14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

Realization of Block Robust Adaptive Filters using Generalized Sliding Fermat Number Transform

H. Alaeddine, E.H. Baghious, G. Madre and G. Burel
Laboratoire d'Electronique et des Systèmes de Télécommunications - UMR CNRS 6165
6, avenue Le Gorgeu - CS 93837 - 29238 BREST cedex 3 - FRANCE
phone: +(33) 298100070, email: (hamze.alaeddine, baghious)@univ-brest.fr

***Abstract*— This paper is about an efficient implementation of adaptive filtering for echo cancelers. First, a realization of an improved Block Proportionate Normalized Least Mean Squares $(\mathbf{BPNLMS}++)$ using Generalized Sliding Fermat Number Transform $(\mathbf{GSFNT})$ is presented. Unfortunately, during the double-talk mode, the echo cancelers often diverge. We can cope with this problem by employing a double-talk detector formed by two Voice Activity Detectors $(\mathbf{VAD's})$. We propose a general system based on the Robust-Block-$\mathbf{PNLMS}++$ $(\mathbf{RBPNLMS}++)$ adaptive filter combined with a post-filter. The general system was implemented with $\mathbf{GSFNT}$ which can significantly reduce the computation complexity of the filter implantation on Digital Signal Processing $(\mathbf{DSP})$.**

I. Introduction

The problem of echo cancellation is recurrent for all modern communication 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 $(\mathbf{EC})$ uses a Least Mean Square $(\mathbf{LMS})$ adaptive filter. It will therefore be desirable to implement fast-converging algorithms in future echo cancelers. In [2], faster converging algorithms called proportionate normalized least means squares $(\mathbf{PNLMS}++)$ are proposed. In order to gain computational advantages, a realization of this recent $\mathbf{PNLMS}++$ adaptive filter with a blockwise processing is introduced in [3] [4]. The previous Block-$\mathbf{PNLMS}++$ $(\mathbf{BPNLMS}++)$ algorithm is proposed to realize the implementation with Fermat Number Transform $(\mathbf{FNT})$, developed for fast error-free computation of finite digital convolutions [5] [6]. These transforms present the following advantages compared to Fast Fourier Transform $\mathbf{FFT}$ [7]:

- 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 is interesting for implementation into $\mathbf{DSP}$

Hence, the use of $\mathbf{FNT}$ will reduce the delay features by minimizing the computational complexity.

An efficient state space method for implementing the fast $\mathbf{FFT}$ over rectangular windows is proposed for the cases when there is a large overlap among the consecutive input signals. This is called Generalized Sliding $\mathbf{FFT}$ $(\mathbf{GSFFT})$ [8].

Similar to the $\mathbf{GSFFT}$, the Generalized Sliding $\mathbf{FNT}$ $(\mathbf{GSFNT})$ is proposed with the purpose of reducing the complexity of $\mathbf{FNT}$ based convolvers and correlators and thus enlarging the application area for the $\mathbf{FNT}$. We propose the algorithm of the $\mathbf{GSFNT}$ similar to the $\mathbf{GSFFT}$ [9].

The $\mathbf{GSFNT}$ is then used to propose an efficient implementation of the $\mathbf{BPNLMS}++$ algorithm. The complexity of this method is compared with that of standard $\mathbf{FNT}$.

A high convergence rate of the $\mathbf{EC}$ is usually accompanied by a proportional divergence rate in the double-talk situation (i.e., near-end speech). The presence of this mode perturbs the $\mathbf{EC}$ adaptive filter and as a consequence, the echo will not be correctly canceled. To inhibit the divergence of the $\mathbf{EC}$ during double-talk, the standard procedure is to use a level-based double-talk detector formed by two Voice Activity Detector $(\mathbf{VAD's})$. Unfortunately, during the time required by the double-talk detector to detect the mode, the $\mathbf{EC}$ often diverges.

