A new efficient method of adaptive filter using the Galois field arithmetic

H H Alaeddine\textsuperscript{1}, K Charafeddine\textsuperscript{2} and M E Vikhrov\textsuperscript{2}

\textsuperscript{1}Electronics and Physics Department, Faculty of Sciences I - Lebanese University, HKS Laboratory, Beirut, Lebanon
\textsuperscript{2}Department of electric supply of industrial enterprises, National Research University MPEI, Moscow, Russia

E-mail: vikhrovme@mail.ru

Abstract. This paper proposes an efficiently implementation of Multi-Delay Filter Block Recursive Least Squares (MDF-BRLS) algorithm. This implementation uses a particular transform that is defined on a finite ring of integers with arithmetic carried out modulo Fermat numbers. In term of performances, this Fermat Number Transforms (FNT) is ideally suited to digital computation, requiring approximately $N \log_2 N$ additions, subtractions and bit shifts, but no multiplications unlike the Fourier transform (FFT). FNT implementation results confirm that the MDF-BRLS adaptive filter can implement with smallest computational complexity compared with an implementation using the Fourier transform.

1. Introduction

Adaptive filters are widely used in the area of signal processing such as echo cancellation, noise cancellation, channel equalization for communications and networking systems. Necessity of adaptive filters implementations is growing up in many fields. An adaptive filter is a filter that self-adjusts its transfer function according to an optimizing algorithm [1, 2]. Because of the complexity of the optimizing algorithms, most adaptive filters are digital filters that perform digital signal processing and adapt their performance based on the input signal. Adaptive filters are time-varying since their parameters are continually changing in order to meet a performance requirement [3, 4]. The implementation process requires various of performances such as good convergence characteristics, lower execution time and reduce computational complexity. However, it is difficult to satisfy these characteristics simultaneously, so efficient algorithms and efficient architectures are desired.

The design of adaptive filter algorithm is an important part within the design of adaptive filter. The fast Multi-Delay Filter Block Recursive Least Squares (MDF-BRLS) algorithm [5] is one of the efficient adaptive algorithm since here the process of filtering and adaption is done in frequency domain without overlap and occupied the smallest block size using FFT method [6–8].

The linear convolution, used by the MDF-BRLS algorithm, is one of the most important digital signal processing problems. It can be implemented more efficiently based on the Fermat number transform (FNT) than based on the fast Fourier transform (FFT) since the FNT possesses the property of cyclic convolution and low computational complexity. The expensive multiplication in the FFT and inverse (IFFT) can be replaced by bit shifts in the FNT and its inverse (IFNT) with the transform kernel 2 or its integer power [9, 10].
The principal objective of our study is to reduce the computational complexity of the MDF-BRLS algorithm. In order to ensure this, we looked further into the mathematical bases of the Number Theoretic Transform (NTT) and in particular the Fermat Number Transform (FNT), compared to the Fast Fourier Transform (FFT), which allows reduction of several multiplications which are necessary to achieve certain functions such as convolution products.

The rest of the paper is organized as follows. In second section, we will present the MDF-BRLS algorithm. Third Section presents the Number theoretic transform and the implementation technique of the MDF-BLMS algorithm, and in particular the calculation of the circular convolution, via a particular transform, called Extended-FNT. Then, simulation results and computational complexity estimation of the MDF-BRLS algorithm using the Extended-FNT are presented in section four and compared to those obtained using the FFT.

