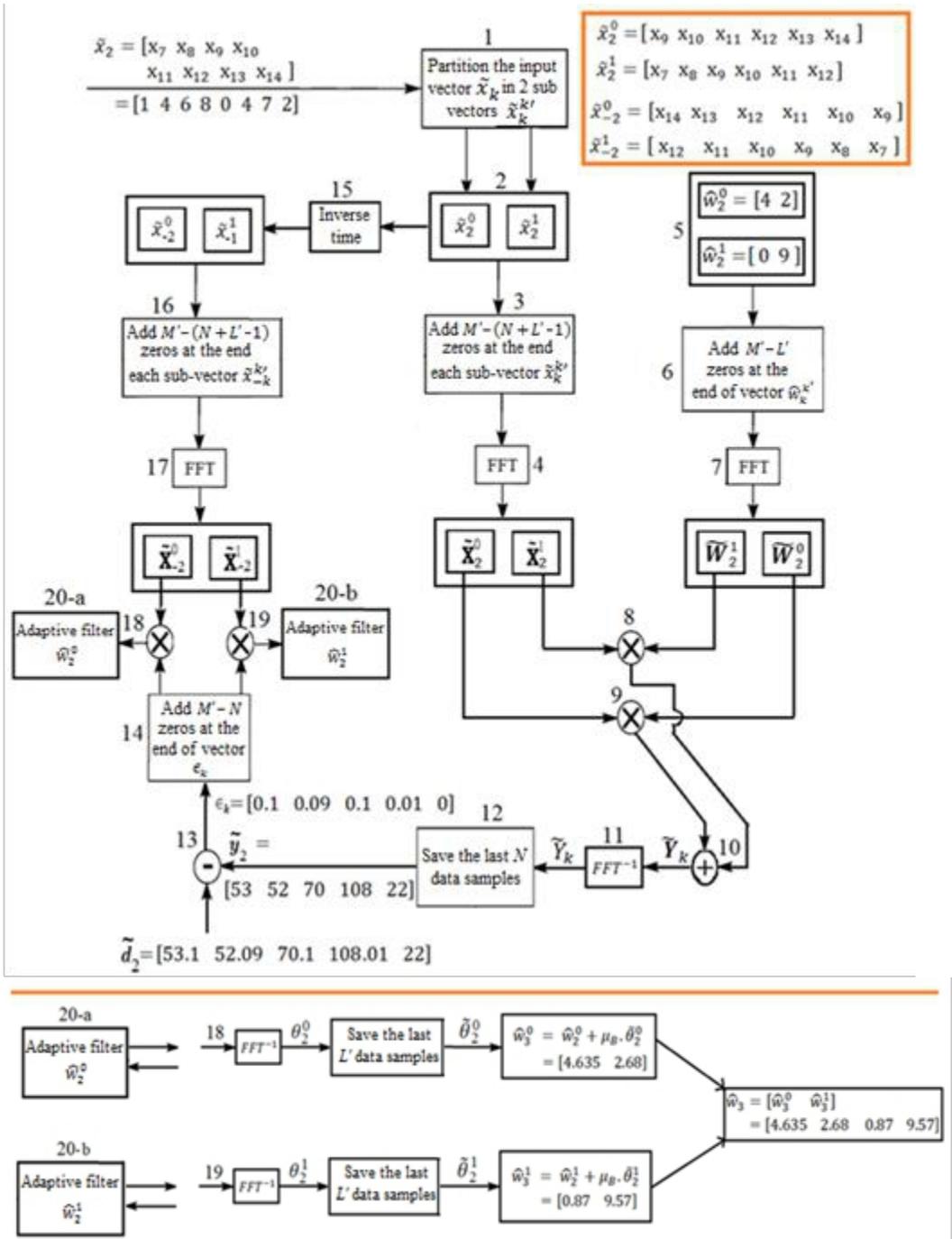


Figure 2. Procedure implementation of MDF-FBLMS adaptive filter for  $N = 5$ ,  $L = 4$ ,  $K' = 2$  and  $\mu_B = 0.5$

