

PAPER

Realization of Multi-Delay Filter Using Fermat Number Transforms

Hamzé Haidar ALAEDDINE^{†a)}, El Houssain BAGHIOUS[†], Guillaume MADRE[†], *Nonmembers*,
and Gilles BUREL[†], *Member*

SUMMARY This paper is about an efficient implementation of adaptive filtering for echo cancelers. The first objective of this paper is to propose a simplified method of the flexible block Multi-Delay Filter (MDF) algorithm in the time-domain. Then, we will derive a new method for the step-size adaptation coefficient. The second objective is about the realization of a Block Proportionate Normalized Least Mean Squares (BPNLMS++) with the simplified MDF (SMDF) implementation. Using the new step-size method and the smaller block dimension proposed by SMDF, we achieve a faster convergence of the adaptive process with a limited computational cost. Then, an efficient implementation of the new procedure (SMDF-BPNLMS++) block filtering is proposed using Fermat Number Transform, which can significantly reduce the computation complexity of filter implantation on Digital Signal Processor.

key words: adaptive filters, echo canceler, Fermat transforms

1. Introduction

The problem of echo cancellation is recurrent for all modern communications systems. The general solutions for reducing the additive echo noise are based on digital filtering process. Several types of adaptive algorithms exist, which give an efficient answer to these audio degradations [1].

To improve conversation quality, the most popular echo canceler (EC) uses a Least Mean Square (LMS) adaptive filter. It will therefore be desirable to implement fast-converging algorithms in future echo cancelers. In [1], faster converging algorithms called proportionate normalized least means squares (PNLMS++) are proposed. In order to gain computational advantages, a realization of this recent PNLMS++ adaptive filter with a blockwise processing is introduced in [2].

In this context, the size of the impulse response can reach several thousand coefficients. Therefore they are unsuitable and expensive for the implemented adaptive algorithm. To limit this problem, we utilise a more flexible adaptive block Multi-Delay Filter (MDF) [3], in order to deal with short impulse response and to reduce the execution time of the adaptive process.

This study proposes a simplified form of the block MDF algorithm (SMDF), in order to reduce the computation complexity. In spite of SMDF efficiency, it is not sufficient to reach a faster convergence of the adaptive process,

in the presence of large variations in the input power spectrum, without a better control of the step size. To achieve this goal, we derive a new method for the step-size adaptation coefficient control. This method consists in reducing the effect of large variations in the input power spectrum which improves the behavior of the SMDF algorithm and provides fast convergence.

In a second purpose, the SMDF algorithm has been implemented with the time-domain version of Block-PNLMS++ (BPNLMS++) algorithm. This new procedure SMDF-BPNLMS++ adaptive filter has a smaller block size and faster adaptation speed.

To design an efficient adaptive filtering into processor DSP in fixed-point arithmetic, we propose to realize the implementation of the new procedure, SMDF-BPNLMS++ algorithm, with Fermat Number Transforms (FNT), developed for fast error-free computation of finite digital convolutions [4]. These transforms present the following advantages compared to Fast Fourier Transform (FFT) [5].

- They require few or no multiplications
- They suppress the use of floating point complex number and allow error-free computation
- All calculations are executed on a finite ring of integers, which are interesting for implementation into DSP

Hence, the use of FNT will reduce the delay features by minimizing the computational complexity.

The rest of the paper is organized as follows. In Sect. 2, we will introduce the general diagram of the MDF algorithm and its proposed form. The new adaptation step-size definition is given in Sect. 3. Section 4 presents the new procedure, SMDF-BPNLMS++. In Sect. 5, we present the concept of Fermat Number Transform, which will be implemented to the adaptive digital filter and in the final part, numerical results of the new procedure realization using fast transform are given.

2. The MDF Adaptive Filter

The MDF algorithm is basically a block frequency-domain adaptive filtering procedure [3], which consists in segmenting the impulse response of length L in K' short successive segments of length L' . This is achieved by using smaller block size, updating the weight vectors more often, and reducing the total execution time of the adaptive process. Throughout the paper, uppercase symbols denote frequency-domain variables, lowercase symbols stand for time-domain

Manuscript received June 18, 2007.

Manuscript revised May 9, 2008.

[†]The authors are with Laboratoire d'Electronique et des Systèmes de Télécommunications - UMR CNRS 6165, 6, avenue Le Gorgeu - CS 93837 - 29238 BREST cedex 3, France.

a) E-mail: hamze.alaeddine@univ-brest.fr

DOI: 10.1093/ietfec/e91-a.9.2571

variables.