2. A multi-delay BRLS algorithm

2.1 Block recursive least squares (BRLS)

The weight vector \( \hat{w}_k \) of BRLS algorithm is given by [5]:

\[
\hat{w}_{k+1} = \hat{w}_k + \frac{\lambda^{-1} P_{k-1}^T}{1 + \lambda^{-1} \hat{x}_k^T P_{k-1} \hat{x}_k} (\hat{x}_k \ast e_k) = \hat{w}_k + \frac{\lambda^{-1} P_{k-1}^T}{1 + q_k} (\hat{x}_k \ast e_k) = \hat{w}_k + \alpha_k (\hat{x}_k \ast e_k) \tag{1}
\]

The different parameters and vectors are exposed below:

- \( \hat{w}_k = [\hat{w}_k(0), \hat{w}_k(1), \ldots \hat{w}_k(L-1)]^T \) where \( L \) represents the length of the impulse response of the filter.
- \( \hat{x}_k \) is the inverse-time of the input sequence \( \hat{x}_k = [\hat{x}_{kN-L+1}, \hat{x}_{kN-L+2}, \ldots, \hat{x}_{k(N+1)L-1}]^T \) of length \( N+L-1 \).
- \( e_k = \tilde{d}_k - \tilde{y}_k = [e_k(0), e_k(1), \ldots, e_k(N-1)]^T \) is the error vector.
- \( \lambda \) represents the forgetting factor.
- \( q_k = \lambda^{-1} \tilde{x}_k^T P_{k-1} \tilde{x}_k \), where \( P_k \) is a matrix of \( M \times M \) dimensions, and characterizes the inverse correlation matrix by the equation seen below:

\[
P_k = \lambda^{-1} \left( P_{k-1} - G_k \left( \hat{x}_k^T P_{k-1} \right) \right) \tag{2}
\]

\( M \) represents the lowest power of 2 larger than or equal to \( N+L-1 \).
- \( G_k = \alpha_k \hat{x}_k \) represents the Kalman gain.
- \( \ast \) denotes the linear convolution.

2.2 Multi-Delay Filer BRLS (MDF-BRLS)

The decomposition of the two vectors \( e_k \) and \( \tilde{x}_k \) without been overlapped was beyond the key idea of the MDF-BRLS algorithm [8], and that is on the contrary of the preceding MDF [6], [7]. That is done mainly according to the following vectors:

\[
\tilde{x}_k = \left[ \tilde{x}_k^0, \tilde{x}_k^1, \ldots, \tilde{x}_k^{N-1} \right]^T
\]

\[
e_k = \left[ \tilde{e}_k^0, \tilde{e}_k^1, \ldots, \tilde{e}_k^{N-1} \right]^T
\]
Wherein they turns, $\tilde{x}_{-k}^k$ and $\tilde{e}_k^k$ are in fact vectors of length $(M / K') < M'$ defined by:

$$\tilde{x}_{-k}^k = \left[ \begin{array}{c} x_{(k+1)M/K'}^{-1} \\ x_{(k+1)M/K'}^{-2} \\ \vdots \\ x_{(k+1)M/K'}^{-1} \end{array} \right]$$

$$\tilde{e}_k^k = \left[ \begin{array}{c} e_{(k+1)M/K'}^{-1} \\ e_{(k+1)M/K'}^{-2} \\ \vdots \\ e_{(k+1)M/K'}^{-1} \end{array} \right]$$

The exhibited problematic of the matrix multiplication among the two coefficients $\alpha_k^k$ and $\psi_k^k = (\tilde{x}_{-k}^k \ast \tilde{e}_k^k)$, and how it can be solved presents in fact the novelty of our choice. Actually, a problem of calculation is presented by the matrix $\alpha_k^k$ which is of $(M \times M)$ dimensions, once an updating of the filter coefficients through the MDF-BRLS algorithm is taken place. There are essentially $N-1$ samples difference among the FFT of the two sequences $\tilde{x}_{-k}^k$ and $\tilde{x}_{-k}^k$. Accordingly, the multiplication between $\alpha_k^k$ and $\psi_k^k$ gives a result differs from that attained by using BRLS algorithm.