To compensate the problem of divergence, we propose a general system based on the conventional $\mathbf{EC}$ combined with a post-filter which attenuates the echo while keeping the near-end signal. During single-talk the post-filter attenuates the residual echo that still exists at the output of the $\mathbf{EC}$.

Although the double-talk detector works well, detection errors do occur, and these result in large amounts of divergence of the adapted filter coefficients. The solution to this problem is to adapt the coefficients by the new robust-$\mathbf{BPNLMS}++$ $(\mathbf{RBPNLMS}++)$ algorithm. We propose to realize the implementation of the previous general system with $\mathbf{GSFNT}$. Hence, the use of $\mathbf{GSFNT}$ will reduce the delay features, by minimizing the computational complexity.

The paper is organized as follows. In Section $\mathbf{II}$, we will introduce the new block diagram of $\mathbf{EC}$ and the double-talk detector. The $\mathbf{RBPNLMS}++$ algorithm and the post-filter are presented in Section $\mathbf{III}$. In Section $\mathbf{IV}$, the $\mathbf{GSFNT}$ is employed and in the final part, numerical results of the general combined system realization using fast transforms are given.

II. Echo Cancellation System and the Double-Talk Detection

The classic method used for echo canceller $(\mathbf{EC})$, is based on adaptive identification of the echo path impulse response $\mathbf{w}$ [1].

The conventional $\mathbf{EC}$ filters the far-end speech $\{\mathbf{x}\}$ by an echo path image $\{\hat{\mathbf{w}}_\mathbf{k}(\mathbf{n})\}_{\mathbf{n=0}}^{\mathbf{L-1}}$ to obtain an echo estimation $\{\hat{\mathbf{y}}\}$ in single-talk situation:

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

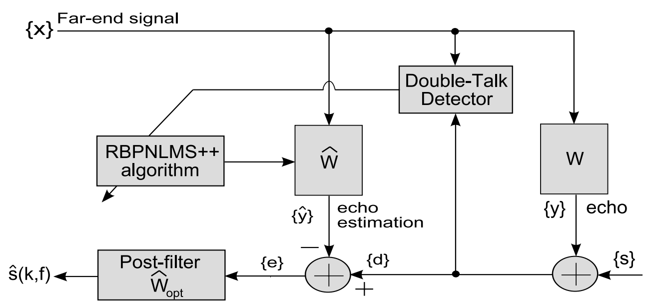

Fig. 1. *Combined acoustic echo cancellation system and double − talk detector*

In double-talk situation, the observation signal $\{\mathbf{d_k}\}$ can be written as follows :

$$\{\mathbf{d_k}\} = \{\mathbf{s_k}\} + \{\mathbf{y_k}\}$$

where $\{\mathbf{s_k}\}$ and $\{\mathbf{y_k}\}$ is the near-end speech and the acoustic echo respectively. Here $\mathbf{k}$ denotes the sample iteration.

Given this context, the $\mathbf{EC}$ often diverges and to compensate we consider a new scheme presented in Figure 1.

The double-talk detector is used in order to process the situations of single-talk or double-talk differently. In fact, the microphone signal can be composed of :

• The far-end speech $\{\mathbf{x}\}$ only (absence of the near-end speech $\{\mathbf{s}\}$). In such case, the echo is present and must be eliminated, the $\mathbf{RBPNLMS}++$ adaptive filter must be updated. The post-filter algorithm is supposed to attenuate the residual echo that still exists at the output of the $\mathbf{EC}$.

• Both the far-end speech and the near-end speech. In such case, the adaptation is stopped and the post-filter is applied.

• The near-end speech only (absence of the far-end speech). In such case, the echo is absent. There is no need for filtering.

It's proven in [11] that the $\mathbf{RPNLMS}++$ adaptive filter is an excellent candidate for both acoustic and network echo cancellation. In the next section, we derive this algorithm using block structure.