**A multi-delay FBRLS algorithm**

The weight vector  $\hat{w}_k$  of FBRLS algorithm is given by (Alaeddine 2012; Alaeddine 2017):

$$\begin{aligned} \hat{w}_{k+1} &= \hat{w}_k + \frac{\lambda^{-1} P_{k-1}}{1 + \lambda^{-1} \tilde{x}_k^T P_{k-1} \tilde{x}_k} \cdot (\tilde{x}_{-k} * \varepsilon_k) \\ &= \hat{w}_k + \frac{\lambda^{-1} P_{k-1}}{1 + q_k} \cdot (\tilde{x}_{-k} * \varepsilon_k) = \hat{w}_k + \alpha_k (\tilde{x}_{-k} * \varepsilon_k) \end{aligned} \tag{10}$$

Where  $\lambda$  is the forgetting factor,  $q_k = \lambda^{-1} \tilde{x}_k^T P_{k-1} \tilde{x}_k$ , and  $P_k$  is the inverse correlation matrix of size  $(M \times M)$ :

$$\begin{aligned} P_k &= \lambda^{-1} \left( P_{k-1} - G_k \left( \tilde{x}_k^T P_{k-1} \right) \right) \\ G_k &= \alpha_k \cdot \tilde{x}_k \text{ represents the Kalman gain.} \end{aligned} \tag{11}$$

The fast convolution denoted by  $*$ , between the vectors  $\tilde{x}_{-k}$  and  $\varepsilon_k$  can be calculated in the frequency domain by:

$$\begin{aligned} \phi_k &= \tilde{x}_{-k} * \varepsilon_k \\ &= \text{last } L \text{ terms of } \left\{ FFT^{-1} \left( \tilde{\mathbf{X}}_{-k} \bullet \tilde{\mathbf{E}}_k \right) \right\} \end{aligned} \tag{12}$$

The error vector  $\varepsilon_k$  is similarly obtained as the MDF-FBLMS algorithm (Eq. 4 and 5).

The first step of the MDF-FBRLS algorithm (Alaeddine 2018) consists of dividing the vector  $\alpha_k (\tilde{x}_{-k} * \varepsilon_k)$  into  $K'$  sub-vectors of smallest size not as previously processed. Then, the main idea of the proposed algorithm is to decompose the two sequences  $\tilde{x}_{-k}$  and  $\varepsilon_k$  without overlapping, unlike the previous MDF (Alaeddine 2017), according to the following relation:

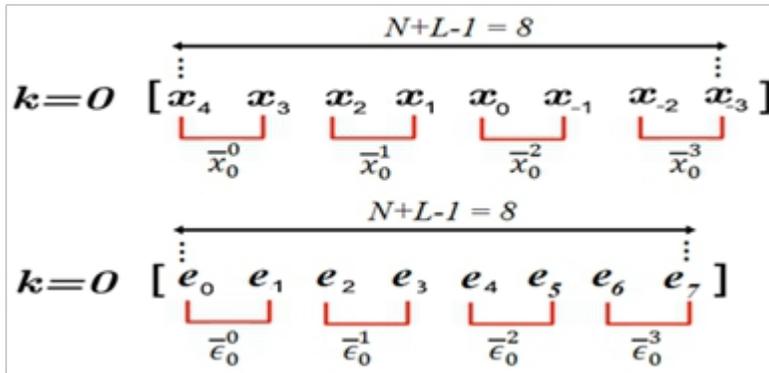
$$\begin{aligned} \tilde{x}_{-k} &= [\bar{x}_{-k}^0 \quad \bar{x}_{-k}^1 \quad \dots \quad \bar{x}_{-k}^{K'-1}]^T \\ \varepsilon_k &= [\bar{\varepsilon}_k^0 \quad \bar{\varepsilon}_k^1 \quad \dots \quad \bar{\varepsilon}_k^{K'-1}]^T \end{aligned}$$