Therefore, the MDF-BRLS proposes to decompose the matrix $\alpha_k^k (k' = 0,1,\ldots,K'-1)$, of size $\left( \frac{M}{K'} \times M \right)$:

$$\alpha_k^k = \left[ \begin{array}{c} \alpha_k^0 \\ \alpha_k^1 \\ \vdots \\ \alpha_k^{K'-1} \end{array} \right]$$

(3)

Then, each matrix $\alpha_k^k$ is decomposed to $K'$ matrices $\alpha_k^{k,j} (i = 0,1,\ldots,K'-1)$, of $\left( \frac{M}{K'} \times \frac{M}{K'} \right)$ size according to the following vector:

$$\alpha_k^{k,j} = \left[ \alpha_k^{k,j:0} \alpha_k^{k,j:1} \ldots \alpha_k^{k,j:K'-1} \right]$$

(4)

Also, the inverse correlation matrix $P_{k-1}$ should also be minimized into $K'$ temporarily matrices $P_{k-1}^k$ of dimension $\left( \frac{M}{K'} \times M \right)$:

$$P_{k-1}^k = \left[ \begin{array}{c} P_{k-1}^{k,0} \\ P_{k-1}^{k,1} \\ \vdots \\ P_{k-1}^{k,K'-1} \end{array} \right]$$

(5)

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

$$P_{k-1}^{k,j} = P_{k-1} \left( k'M/K' \ldots \left( (k'+1)M/K' \right)^{-1} , i'M/K' \ldots \left( (i+1)M/K' \right)^{-1} \right)$$

(6)

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

$$q_k = \lambda^{-1} \tilde{x}_k^T \left( P_{k-1} \tilde{x}_k^k \right) = \lambda^{-1} \tilde{x}_k^T \beta_k$$

(7)
As the vector $P_{k-1}$, we suggest the division of the vector $\beta_i$ into $K$ sub-vectors $\beta_k^i$. Each sub-vector $\beta_k^i$ is defined by:

$$\beta_k^i = P_{k-1}^i \cdot \tilde{x}_k = \left[ P_{k-1}^{i,0} \ P_{k-1}^{i,1} \ldots P_{k-1}^{i,K-1} \right] \cdot \left[ \begin{array}{c} x_k^0 \\ x_k^1 \\ \vdots \\ x_k^{K-1} \end{array} \right] = \sum_{i=0}^{K-1} P_{k-1}^{i,i} \cdot \tilde{x}_k$$  \hspace{1cm} (8)

where $\tilde{x}_k = \left[ x_{\left( \frac{k}{K} \right)} \ x_{\left( \frac{k}{K+1} \right)} \ldots x_{\left( \frac{k}{K} \right)} \right]^T$

Thus, $q_i$ and $\alpha_k^i$ which are consecutively scalar and matrix could be written as related by the two equations below:

$$q_k = \lambda^{-1} \tilde{x}_k \cdot \beta_k = \lambda^{-1} \left[ \begin{array}{c} x_k^0 \\ x_k^1 \\ \vdots \\ x_k^{K-1} \end{array} \right] \cdot \left[ \begin{array}{c} \beta_0^i \\ \beta_1^i \\ \vdots \\ \beta_{K-1}^i \end{array} \right] = \lambda^{-1} \sum_{i=0}^{K-1} x_k^i \cdot \beta_k^i$$ \hspace{1cm} (9)

$$\alpha_k^i = \frac{\lambda^{-1} \cdot P_{k-1}^i}{1 + q_k} = \frac{\lambda^{-1} \left[ P_{k-1}^{i,0} \ P_{k-1}^{i,1} \ldots P_{k-1}^{i,K-1} \right]}{1 + \lambda^{-1} \sum_{i=0}^{K-1} x_k^i \cdot \beta_k^i}$$ \hspace{1cm} (10)

Count on the earlier equations, the updated weight equation of $L$-points MDF-BRLS algorithm is presented now by:

$$\tilde{w}_{k+1} = \tilde{w}_k^i + \alpha_k^i \cdot \psi_{k}^i = \tilde{w}_k^i + \sum_{i=0}^{K-1} \alpha_k^i \cdot \left( \psi_{k}^i \right)$$ \hspace{1cm} (11)