A. Principle

Let us consider the convolution between the input sequence $\{x\}$ of length N and the L -point FIR filter \hat{w}_k . Let L' be an integer number. Let us assume that L is a integer multiple of L' , $L = K'L'$, and divide the impulse response \hat{w}_k into K' segments of length L' and use short FFT of size P , where P is the smallest power of 2, greater or equal with $N + L' - 1$. We define then K' vectors $\hat{w}_k^{k'}$ as:

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

The k th segment $\hat{w}_k^{k'}$ of the vector \hat{w}_k will be given by the following expression:

$$\hat{w}_k^{k'} = [\hat{w}_k(k'L') \dots \hat{w}_k((k' + 1)L' - 1)]^T, \quad 0 \leq k' \leq K' - 1$$

The basic operation underlying block-convolution algorithm is the division of the input signal $\{x\}$ into overlapping sections. The size of the sections is set to $N + L' - 1$.

$$x_k^{k'} = [x_{kN-(k'+1)L'+1} \dots x_{kN-k'L'+N-1}]^T \quad (2)$$

We define $\tilde{X}_k^{k'}$ and $\tilde{W}_k^{k'}$ as the $(P \times 1)$ vector whose entries are respectively equal to the FFT coefficients of the k th block of the input section $x_k^{k'}$ and $\hat{w}_k^{k'}$, zeros padded to reach the required size of P .

$$\tilde{X}_k^{k'} = FFT\left(\begin{bmatrix} x_k^{k'} \\ 0_{1 \times (P-(N+L'-1))} \end{bmatrix}\right)^T \quad (3)$$

$$\tilde{W}_k^{k'} = FFT\left(\begin{bmatrix} \hat{w}_k^{k'} \\ 0_{1 \times (P-L')} \end{bmatrix}\right)^T \quad (4)$$

where FFT is the forward Fast Fourier Transform. From the $(P \times 1)$ frequency-domain output vector, \hat{Y}_k , as the following product:

$$\hat{Y}_k = [\tilde{X}_k^0 \dots \tilde{X}_k^{K'-1}] \bullet \begin{bmatrix} \tilde{W}_k^0 \\ \vdots \\ \tilde{W}_k^{K'-1} \end{bmatrix} = \sum_{k'=0}^{K'-1} \tilde{X}_k^{k'} \bullet \tilde{W}_k^{k'} \quad (5)$$

The inverse FFT (FFT^{-1}) of \hat{Y}_k , is a P -point sequence corresponding to the sum of the circular convolution of the input signal sections $\tilde{X}_k^{k'}$ by the corresponding segments of the impulse response $\tilde{W}_k^{k'}$, ($0 \leq k' \leq K' - 1$), is given by:

$$\tilde{y}_k = FFT^{-1}(\hat{Y}_k) = FFT^{-1}\left(\sum_{k'=0}^{K'-1} \tilde{X}_k^{k'} \bullet \tilde{W}_k^{k'}\right) \quad (6)$$

where the operator \bullet denotes the term by term multiplication.

The last N points correspond to the final filtered output samples \hat{y}_k while the first $(P - N)$ samples of \tilde{y}_k correspond to circular convolution in which time-aliasing has occurred.

The adaptive filter, fed by an input $\{x\}$, is recursively adapted so as to minimize a mean-squared difference between the actual output of the adaptive filter and a desired

signal $\{\tilde{d}\}$ [1].

The residual echo vector ϵ_k is defined as:

$$\epsilon_k = \tilde{d}_k - \hat{y}_k = [e_{kN} \dots e_{(k+1)N-1}]^T \quad (7)$$

The sequence of the path echo for the iteration k , is given by $\tilde{d}_k = [d_{kN} \dots d_{(k+1)N-1}]^T$.

In MDF the time-domain tap weights vector is divided into K' segments with the L' -dimensional of each segment:

At each block index k , the update can be written:

$$\hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \mu_k \bullet FFT^{-1}(\tilde{X}_k^{k'} \bullet E_k) \quad (8)$$

The definitions and roles of the various terms involved in this expression are given below:

$-E_k$ is the frequency-domain error signal vector, as:

$$E_k = FFT\left(\begin{bmatrix} \epsilon_{-k}^T \\ \vdots \\ 0_{1 \times (P-N)} \end{bmatrix}\right)^T \quad (9)$$

$-\mu_k$ is the new step-size parameter, which controls the adaptive behavior of the algorithm (see definition in Sect. 3).

B. The Simplified MDF (SMDF)

In this subsection, we extend the idea of the MDF adaptive filter further. To achieve the goal of minimizing the computation complexity, the solution is $N = L'$. To explain the benefit of this choice, we present an example in Fig. 1. This example shows an input signal $\{x\}$ of length $N + L - 1 = 15$, divided into $K' = 3$ segments, $x_k^{k'}$, of length $N + L' - 1 = 7$ with $N = L' = 4$ and $L = 12$.

From this example, we note for $N = L'$, that $x_k^{k'} = x_{k-1}^{k'-1}$, i.e., $\tilde{X}_k^{k'} = \tilde{X}_{k-1}^{k'-1}$, $1 \leq k' \leq K' - 1$. This means that at each iteration it is necessary only to evaluate the FFT transform for the first entry block \tilde{X}_k^0 , the other vectors $\tilde{X}_k^1, \dots, \tilde{X}_k^{K'-1}$ being calculated at the step $k - 1$. We have in this case, a significant computation saving of the FFT by a factor $(K' - 1)$, which reduces the computational complexity of the proposed algorithm (SMDF) compared to the basic algorithm MDF.