Where  $\bar{x}_{-k}^{k'}$  and  $\bar{\varepsilon}_k^{k'}$  are two vectors of length  $(M / K') < M'$  defined by:

$$\bar{x}_{-k}^{k'} = \left[ \mathbf{x} \left( \frac{(k'+1)M}{K'} - 1 \right) \quad \mathbf{x} \left( \frac{(k'+1)M}{K'} - 2 \right) \quad \dots \quad \mathbf{x} \left( \frac{k'M}{K'} \right) \right]^T$$

$$\bar{\varepsilon}_k^{k'} = [e_{\binom{k'M}{K'}} e_{\binom{k'M}{K'}+1} \dots e_{\binom{(k'+1)M}{K'}-1}]^T$$

The proposed decomposition procedure, without overlapping, is shown in Figure (3) for  $M = 8, N = 5$  and  $L = K' = 4$ .



**Figure 3.** Decomposition of two vectors,  $\tilde{x}_{-k}$  and  $\varepsilon_k$  into  $K' = 4$  sub-vectors,  $\bar{x}_{-k}^{k'}$  and  $\bar{\varepsilon}_k^{k'}$  for  $M = 8, N = 5$  and  $L = 4$

The originality beyond this choice resides in the problem presented by the matrix multiplication between both coefficients  $\alpha_k^{k'}$  and  $\psi_k^{k'} = (\bar{x}_{-k}^{k'} * \bar{\varepsilon}_k^{k'})$ . Actually, the calculation of the FFT of both sequences  $\tilde{x}_{-k}$  and  $\tilde{x}_{-k}^{k'}$  of length  $M$  and  $M'$  respectively based on the BLMS and MDF-FBLMS algorithms, shows a result with common parts of samples.

Consequently, the convolution of  $\tilde{\theta}_k^{k'} = \tilde{x}_{-k}^{k'} * \varepsilon_k$ , multiplied by the block step size  $\mu_B$  gives a similar result of that done based on FBLMS algorithm.

On the other hand, the block step size  $\mu_B$  in the FBRLS algorithm is replaced by  $\alpha_k$  coefficient of size  $(M \times M)$ . This matrix creates a problem of calculation during the updating of the filter coefficients, by the MDF-FBRLS algorithm. Actually, the difference between the FFT of both sequences  $\tilde{x}_{-k}^{k'}$  and  $\tilde{x}_{-k}$  is  $N-1$  samples. Consequently, the result of multiplication between the two coefficients  $\alpha_k^{k'}$  and  $\psi_k^{k'}$  is not the same as obtained by the FBRLS algorithm.

Then the convolution  $\psi_k^{k'} = (\bar{x}_{-k}^{k'} * \bar{\varepsilon}_k^{k'})$  can be calculated in the frequency domain via FFT as:

$$\Psi_k^{k'} = \left( \bar{\mathbf{X}}_{-k}^{k'} \bullet \bar{\mathbf{E}}_k^{k'} \right) \tag{13}$$

The FFT and its inverse (IFFT) of a sequence of length  $M$  can be calculated by finding  $K'$  FFT, each of length  $\frac{M}{K'}$ , are proposed by the following equations:

$$\bar{\mathbf{X}}_{-k}^{k'}(m) = \sum_{n=0}^{M-1} x(n).e^{-j2\pi mn/M} \tag{14}$$

$$\begin{aligned} \psi_k^{k'}(m) &= \bar{x}_{-k}^{k'}(m) * \bar{\varepsilon}_k^{k'}(m) \\ &= \frac{1}{M} \sum_{n=0}^{M-1} \Psi_k^{k'}(m).e^{j2\pi mn/M} \end{aligned} \tag{15}$$

where  $m = k' \frac{M}{K'}, \left( k' \frac{M}{K'} \right) + 1, \dots, \left( (k'+1) \frac{M}{K'} \right) - 1$

The vector  $\bar{\mathbf{E}}_k^{k'}$  can be calculated using the same equation (14).