The $L = K \cdot L'$ samples of weight $\tilde{w}_{k+1}$ occurs in the last $n = \frac{K' \cdot L}{M}$ products $\alpha_k^j \cdot \psi_{k}^j$, $j = n, \ldots, K'-1$. For this reason and to get $K'$ vectors of length $L'$, the result of each product $\alpha_k^j \cdot \psi_{k}^j$ of $\frac{M}{K'}$ size should be divided to $\frac{M}{K' \cdot L'}$. This technique reduces the computational complexity of the MDF-BRLS algorithm because the first $\frac{K' \cdot (M - L)}{M}$ products of $\alpha_k^i \cdot \psi_{k}^i, k = 0, 1, \ldots, K' \cdot \left( \frac{M - L}{M} \right) - 1$, must be excluded, therefore the weight vector $\tilde{w}_{k}$ of MDF-BRLS algorithm takes the following concluding form:

$$\tilde{w}_{k+1}^i = \tilde{w}_k^i + \alpha_k^i \cdot \psi_{k}^i = \tilde{w}_k^i + \sum_{i=0}^{K-1} \alpha_k^i \cdot \left( \psi_{k}^i \right)$$ \hspace{1cm} (12)

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

$$H_k^j = \sum_{i=0}^{K-1} \alpha_k^{j,i} \cdot \left( \psi_{k}^i \right)$$ \hspace{1cm} (13)
On the other hand, to get the inverse correlation matrix \( P_k \) (2) updated, a similar decomposition method of that presented in the previous equations must be respected:

\[
P_k = \lambda^{-1} \left( P_{k-1} - \frac{\lambda^{-1} \left( P_{k-1} \tilde{x}_k \right) \left( \tilde{x}_k^T P_{k-1} \right)}{1 + \lambda^{-1} \tilde{x}_k^T P_{k-1} \tilde{x}_k} \right) = \lambda^{-1} \left( P_{k-1} - \frac{\lambda^{-1} \left( \beta_k^{k-1} \tilde{\beta}_k^{k-1} \right)}{1 + \lambda^{-1} \sum_{i=0}^{K-1} \tilde{x}_k^T \beta_k^i} \right) \quad (14)
\]

Each matrix \( P^{i,k}_{k-1} \) \((i = 0, 1, \ldots, K-1)\) of size \( \frac{M}{K^i} \times \frac{M}{K^i} \) is defined as follow:

\[
P^{i,k}_{k-1} = \begin{bmatrix}
P^{0,k}_{k-1} \\
P^{1,k}_{k-1} \\
\vdots \\
P^{K^i-1,k}_{k-1}
\end{bmatrix}
\]

\[
p^{i,k}_{k-1} = P_{k-1} \left( i \frac{M}{K} \ldots \left( i+1 \frac{M}{K} \right) - 1, \ k \frac{M}{K} \ldots \left( k+1 \frac{M}{K} \right) - 1 \right) \quad (15)
\]

Taking into account the matrix \( P_k \), we define the vector \( \tilde{\beta}_k \) as follow:

\[
\tilde{\beta}_k = [\tilde{\beta}_k^0 \ \tilde{\beta}_k^1 \ \ldots \ \tilde{\beta}_k^{K-1}]
\]

\[
\tilde{\beta}_k^{k-1} = \tilde{x}_k^T P^{k-1}_{k-1} \begin{bmatrix}
\tilde{x}_k^0 \tilde{x}_k^1 \ \ldots \ \tilde{x}_k^{K-1}
\end{bmatrix} = \sum_{i=0}^{K-1} \tilde{x}_k^i \beta_k^{i,k} \quad (17)
\]

The MDF-BRLS algorithm processes the adaptive filter using the smallest block size. Consequently, it allows reducing, as much as possible, the computational complexity and the execution time of this algorithm.