In this particular case, when $N = L'$, the residual echo

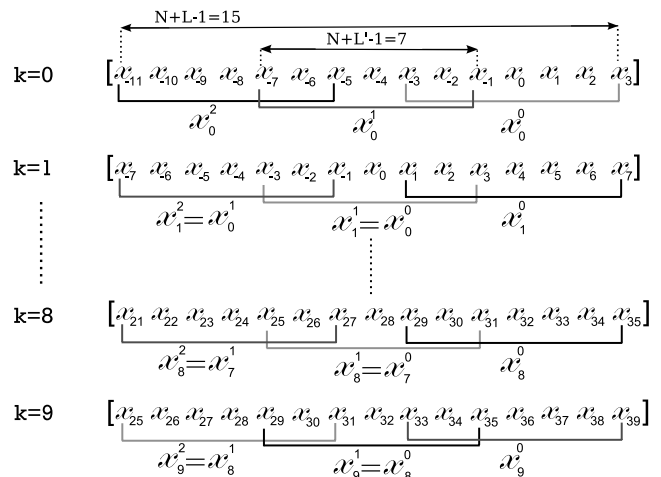


Fig. 1 Composition of vectors $\tilde{X}_k^{k'}$ for 10 blocks, $N = L' = 4$ and $L = 12$.

vector ϵ_k , defined by (6) and (7), becomes:

$$\epsilon_k = \tilde{d}_k - FFT^{-1} \left(\tilde{X}_k^0 \bullet \tilde{W}_k^0 + \sum_{k'=1}^{K'-1} \tilde{X}_{k-1}^{k'-1} \bullet \tilde{W}_k^{k'} \right) \quad (10)$$

The weights update equation of SMDF adaptive filter is given by:

$$\begin{cases} \hat{w}_{k+1}^0 = \hat{w}_k^0 + \mu_k \bullet FFT^{-1} (\tilde{X}_k^0 \bullet E_k) \\ \hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \mu_k \bullet FFT^{-1} (\tilde{X}_{k-1}^{k'-1} \bullet E_k), \\ 1 \leq k' \leq K' - 1 \end{cases} \quad (11)$$

The elements of each N -dimensional vector $\epsilon_k = \tilde{d}_k - \hat{y}_k$ are defined as:

$$\begin{cases} \epsilon_{kN+j} = \tilde{d}_{kN+j} - \sum_{i=0}^{L'-1} x_{kN+j-i}^0 \hat{w}_k^0(i), & k' = 0 \\ \epsilon_{kN+j} = \tilde{d}_{kN+j} - \sum_{i=0}^{L'-1} \left(\sum_{k'=1}^{K'-1} x_{(k-1)N-(k'-1)L'+j-i}^{k'-1} \hat{w}_k^{k'}(i) \right) \end{cases} \quad (12)$$

where $1 \leq k' \leq K' - 1$ and $j \in [0, N - 1]$.

Consider $\Phi_k = \tilde{X}_k^{k'} \bullet E_k$ written out for the j th weight:

$$\Phi_k(i) = \sum_{j=0}^{N-1} x_{kN-k'L'+j-i}^{k'} \epsilon_{kN+j}, \quad 0 \leq i \leq L' - 1$$

Substituting $n = kN - k'L' + j - i$, $\Phi_k(i)$ becomes

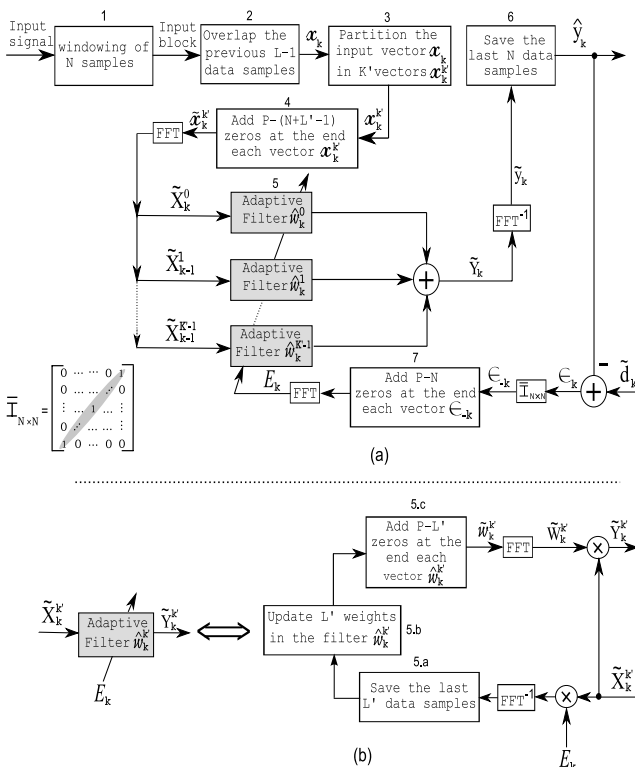


Fig. 2 A simplified MDF (SMDF) adaptive filter.