The originality beyond this proposal resides in the segmentation of the matrix  $\alpha_k$ , into  $K'$  temporarily matrices  $\alpha_k^{k'} (k' = 0, 1, \dots, K'-1)$ , of size  $\left( \frac{M}{K'} \times M \right)$ :

$$\alpha_k = \begin{bmatrix} \alpha_k^0 \\ \alpha_k^1 \\ \vdots \\ \alpha_k^{K'-1} \end{bmatrix} \tag{16}$$

We propose to divide each matrix  $\alpha_k^{k'}$  into  $K'$  matrices  $\alpha_k^{k',i} (i = 0, 1, \dots, K'-1)$ , each of size  $\left( \frac{M}{K'} \times \frac{M}{K'} \right)$ :

$$\alpha_k^{k'} = \begin{bmatrix} \alpha_k^{k',0} & \alpha_k^{k',1} & \dots & \alpha_k^{k',K'-1} \end{bmatrix} \tag{17}$$

Therefore, the matrix  $P_{k-1}$  must be divided also into  $K'$  temporarily matrices  $P_{k-1}^{k'}$  of size  $\left( \frac{M}{K'} \times M \right)$ :

$$P_{k-1}^{k'} = \begin{bmatrix} P_{k-1}^{k',0} & P_{k-1}^{k',1} & \dots & P_{k-1}^{k',K'-1} \end{bmatrix} \quad (18)$$

Each matrix  $P_{k-1}^{k'}$  is composed of  $K'$  matrices  $P_{k-1}^{k',i}$  of size  $\left(\frac{M}{K'} \times \frac{M}{K'}\right)$ :

$$P_{k-1}^{k',i} = P_{k-1} \left( k' \frac{M}{K'} \dots \left( (k'+1) \frac{M}{K'} \right) - 1, i \cdot \frac{M}{K'} \dots \left( (i+1) \frac{M}{K'} \right) - 1 \right) \quad (19)$$

From the above matrix decompositions, we define the calculation of the scalar  $q_k$  according to this relation:

$$q_k = \lambda^{-1} \tilde{x}_k^T \cdot (P_{k-1} \tilde{x}_k) = \lambda^{-1} \tilde{x}_k^T \cdot \beta_k \quad (20)$$

As the vector  $P_{k-1}$ , we propose to divide the vector  $\beta_k$  into  $K'$  sub-vectors  $\beta_k^{k'}$ . Each sub-vector  $\beta_k^{k'}$  is defined by:

$$\begin{aligned} \beta_k^{k'} &= P_{k-1}^{k'} \cdot \tilde{x}_k = \begin{bmatrix} P_{k-1}^{k',0} & P_{k-1}^{k',1} & \dots & P_{k-1}^{k',K'-1} \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_k^0 \\ \bar{x}_k^1 \\ \vdots \\ \bar{x}_k^{K'-1} \end{bmatrix} \\ &= \sum_{i=0}^{K'-1} P_{k-1}^{k',i} \cdot \bar{x}_k^i \end{aligned} \quad (21)$$

$$\text{where } \bar{x}_k^{k'} = \left[ \mathbf{x} \left( k' \frac{M}{K'} \right) \mathbf{x} \left( k' \frac{M}{K'} + 1 \right) \dots \mathbf{x} \left( (k'+1) \frac{M}{K'} - 1 \right) \right]^T$$

Consequently, the scalar  $q_k$  and the matrix  $\alpha_k^{k'}$  can be written in the following equations:

$$\begin{aligned} q_k &= \lambda^{-1} \tilde{x}_k^T \cdot \beta_k = \lambda^{-1} \cdot \begin{bmatrix} \bar{x}_k^0 & \bar{x}_k^1 & \dots & \bar{x}_k^{K'-1} \end{bmatrix} \cdot \begin{bmatrix} \beta_k^0 \\ \beta_k^1 \\ \vdots \\ \beta_k^{K'-1} \end{bmatrix} \\ &= \lambda^{-1} \cdot \sum_{i=0}^{K'-1} \bar{x}_k^i \cdot \beta_k^i \end{aligned} \quad (22)$$

$$\alpha_k^{k'} = \frac{\lambda^{-1} \cdot P_{k-1}^{k'}}{1 + q_k} = \frac{\lambda^{-1} \cdot [P_{k-1}^{k',0} \quad P_{k-1}^{k',1} \dots P_{k-1}^{k',K'-1}]}{1 + \lambda^{-1} \cdot \sum_{i=0}^{K'-1} \bar{x}_k^i \cdot \beta_k^i} \quad (23)$$