3. **Extended number theoretic transform (ENTT)**

3.1 **The Fermat number transform (FNT)**

The Fermat Number Transform (FNT) belongs to class of transform known as Number Theoretic Transform (NTTs), defined over Galois Field (GF(\( \tilde{M} \))). The onward and opposite FNTs are well-defined as \( [9] \):

\[
X_k(m) = \left\{ \sum_{n=0}^{M-1} x(n) \delta^{n,m} \right\}_{F_k} \quad (18)
\]

\[
x_k(n) = \left\{ M^{-1} \sum_{m=0}^{M-1} X(m) \delta^{-n,m} \right\}_{F_k} \quad (19)
\]

\( m, n = 0, 1, \ldots, M-1. \)

Respectively, where \( F_k \) is the characteristic of the Galois Field (GF(\( F_k \))), \( M \) is the transform length,
\( \hat{\delta} \) is a root of unity of order \( M \) in \( GF(F_t) \) and \( \langle \hat{\delta} \rangle_{F_t} \) stands for residue reduction modulo \( F_t \). The characteristic, \( F_t \), is the \( t^{th} \) Fermat number, defined as \[ 9, 10: \]

\[ F_t = 2^t + 1 = 2^{2^t} + 1 \quad (20) \]

The choice of a modulo equal to a Fermat number \( F_t \) offers numerous possibilities for the length \( M \) of the transform (Table 1). The values of \( M \) and \( \hat{\delta} \) associated with a FNT are given by \( M = 2^{t+1-i} \) and \( \langle \hat{\delta} = 2^{2^i} \rangle_{F_t} \) with \( 0 \leq i < t \).

### Table 1. Constraints between \( F_t \), \( \hat{\delta} \) and \( M \).

| Modulus | Dynamic range | \( M \) |
|---------|---------------|-------|
| \( F_t \) | Bits | \( \hat{\delta} = 2 \) | \( \hat{\delta} = 3 \) | \( \hat{\delta} = \sqrt{2} \) |
| \( F_t = 2^k + 1 \) | \( 2^7 \) | 16 | 256 | 32 |
| \( F_t = 2^{16} + 1 \) | \( 2^{15} \) | 32 | 65536 | 64 |
| \( F_t = 2^{32} + 1 \) | \( 2^{31} \) | 64 | - | 128 |

In an FNT over \( F_t \), the quantity \( \hat{\delta} = \sqrt{2} \) represents the integer \( 2^{2^{2t} - 2} \left( 2^{2^{t-1}} - 1 \right) \).

It can be shown that the FNT possess the Linear Convolution Property (LCP). The FNT can therefore be employed to perform convolution or correlation via point-by-point multiplication in the transform-domain in the same manner as the FFT, thus:

\[ f_k = z_k \ast h_k = FNT^{-1} \left( FNT(z_k) \bullet FNT(h_k) \right) \quad (21) \]

Since the FNT is defined over a finite ring, convolution (or correlation) via (21) has the highly desirable property that no roundoff errors are introduced. Note, however, that the filter output, \( f_k \), must lie in the range \( \left[ -\left( F_t / 2 \right), \left( F_t / 2 \right) \right] \), which requires that, the relation:

\[ |k_n|_{\text{max}}, |h_k|_{\text{max}} \leq \sqrt{2^{2^{t-1} - 2} / 2^{2^{t-2}}} \quad (22) \]

Be satisfied. Otherwise, severe amplitude aliasing results. A similar restriction applies to the output when the NTT is used for correlation.

#### 3.2 Extended Fermat number transform (EFNT)

To achieve the goal of implementing the MDF-BRLS algorithm without multiplications, the solution is to extend the idea of the FNT. Indeed, in MDF-BRLS algorithm, the linear convolution between two sequences, each of length \( M \), must be calculated by finding \( 2K' \) FNT and \( K' \) IFNT each transform of length \( M/K' \).

\[ f_k^{k'} = z_k^{k'} \ast h_k^{k'} = EFNT^{-1} \left( EFNT \left( z_k^{k'} \right) \bullet EFNT \left( h_k^{k'} \right) \right) \quad (23) \]

\( k' = 0, 1, \ldots, K' - 1 \).