$$\Phi_k(i) = \sum_{n=kN-k'L'-i}^{(k+1)N-k'L'-i-1} x_n^{k'} \epsilon_{n+k'L'+i} = x_k^{k'}(i) * \epsilon_{-k}(i) \quad (13)$$

Moreover, Eq. (11) of SMDF adaptive filter has been redefined as a correlation computation. Thus, the elements of L' -dimensional weights vectors are given by:

$$\begin{cases} \hat{w}_{k+1}^0(i) = \hat{w}_k^0(i) + \mu_k(i) (x_k^0(i) * \epsilon_{-k}(i)) \\ \hat{w}_{k+1}^{k'}(i) = \hat{w}_k^{k'}(i) + \mu_k(i) (x_{k-1}^{k'-1}(i) * \epsilon_{-k}(i)), & k' > 0 \end{cases} \quad (14)$$

The block diagram of the proposed algorithm SMDF is depicted in Fig. 2 and clearly illustrates the efficient block structure.

The SMDF algorithm is also a very flexible solution for echo cancellation.

3. New Adaptation Step-Size Definition

To ensure the convergence of the adaptive filter at marginal computational cost, simple update rules have to be investigated. Defining a new control of the step-size in Eq. (8) seems to be a good choice. The step-size μ_k is given by [3]:

$$\mu_k = \text{diag}(K' \mu \Gamma_k) \quad (15)$$

where Γ_k is the transform domain normalization vector and μ is the block step size.

Indeed, Γ_k is an important parameter which controls the convergence of the adaptation process. It is given by:

$$\Gamma_k = \left[\frac{1}{Z_k(0)}, \frac{1}{Z_k(1)}, \dots, \frac{1}{Z_k(P-1)} \right]^T \quad (16)$$

where the r th spectral component $Z_k(r)$ is the input signal $\{x\}$ estimated power spectral density (psd):

$$Z_{k+1}(r) = \lambda Z_k(r) + (1 - \lambda) \sum_{k'=0}^{K'-1} |\tilde{X}_k^{k'}(r)|^2 \quad (17)$$

where $\tilde{X}_k^{k'}(r)$ is the r th component of the input vector FFT and λ is some constant smoothing coefficient chosen in the $[0, 1]$.

According to the particular conditions, $N = L'$, a new definition of μ_k is proposed here which can significantly reduce the computation complexity. In this case, $\forall k' > 0$, we have the input signal, $\tilde{X}_k^{k'}$, given by the latest FFT block transform $\tilde{X}_{k-1}^{k'-1}$. Then we consider the vectorial calculus of μ_k in the following form:

$$\mu_k = K' \mu \Gamma_K \quad (18)$$

where $\Gamma_K(r)$ is the new parameter proposed which controls the convergence of the adaptation process defined by:

$$\begin{aligned} \Gamma_K(r) &= \frac{1}{Z_k(r)} \\ Z_{k+1}(r) &= \lambda Z_k(r) + (1 - \lambda) |\tilde{X}_k^0(r)|^2 \end{aligned} \quad (19)$$

The power estimate and normalisation in (16)–(19) reduces the effect of large variations in the input power spectrum which improves the convergence speed of the simplified algorithm SMDF and consequently the proposed algorithm SMDF-BPNLMS++.

4. A Multi-Delay BPNLMS++ Algorithm

This part presents a new block adaptive filtering procedure in which the filter coefficients are divided into K' segments in accordance with a SMDF-BPNLMS++. This is achieved by using smaller block size and the new procedure of the step-size, reducing the total execution time of the BPNLMS++ adaptive filter and providing fast convergence of the modes of the adaptive process.

We will briefly present the adaptive filtering method issued from BPNLMS++ algorithm, in defined [2].

The weight update equation of BPNLMS++ algorithm, using fast convolution method based on FFT techniques, is given by:

$$\begin{aligned}\hat{w}_{k+1} &= \hat{w}_k + \mu \frac{G_k (\epsilon_k * \tilde{x}_{-k})}{G_k (\tilde{x}_k * \tilde{x}_{-k}) + \beta} \\ &= \hat{w}_k + \left(\frac{G_k}{G_k (\tilde{x}_k * \tilde{x}_{-k}) + \beta} \right) \cdot (\mu \cdot (\epsilon_k * \tilde{x}_{-k}))\end{aligned}\quad (20)$$

in which $\tilde{x}_k = [x_{kN-L+1} \dots x_{(k+1)N-1}]^T$ is $(N+L-1)$ -dimensional vector, the operator $*$ denotes the linear convolution and β is a regularization parameter which prevents division by zero. In this algorithm, for even-numbered time steps the matrix G_k is chosen to be the identity matrix:

$$G_k = I_L \quad (21)$$

while for odd-numbered steps it is chosen as follows:

$$G_k = \text{diag}[g_k(0), \dots, g_k(L-1)] \quad (22)$$

G_k is a diagonal matrix that adjusts the step-size of the individual taps of the filter:

$$g_k(n) = \frac{\gamma_k(n)}{\frac{1}{L} \sum_{m=0}^{L-1} \gamma_k(m)} \quad (23)$$

where $\gamma_k(n) = \max\{\rho v_k, |\hat{w}_k(n)|\}$, $n \in [0, L-1]$ and $v_k = \max\{\delta, |\hat{w}_k(0)|, \dots, |\hat{w}_k(L-1)|\}$. The terms δ and ρ are typically chosen equal to $\frac{\epsilon}{L}$ and 10^{-2} respectively.

The SMDF implementation of the BPNLMS++ algorithm supposes the expressions of the N samples from the error sequence by Eq. (10).

In proposed algorithm SMDF-BPNLMS++ the time-domain tap weight vector, \hat{w}_k , is divided into K' segments $\hat{w}_k^{k'}$ of length L' : $1 \leq k' \leq K' - 1$.

At each block index k , the weight update equation of SMDF-BPNLMS++ adaptive filter can be written:

$$\hat{w}_{k+1}^{k'} = \hat{w}_k^{k'} + \left(\frac{G_k}{\tilde{x}_k^T G_k \tilde{x}_k + \beta} \right) u_k^{k'} \quad (24)$$

β corresponds to the variance of the signal $\{x\}$ [6], and G_k is similarly obtained as the BPNLMS++ algorithm in this part. The vector $u_k^{k'}$ is calculated, at each iteration, using Eq. (11), by:

$$\begin{cases} u_k^0 = \text{Last } L' \text{ terms of } \left\{ \mu_k \bullet FFT^{-1} \left[\left(\tilde{X}_k^0 \bullet E_k \right) \right] \right\} \\ u_k^{k'} = \text{Last } L' \text{ terms of } \left\{ \mu_k \bullet FFT^{-1} \left[\left(\tilde{X}_k^{k'-1} \bullet E_k \right) \right] \right\}, \\ k' > 0 \end{cases} \quad (25)$$

μ_k is the new step-size defined in Sect. 3. The different elements of the vector $u_k^{k'}$ is given by:

$$\begin{aligned}u_k^0(i) &= \mu_k(i) \sum_{n=kN-1}^{(k+1)N-i-1} x_n^0 \epsilon_{n+i} = \mu_k(i) (x_k^0(i) * \epsilon_{-k}(i)) \\ u_k^{k'}(i) &= \mu_k(i) \sum_{n=kN-k'L'-1}^{(k+1)N-k'L'-i-1} x_n^0 \epsilon_{n+k'L'+i} = \mu_k(i) (x_{k-1}^{k'-1}(i) * \epsilon_{-k}(i)), \\ k' &> 0\end{aligned}\quad (26)$$

where $*$ indicates convolution and $i \in [0, L' - 1]$.

The weight update equation of L -point SMDF-BPNLMS++ adaptive filter, \hat{w}_k , is given by:

$$\begin{bmatrix} \hat{w}_k^0 \\ \vdots \\ \hat{w}_k^{K'-1} \end{bmatrix} = \begin{bmatrix} \hat{w}_{k+1}^0 \\ \vdots \\ \hat{w}_{k+1}^{K'-1} \end{bmatrix} + \left(\frac{G_k}{\tilde{x}_k^T G_k \tilde{x}_k + \beta} \right) \begin{bmatrix} u_k^{0T} \\ \vdots \\ u_k^{K'-1T} \end{bmatrix} \quad (27)$$

With reference to Eqs. (1) and (27), the L -point SMDF-BPNLMS++ adaptive filter, \hat{w}_k , can be written:

$$\hat{w}_{k+1} = \hat{w}_k + \left(\frac{G_k}{\tilde{x}_k^T G_k \tilde{x}_k + \beta} \right) \tilde{u}_k \quad (28)$$

where

$$\tilde{u}_k = [u_k^{0T} u_k^{1T} \dots u_k^{K'-1T}]^T \quad (29)$$

Analysis of convergence properties and computational complexity shows that the block SMDF-BPNLMS++ adaptive filter permits fast implementation while keeping good performance.