Taking into account the previous equations, the weight update equation of  $L'$ -points MDF-FBRLS algorithm is proposed by:

$$\hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \alpha_k^{k'} \cdot \psi_k^{k'} = \hat{w}_k^{k'} + \sum_{i=0}^{K'-1} \alpha_k^{k',i} \cdot (\psi_k^i) \quad (24)$$

The  $L = K' \cdot L'$  samples of the weight  $\hat{w}_{k+1}^{k'}$  exists in the last  $n = \frac{K' \cdot L}{M}$  products  $\alpha_k^j \cdot \psi_k$ ,  $j = n, \dots, K'-1$ . That's why, the result of each product  $\alpha_k^j \cdot \psi_k$  of size  $\frac{M}{K'}$  must be divided into  $\frac{M}{K' \cdot L'}$ , in order to have  $K'$  vectors of length  $L'$ . This technique reduces the computational complexity of the MDF-FBRLS algorithm because the first  $\frac{K'(M-L)}{M}$  products of

$\alpha_k^{k'} \cdot \psi_k, k' = 0, 1, \dots, \frac{K'(M-L)}{M} - 1$ , must be excluded, therefore:

$$\alpha_k^j = \frac{\lambda^{-1} \cdot P_{k-1}^j}{1 + q_k} = \frac{\lambda^{-1} \cdot [P_{k-1}^{j,0} \quad P_{k-1}^{j,1} \dots P_{k-1}^{j,K'-1}]}{1 + \lambda^{-1} \cdot \sum_{i=0}^{K'-1} \bar{x}_k^i \cdot \beta_k^i}, j = n, \dots, K'-1 \quad (25)$$

Consequently, the weight vector  $\hat{w}_k$  of MDF-FBRLS algorithm takes the following concluding form:

$$\hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \alpha_k^{k'} \cdot \psi_k^{k'} = \hat{w}_k^{k'} + u_k^{k'} \quad (26)$$

with  $u_k^{k'} = H_k^j (k' \cdot L' \dots ((k'+1)L') - 1)$ ,  $j = n, \dots, K'-1$  (27)

$$H_k^j = \sum_{i=0}^{K'-1} \alpha_k^{j,i} \cdot (\psi_k^i) \quad (28)$$

The update of the inverse correlation matrix  $P_k$  (Eq. 11) must have the same decomposition technique proposed in the previous equations:

$$P_k = \lambda^{-1} \left( P_{k-1} - \frac{\lambda^{-1} (P_{k-1} \tilde{x}_k) (\tilde{x}_k^T P_{k-1})}{1 + \lambda^{-1} \tilde{x}_k^T P_{k-1} \tilde{x}_k} \right)$$

$$= \lambda^{-1} \cdot \left( P_{k-1}^{k'} - \frac{\lambda^{-1} \cdot (\beta_k^{k'} \cdot \bar{\beta}_k^{k'})}{1 + \lambda^{-1} \cdot \sum_{i=0}^{K'-1} \bar{x}_k^i \cdot \beta_k^i} \right) \quad (29)$$

Each matrix  $P_{k-1}^{i,k'} (i = 0, 1, \dots, K'-1)$  of size  $\left(\frac{M}{K'} \times \frac{M}{K'}\right)$  is defined as follow:

$$P_{k-1}^{k'} = \begin{bmatrix} P_{k-1}^{0,k'} \\ P_{k-1}^{1,k'} \\ \vdots \\ P_{k-1}^{K'-1,k'} \end{bmatrix}$$

$$P_{k-1}^{i,k'} = P_{k-1} \left( i \cdot \frac{M}{K'} \dots \left( (i+1) \frac{M}{K'} \right) - 1, \quad k' \cdot \frac{M}{K'} \dots \left( (k'+1) \frac{M}{K'} \right) - 1 \right) \quad (30)$$