The extended FNT (EFNT) and its inverse are proposed by the following equations (24 and 25).

\[ F_k^{k'}(m) = EFNT \left( f_k^{k'}(m) \right) = \left( EFNT \left( z_k^{k'}(m) \right) \bullet EFNT \left( h_k^{k'}(m) \right) \right) = Z_k^{k'}(m) \bullet H_k^{k'}(m) \]
\[ f_k^m(n) = \left\langle \sum_{n=0}^{2^{m+1}-1} f_k(n) \left( 2^{2^m} \left( 2^{m+1} - 1 \right) \right)^{-n} \right\rangle_{F_k} \]  

To highlight, schematically, the efficiency of the Linear Convolution (LC) implementing by the Extended FNT, we present a block diagram depicted in Figure 1 for \( L = K' = 2 \) and \( M = 4 \). The parallel computing of the \( K' \) LC, without multiplications, decreases the execution time and the computational complexity of the MDF-BRLS algorithm.

\[ F_k = [F_k^0, F_k^1, \ldots, F_k^{K'-1}] \]

**Figure 1.** Block diagram of Linear Convolution using EFNT for \( L = K' = 2 \) and \( M = 4 \)

### 4. Simulation results

The performance of the MDF-BRLS algorithm implemented with FFT and Extended-FNT transforms is evaluated by computer simulation using Matlab. In these simulations, the MDF-BRLS algorithm is considered, in single-talk situation, with the parameters listed in Table 2.

**Table 2.** Simulation parameters.

| Parameter                          | Value       |
|------------------------------------|-------------|
| Number of new input samples        | \( N = 129 \) |
| Length of filter’s impulse response| \( L = 128 \) |
| The forgetting factor              | \( \lambda = 1 \) |
| Length of input sequence           | \( M = 256 \) |
| Number of sub-vectors              | \( K' = 32 \) |
| Length of sub-impulse response     | \( L' = 4 \) |
| FFT/EFNT length                    | \( M / K' = M' = 8 \) |
| Value of Fermat number             | \( F_3 = 2^8 + 1 \) |

The size of the impulse response is \( L = 128 \), that matches a delay of 16 ms used for a sampling rate of 8 kHz. In single-talk state, the acoustic echo cancellation system should reduce the echo to around 24 dB for delays lesser than 25 ms and to around 40 dB for delay beyond 25 ms.
The criteria used to evaluate the performance of MDF-BRLS algorithm implemented with FFT and EFNT are compared as follows:

- The filter weight error convergence $N_m$:

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

- The response of the echo track impulse
- The computational complexity

Based on the equation of the weight error convergence, the MDF-BRLS performance is simulated and presented in Figure 2.

![Figure 2. Adaptation normalized misalignment for MDF-BRLS algorithm implemented with FFT (solid line) and Extended-FNT (dashed line)](image)

The result obtained in figure just before presents a perfect construction of echo’s path of the MDF-BRLS algorithm (convergence is around $-78$ dB), using FFT and Extended-FNT transforms. This figure shows also that both transforms, FFT and Extended-FNT, do not reveal any significant difference in terms of convergence.

The impulse response of the echo path calculated by both transforms FFT and Extended-FNT is seen in Figure 3.

This result obtained in figure above, by using the MDF-BRLS algorithm were analyzed with both transforms, FNT and Extended FNT revealed that there was no difference in terms of construction of echo’s path (error difference is around $10^{-16}$ dB). Furthermore, the result obtained presents a perfect construction of echo’s path, which means that the residual echo is unheard at the output of the echo cancellation system using both methods of implementation.

Based on the obtained values, seen in Table III, the differences between both methods of implementation reside mainly in the computational complexity of the filter.