5. Fermat Number Transform

A realization of SMDF-BPNLMS++ adaptive filters incorporates fast convolution algorithms using the FFT and appropriate sectioning of data. Since the Fermat Number Transform (FNT) has several desirable properties mentioned in this part, a realization of SMDF-BPNLMS++ filters using the FNT instead of the FFT should be considered. The forward and inverse FNT's are defined as [4]:

$$X(k) = \left\langle \sum_{n=0}^{M-1} x(n) \alpha^{nk} \right\rangle_{F_t}, \quad k = 0, 1, \dots, M-1 \quad (30)$$

$$x(n) = \left\langle M^{-1} \sum_{k=0}^{M-1} X(k) \alpha^{-nk} \right\rangle_{F_t}, \quad n = 0, 1, \dots, M-1 \quad (31)$$

Table 1 Possible combinations of FNT parameters.

t	modulo F_t	M for $\alpha = 2$	M for $\alpha = 4$	M for $\alpha = \sqrt{2}$
2	$2^4 + 1$	8	4	16
3	$2^8 + 1$	16	8	32
4	$2^{16} + 1$	32	16	64
5	$2^{32} + 1$	64	32	128
6	$2^{64} + 1$	128	64	256

respectively, where M is the transform length, F_t is the t th Fermat number, α is a root of unity, and $\langle \cdot \rangle_{F_t}$ stands for residue reduction modulo F_t .

For an efficient implementation of FNT on processor, the choice of parameters is important. If possible, the values M and α are chosen as a power of 2 to allow replacement of multiplications by bit shifts. The particular modulo equal to a Fermat number, $F_t = 2^{2^t} + 1$ with $t \in \mathbb{N}$, offers numerous possibilities for length M of the transform. The values of M and α associated to a Fermat Number Transform are given by $M = 2^{t+1-i}$ and $\langle \alpha = 2^{2^i} \rangle_{F_t}$ with $0 \leq i < t$ [5] (see Table 1).

Some tests have shown that an FNT-based convolution reduces the computation time by a factor of 3 to 5 compared to the FFT implementation [4].

6. Numerical Results

Numerical simulations have been conducted to evaluate the performance of the proposed SMDF-BPNLMS++ algorithm implemented with FNT transform. In these tests, the SMDF-BPNLMS++ algorithm is compared to the BPNLMS++ algorithm into *MatLab* software.

The criteria used to evaluate the performances of both FNT implemented BPNLMS++ and SMDF-BPNLMS++ algorithms are compared, and are as follows:

- The filter weight error convergence N_m :

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

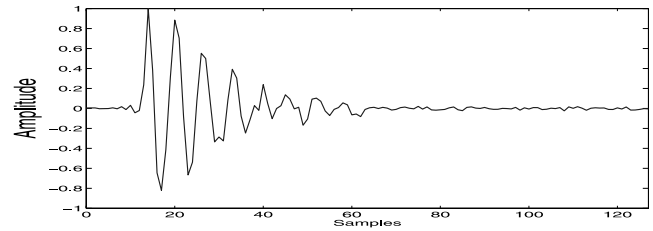
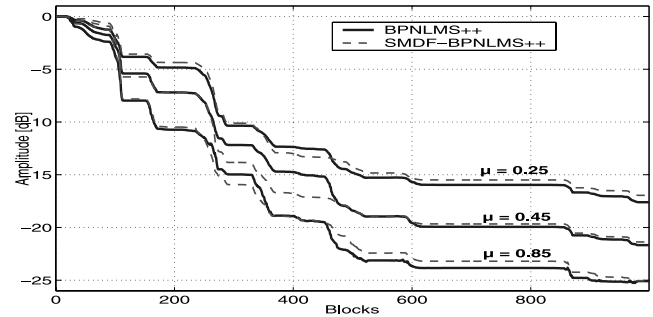
where w and \hat{w} are respectively the real and the estimated impulse response of the experimental echo channel.

- The Echo Return Loss Enhancement (ERLEC) of the compensator \hat{w}_k .

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

- The instantaneously Echo Return Powers.
- The computational complexity of both FNT implemented BPNLMS++ and SMDF-BPNLMS++ algorithms.

The BPNLMS++ and SMDF-BPNLMS++ adaptive filter algorithms are investigated in single-talk situation where empirical values for the parameters are chosen as $\beta = 10^2$, $\delta = \frac{1}{L}$, $\rho = 10^{-2}$, $\gamma = 0.9$. The dimension of the computed block in filter processing is taken with $N = L' = 32$ where $K' = 4$, and $L = 128$, which corresponds


Fig. 3 Impulse response of the echo path.

Fig. 4 Adaptation normalized misalignment for two algorithms: BPNLMS++ (solid line) and SMDF-BPNLMS++ (dashed line) with $\mu = 0.25, 0.45$, and 0.85 .

to a 16 ms impulse response for a sampling rate of 8 kHz. In practice, the length of the echo-path impulse response in the telephone network usually varies between about 4 ms and 64 ms and the sampling rate is usually 8 kHz.

The impulse response of the filter w used in the simulations is given in Fig. 3.

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 delays exceeding 25 ms [7].

Figure 4 represents the convergence of both algorithms (BPNLMS++ and SMDF-BPNLMS++) using $\mu = 0.25, 0.45$, and 0.85 . This figure shows that both algorithms do not reveal any significant difference in terms of convergence for each value of μ . It shows also that for $\mu = 0.85$, these algorithms present a faster convergence.