Taking into account the matrix  $P_k$ , we define the vector  $\bar{\beta}_k$  as follow:

$$\bar{\beta}_k = [\bar{\beta}_k^0 \quad \bar{\beta}_k^1 \dots \bar{\beta}_k^{K'-1}] \quad (31)$$

$$\begin{aligned} \text{Where } \bar{\beta}_k^{k'} &= \tilde{x}_k^T \cdot P_{k-1}^{k'} = \begin{bmatrix} \bar{x}_k^{0T} & \bar{x}_k^{1T} & \dots & \bar{x}_k^{K'-1T} \end{bmatrix} \cdot \begin{bmatrix} P_{k-1}^{0,k'} \\ P_{k-1}^{1,k'} \\ \vdots \\ P_{k-1}^{K'-1,k'} \end{bmatrix} \\ &= \sum_{i=0}^{K'-1} \bar{x}_k^{iT} \cdot P_{k-1}^{i,k'} \end{aligned} \quad (32)$$

The proposed algorithm MDF-FBRLS, allows:

- The segmentation of each vectors of length  $M$ , without being overlapping unlike the previous one, into  $K'$  sub-vectors of smaller length than  $MDF \ ((M / K') < M')$ .
- The segmentation of each matrix of size  $(M \times M)$  into  $K'$  matrices of smaller size  $\left(\frac{M}{K'} \times \frac{M}{K'}\right)$ .
- The parallel computing of the  $K'$  sub-vectors and matrices, which furthermore reduces the execution time of the proposed algorithm.

### Simulation results

Numerical simulations have been conducted to evaluate the performance of the proposed MDF-FBRLS algorithm. In these tests, the MDF-FBRLS algorithm is compared to the FBRLS algorithm through Matlab software.

The criteria used to evaluate the performance of both FFT implemented FBRLS and MDF-FBRLS algorithms are compared as follows:

- The Echo Return Loss Enhancement (ERLEC) of the compensator

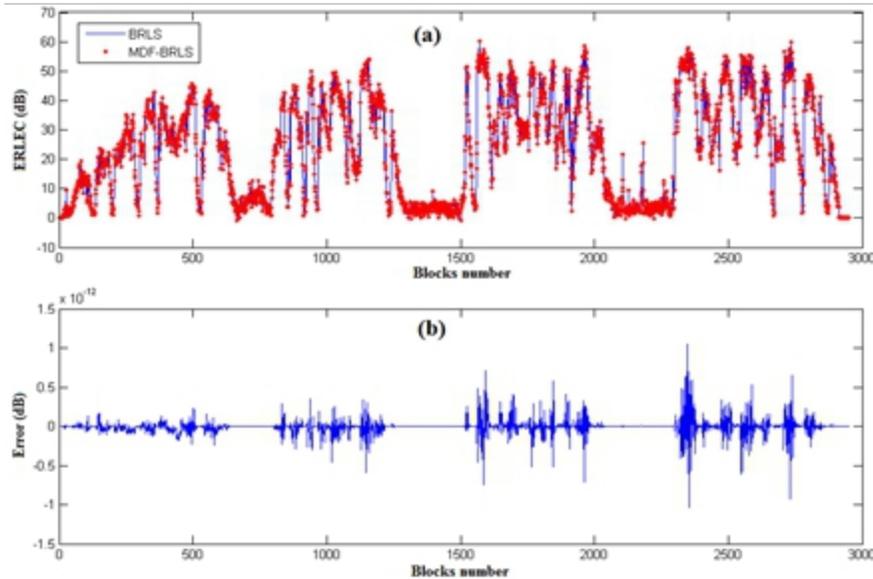
$$\hat{w}_k.$$

$$ERLEC = 10 \cdot \log_{10} \left[ \frac{\sum_{n=(k-1)N+1}^{k \cdot N} (d_n)^2}{\sum_{n=(k-1)N+1}^{k \cdot N} (d_n - \hat{y}_n)^2} \right]$$

- The impulse response of the echo path compared to the adaptive filter's coefficients.
- The execution time for the both algorithms to compute FFT.
- The computational complexity.