As well known, that the FFT and Extended-FNT are fundamentally having the same implementing, our EFNT-based block adaptive filtering has been applied with convolution steps based on the FFT procedure. One difference that distinguishes the Extended-FNT computation from other techniques, arises from the use of finite arithmetic modulo, the Fermat number $F_j = 2^k + 1$, that can be realized based on the conventional binary arithmetic.
The global computational effectiveness of the implementation is directly related with the computational complexity of the algorithm MDF-BRLS based on both transforms, FFT and EFNT. The size of FFT and the EFNT used and their numbers, determines the operations number as additions, bit shifts and multiplications. A $M'$-point EFNT can be shown to require $\frac{3}{2}M'\log_2(M')$ basic operations such as bit shifts and additions but without multiplication. Based on this hypothesis, Table 3 gives the total operation number per block for each linear convolution used by MDF-BRLS algorithm and implemented with the both transforms, FFT and EFNT.

**Table 3.** Basic operations and multiplications required for each linear convolution implemented with FFT and extended-FNT

| Transformation | Basic operations | Multiplications |
|----------------|------------------|-----------------|
| FFT            | $3.K'.M'\log_2(M')= 2304$ | $K'.M'\left(1 + \frac{3}{2}\log_2(M')\right)= 1408$ |
| Extended-FNT   | $\frac{9.K'}{2}.M'\log_2(M')= 3456$ | $K'.M'= 256$ |

Besides, this previous realization of the MDF-BRLS filter is computationally further competent because of the computational proficiency of the Extended Fermat number transform.

**5. Conclusion**

Throughout this paper, an Extended Fermat Number Transform (EFNT) to reduce enormously the computational complexity and in particular the multiplications have been presented. Moreover, an implementation of the MDF-BRLS algorithm using EFNT reduced enormously the number of multiplications of the implantation compared with an implementation of the same algorithm using FFT. The algorithm using EFNT finds a numerous application in several domains include, power electronics control [11], [12], radar system, echo cancelation in teleconference room and in headset. Our proposition can be more advanced to tackle the double talk situation, and other matters.

**References**
[1] Farhang-Boroujeny B 1998 *Adaptive Filters. Theory and Applications* (New York: Wiley & Sons)

[2] Bashirov M G, Khusnutdinova I G, Khusnutdinova L G and Usmanov D R 2016 Electromagnetic-Acoustic Method for Assessing the Technical Condition of Energy Equipment *Industrial energy* **12** 8–13

[3] Mohammed M S and Kim K S 2016 Preliminary Treatment of a Signal Based on The Compression of Noise for a Filter Using a Recursive Algorithm of The Least Squares in Ultrasonic Material Control *Defectoscopy* **6** 51–9

[4] Benesty J, Gansler T, Morgan D, Sondhi M and Gay S L 2001 *Advances in Network and Acoustic Echo Cancellation* (Berlin: Springer-Verlag)

[5] Alaeddine H H, Ghaith A, Al-Sahili M and Fadlallah M 2017 New Approach of Performance Analysis of RLS adaptive Filter Using a Block-Wise Processing *European Academic Research* **V** 1030–53

[6] Soo J S and Pang K K 1990 Multidelay block frequency domain adaptive filter *IEEE Transactions on Acoustics, Speech, and Signal Processing* **38** 373–76

[7] Buchner H, Benesty J, Gansler T and Kellermann W 2006 Robust Extended Multidelay Filter and Double-Talk Detector for Acoustic Echo Cancellation *IEEE Transactions on Audio, Speech and Language Processing* **14** 1633–44

[8] Alaeddine H H, Houssini M, Baghious E H and Burel G 2018 *IEEE Int. Conf. on Selected Topics in Mobile and Wireless Networking* (Morocco, Tangier) pp 1-6

[9] Julien G A 1991 Number theoretic techniques in digital signal processing *Advances in Electronics and Electron Physics* **80** 69–163

[10] Alaeddine H H, Bazzi O, Alaeddine A and Burel G 2012 Fast convolution using generalized sliding Fermat number transform with application to digital filtering *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E95.A** 6 1007–17

[11] Charafeddine K, Sangov K and Tsyruk S 2017 *IEEE-IT-DREPS Conf.* (Jordan, Amman) pp 153–8

[12] Falshina V A and Kulikov A L 2011 Algorithms for Simplified Digital Filtration of Electrical Signals of Industrial Frequency *Industrial Energy* **4** 39–46