Then, we will evaluate the performance of the SMDF-BPNLMS++ algorithm using the new step size (see definition in Sect. 3) and compare it to the BPNLMS++ algorithm using $\mu = 0.85$.

In Fig. 5, we could show that the SMDF-BPNLMS++ represented with dashed plot, presents a faster adaptation convergence than BPNLMS++ algorithm given with solid plot.

In Fig. 6(a), the performance of the SMDF-BPNLMS++ is measured by the Echo Return Loss Enhancement of the compensator \hat{w}_k . We observe the fast adaptation obtained by the SMDF-BPNLMS++ (dashed curve) algorithm in contrast with the slower BPNLMS++ algorithm variant.

Figure 6(b), represents the instantaneously power of the echo signal and the instantaneously Echo Return Powers obtained with the classical BPNLMS++ and the SMDF-

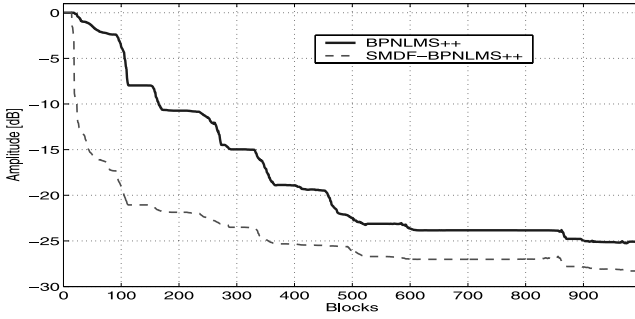


Fig. 5 Adaptation normalized misalignment for two algorithms: BPNLMS++ (solid) and SMDF-BPNLMS++ (dashed).

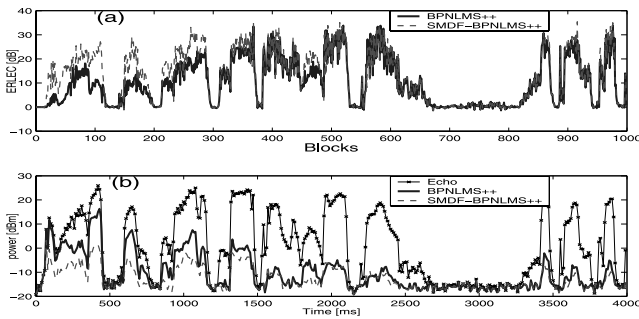


Fig. 6 (a) Echo Return Loss Enhancement of the compensator \hat{w}_k . (b) Echo Return Powers for two algorithms: BPNLMS++ (solid) and SMDF-BPNLMS++ (dashed).

BPNLMS++ algorithms. It can be seen that the last algorithm provides much larger echo attenuation than the classical BPNLMS++ algorithm.

It is noted that the SMDF-BPNLMS++ algorithm using smaller block size and the new method of step-size adaptation coefficient, is found to be faster than the BPNLMS++ algorithm.

The last criterion of comparison between the two algorithms, BPNLMS++ and SMDF-BPNLMS++, is that of the computational complexity.

As a realization of block adaptive filters incorporates fast convolution algorithms and appropriate sectioning of data, an implementation of SMDF-BPNLMS++ filters using the FNT, instead of the FFT, could be considered for real data [4], [5].

As the FFT and FNT realizations are basically equivalent, our FNT-based block adaptive filtering has been implemented with convolution procedures using the FFT technique. The only difference in the FNT computation comes from the use of finite arithmetic modulo, the Fermat number $F_4 = 2^{16} + 1$, which can be implemented using conventional binary arithmetic.

The overall computational efficiency of a realization is directly associated with the computational complexity of both algorithms (BPNLMS++ and SMDF-BPNLMS++) using the FNT. The number of operations (multiplications, additions, bit shifts) of both algorithms (BPNLMS++ and SMDF-BPNLMS++) is directly proportional to the num-

Table 2 Operations (basic operations and multiplications) required for both FNT implemented BPNLMS++ and SMDF-BPNLMS++ for one block.

algorithm	basic operations	multiplications
BPNLMS++	$(35L) \log_2(2L)$	$6L$
SMDF-BPNLMS++	$(10K' + 15)L' \log_2(2L') + (K' - 1)(2L' - 1)$	$4K'L'$

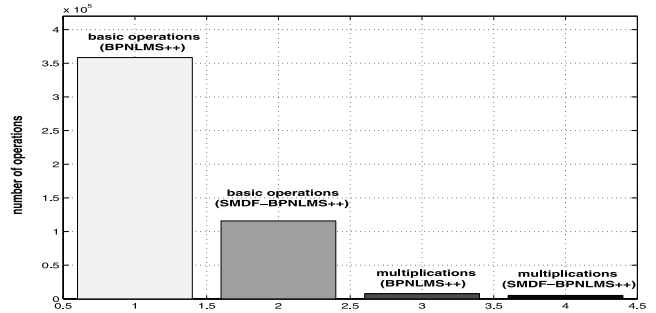


Fig. 7 Operations required for both FNT implemented BPNLMS++ and SMDF-BPNLMS++ algorithms (10 blocks).

ber and size of the FNT used. A $2L$ -point FNT can be shown to require $5L \log_2(2L)$ basic operations such as bit shifts and additions but no multiplication. With this assumption, the total operation number per block for the convolution of the different algorithms (BPNLMS++ and SMDF-BPNLMS++) is given by Table 2.