The two algorithms are investigated in single-talk situation where empirical values for the parameters are chosen as  $N = 129$ ,  $K' = 32$ ,  $M = 256$ ,  $L = 4$ , and  $\delta = 8$ . The dimension of the impulse response is  $L = 128$ , which corresponds a delay of 16 ms for a sampling rate of 8 kHz. In single-talk situation, 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 delay exceeding 25 ms.

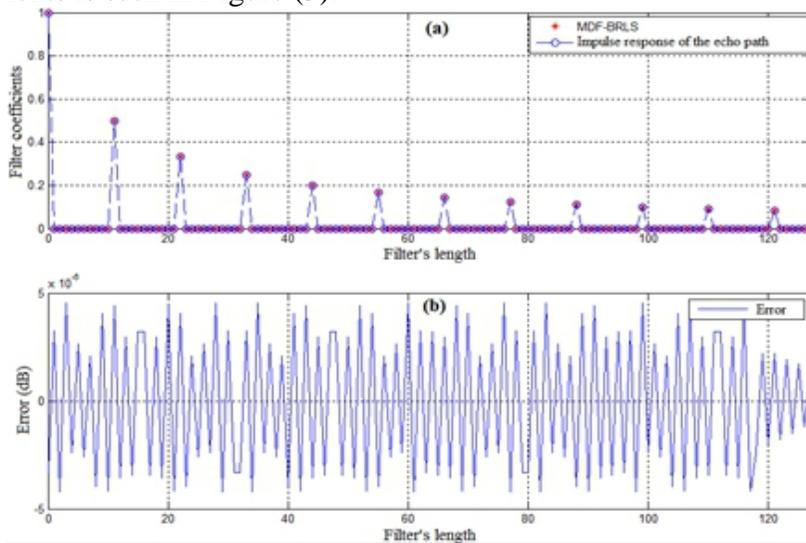
The performance of the MDF-FBRLS algorithm is measured using the ERLEC equation and seen in Figure (4).



**Figure 4.** (a) The Echo Return Loss Enhancement (ERLEC) of the compensator  $\hat{W}_k$ . (b) Difference of both algorithm in term of ERLEC

It can be noticed that both algorithms do not reveal any significant difference in terms of ERLEC (error difference is around  $10^{-12}$  dB). In our case, both algorithms provide an echo reduction of about 48 dB for a delay of 16 ms.

The impulse response of the echo path compared to the adaptive filter's coefficients is seen in Figure (5).



**Figure 5.** (a) Impulse response of the echo path. (b) Difference between the echo path and the MDF-FBRLS's filter coefficients

The result obtained in Figure (5), presents a perfect construction of echo’s path by using the MDF-FBRLS algorithm. Which means that the residual echo is unheard at the output of the echo cancellation system.

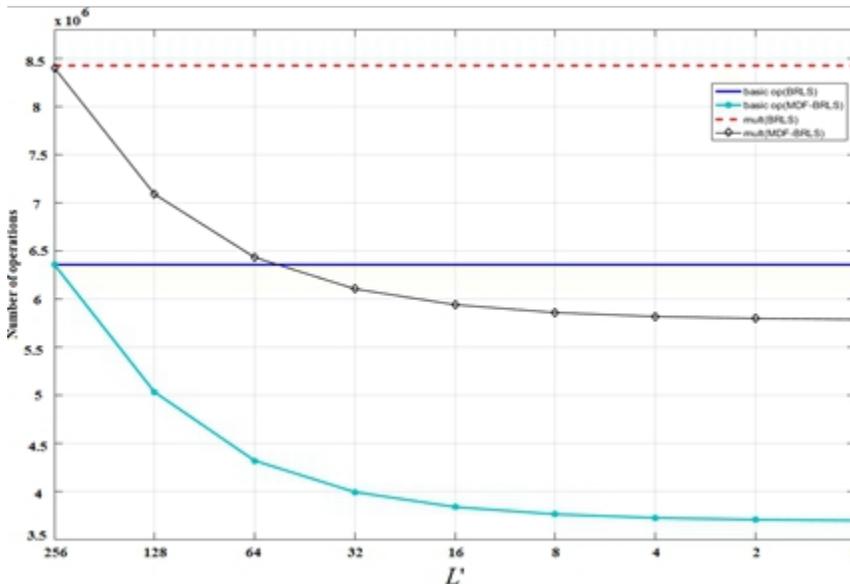
Based on the obtained values, seen in table (1) and in figure (6), the differences between both algorithms reside mainly in two points, which are the execution time and the computational complexity of the filter. As seen in the table, with the increase of  $K'$ , the length ( $L' = L / K'$ ) of MDF-FBRLS filter decreases along with the execution time of FFT.