With reference to the previous table, the required operations (multiplications, additions, bit shifts) of 10 blocks for both FNT implemented BPNLMS++ and SMDF-BPNLMS++ algorithms are represented in Fig. 7.

Moreover, this last implementation of the SMDF-BPNLMS++ filter is computationally more efficient due to the computational efficiency of the Fermat Number Transform.

7. Conclusion

This paper proposed a simplified form of the flexible adaptive Multi-Delay Filter (MDF). In the presence of large variations in the input power spectrum, the MDF is not sufficient to reach the uniform convergence. To inhibit this problem, we derived a new method for the step-size adaptation coefficient.

Afterwards, we proposed a new procedure based on the BPNLMS++ algorithm with the simplified MDF (SMDF) implementation in order to provide fast convergence and to reduce the execution time of the adaptive process. Moreover, an implementation of the new procedure (SMDF-BPNLMS++) algorithm using FNT reduced the computational complexity of the implantation compared with an implementation of the BPNLMS++ algorithm using also FNT.

References

- [1] D.L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancelers," IEEE Trans. Speech Audio Process., vol.8,

no.5, pp.508–518, Sept. 2000.

- [2] E.H. Baghious, G. Madre, H. Alaeddine, and G. Burel, "Realization of adaptive blocks digital filters using Fermat Number Transform," International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC'04), Brest, France, July 2004.
- [3] J.S. Soo and K.K. Pang, "Multidelay block frequency domain adaptive filter," IEEE Trans. Acoust. Speech Signal Process., vol.38, no.2, pp.373–376, Feb. 1990.
- [4] R.C. Agarwal and C.S. Burrus, "Number theoretic transform to implement fast digital convolution," Proc. IEEE, vol.63, pp.550–560, April 1975.
- [5] G.A. Julien, "Number theoretic techniques in digital signal processing," J. Advances in Electronics and Electron Physics, vol.80, pp.69–163, 1991.
- [6] T.G. Gänslér, S.L. Gay, M.M. Sondhi, and J. Benesty, "Double-talk robust fast converging algorithms for network echo cancellation," IEEE Trans. Speech Audio Process., vol.8, no.6, pp.656–663, Nov. 2000.
- [7] A. Gilloire and M. Vetterli, "Performance evaluation of acoustic echo controls: Required values and measurement procedures," Annales des Télécommunications, vol.49, no.7-8, pp.368–372, 1994.



Gilles Burel was born in 1964. He received the M.Sc. degree from Ecole Supérieure d'Electricité, Gif Sur Yvette, France, in 1988 and the Ph.D. degree from University of Brest, France, 1991. From 1988 to 1997 he was a member of the technical staff of Thomson CSF, then Thomson Multimedia, Rennes, France, where he worked on image processing and pattern recognition applications as project manager. Since 1997, he has been Professor of Digital Communications, Image and Signal Processing at the University of Brest. He is Director of the Doctoral School SICMA and Associate Director of the Lab-STICC Laboratory (UMR CNRS 3192). He is the author of 19 patents, one book and more than one hundred scientific papers. His present research interests are in signal processing for digital communications with emphasis on MIMO systems and interception of communications.



Hamzé Haidar Alaeddine was born in Liban, in 1980. He received the B.Sc. degree in electronics from the Lebanese University, Liban, in 2002, and the M.Sc. degree from University of Brest, France, in 2003. In 2007, he received the Ph.D. degree from University of Brest, France. His research interests include Signal Processing for Telecommunications, adaptive filtering, echo cancellation, number theoretic transform.



El Houssaïn Baghious was born in 1955 in Morocco. He received the Ph.D. degree from University of Brest, France, in 1991. Since 1992, he has been an Associate Professor and a researcher at the Lab-STICC Laboratory (UMR CNRS 3192), University of Brest, France. His research interests are in the areas of Number Theoretic Transform and its applications to signal processing for digital communications.



Guillaume Madre received the M.Sc. degree in electrical engineering in 2000 and the Ph.D. degree in 2004, both from Brest University, France. In 2000, he joined the Electronics and telecommunications Systems Laboratory of Brest University. His research interests were signal processing, speech coding and Voice over IP. Since 2006, he works in a private industry research division to develop Telecoms softwares. He currently manages the implementation of VoIP gateway services using SIP protocol in an IMS network architecture.

in an IMS network architecture.