At  $K' = 1$ , the MDF-FBRLS algorithm behaves as the FBRLS one, and thus, it takes the highest time to execute the FFT.

The FFT of length  $M$ , requires  $\frac{M}{2}(\log_2 M)$  butterflies and each butterfly need two complex additions and one complex multiplication. The computational complexity of both algorithms with  $L' = [1 \ 2 \ 4 \ 8 \ 16 \ 32 \ 64 \ 128 \ 256]$  and  $L = 256$  is shown in Figure (6). It can be noticed that with the decrease of  $L'$ , the computational complexity of MDF-FBRLS algorithm decrease, unlike the one of FBRLS algorithm.

**Table 1.** Execution time for MDF-FBRLS algorithm to compute the FFT

$K'$	Execution time (in ms)						
	1	2	4	16	64	128	256
MDF-FBRLS	7.103	2.882	1.836	0.867	0.471	0.221	0.124



**Figure 6.** Operations number of both algorithms, FBRLS and MDF-FBRLS

## Conclusion

Throughout this paper, a new algorithm to reduce enormously the time of FFT execution and the computational complexity has been presented, and simulated. The obtained algorithm finds many applications in several fields include, radar system, echo cancelation in teleconference room and in headset. Our suggestion can be developed furthermore to handle the double talk situation, and many other issues.

## References:

1. B. Farhang-Boroujeny (1998). Adaptive Filters. Theory and Applications. Wiley & Sons, ISBN 978-0-471-98337-8.
2. J. Benesty, T. Gansler, D. Morgan, M. Sondhi, and S.L. Gay (2001). Advances in Network and Acoustic Echo Cancellation. Springer-Verlag,
3. C. Stanciu, J. Benesty, J. Paleologu, T. Gansler, and S. Ciochina (2013). A widely linear model for stereophonic acoustic echo cancellation. Signal Processing, vol. 93, pp. 511-516.
4. G.A. Clark, S.K. Mitra, and S.R. Parker (1981). Block implementation of adaptive digital filters. IEEE Trans. on Circuits and Systems. vol. CAS-28, pp. 744-752.
5. H. H. Alaeddine, O. Bazzi, A. Alaeddine, and G. Burel (2012). Fast convolution using generalized sliding Fermat number transform with application to digital filtering. IEICE Journal, E95-A (6), pp.1007-1017.
6. H. H. Alaeddine, A. Ghaith, M. Al-Sahili, and M. Fadlallah (2017). New Approach of Performance Analysis of RLS adaptive Filter Using a Block-Wise Processing. EAR Journal, vol. V, pp. 1030-1053.
7. J. S. Soo and K. K. Pang (1990). Multidelay block frequency domain adaptive filter. IEEE Trans. Acoust. Speech Signal Process. vol. 38, pp. 373-376.
8. H. Buchner, J. Benesty, T. Gansler, and W. Kellermann (2006). Robust Extended Multidelay Filter and Double-Talk Detector for Acoustic Echo Cancellation. IEEE Trans. on Audio, Speech and Language Processing. vol. 14, pp. 1633-1644.
9. H. H. Alaeddine, M. Houssini, El H. Baghious and G. Burel (2018). New approach for the treatment of FBRLS algorithm with long impulse response. IEEE Mownet (pp.1-6).