III. ROBUST ALGORITHMS AND POST-FILTER

A. Robust PNLMS++ Using Block Structure

This subsection presents a new block adaptive filtering procedure in which the filter coefficients are adjusted once per each output block in accordance with a $\mathbf{RPNLMS}++$ algorithm. Note that the previous robust-$\mathbf{LMS}$-type adaptation, which adjusts parameters once per each sample, is in fact a special case with a unitary block length.

The filtering equation 1 is then written in matrix form as [3]:

$$\mathbf{\hat{Y}_k} = \begin{bmatrix} x_{kN} & \dots & x_{kN-L+1} \\ \vdots & \dots & x_{kN-L+2} \\ x_{(k+1)N-1} & \dots & x_{(k+1)N-L} \end{bmatrix} \begin{bmatrix} \hat{w}_k(0) \\ \dots \\ \hat{w}_k(L-1) \end{bmatrix}$$

$$\mathbf{\hat{Y}_k} = \begin{bmatrix} \hat{y}_{kN} & \dots & \hat{y}_{(k+1)N-1} \end{bmatrix}^{\mathbf{T}} = \chi_{\mathbf{k}}\mathbf{\hat{w}_k} \quad (2)$$

where $\chi_{\mathbf{k}}$ is a $(\mathbf{N}\times\mathbf{L})$-Toeplitz matrix and $\mathbf{\hat{w}_k}$ is an estimated echo path vector.

Note that, in the block structure, the term $\mathbf{k}$ is no longer the sample iteration but becomes the block index.

The block-$\mathbf{RNLMS}$ algorithm uses a fast convolution method based on $\mathbf{FFT}$ techniques, in computing the filter outputs and updating its weights, given respectively by equations 3 and 4 .

$$\mathbf{\hat{Y}_k} = \mathbf{\hat{w}_k} * \mathbf{X_k} \quad (3)$$

$$\mathbf{\hat{w}_{k+1}} = \mathbf{\hat{w}_k} + \mu \frac{\mathbf{sign}\left(\epsilon_{\mathbf{k}}\right) * \left(\mathbf{X_{-k}}\right)}{\mathbf{X_k} * \mathbf{X_{-k}} + \beta} \psi\left(\frac{|\epsilon_{\mathbf{k}}|}{\mathbf{s_k}}\right) \mathbf{s_k} \quad (4)$$

in which $\mathbf{X_k} = \begin{bmatrix} x_{kN-L+1} & \dots & x_{(k+1)N-1} \end{bmatrix}^{\mathbf{T}}$ is $(\mathbf{N}+\mathbf{L}-\mathbf{1})$-dimensional vector and $\psi(.)$ is a limiter given by equation 5, [11]. The operator $*$ denoting the linear convolution, β is a regularization parameter which prevents division by zero and $\mu \in [\mathbf{0}, \mathbf{2}]$ is the adaptation step. Typically, β corresponds to the variance of the signal $\{\mathbf{x}\}$.

$$\psi\left(\frac{|\epsilon_{\mathbf{k}}|}{\mathbf{s_k}}\right) = \mathbf{min}\left\{\frac{|\epsilon_{\mathbf{k}}|}{\mathbf{s_k}}, \mathbf{k_0}\right\} \quad (5)$$

where $\mathbf{k_0} = \mathbf{1.1}$ and the estimate of the scale factor, $\mathbf{s_k}$, can be calculated as follows :

$$\mathbf{s_k} = \lambda \mathbf{s_{k-1}} + \frac{\mathbf{1}-\lambda}{\mathbf{a}} \mathbf{s_{k-1}} \psi\left(\frac{|\epsilon_{\mathbf{k}}|}{\mathbf{s_{k-1}}}\right) \quad (6)$$

where $\lambda \in [\mathbf{0.99}, \mathbf{1}]$, and $\mathbf{a} \in [\mathbf{0.60}, \mathbf{0.74}]$. The initial value of the scale factor can be chosen as $\mathbf{s_{-1}} = \sigma_{\mathbf{x}}$, where $\sigma_{\mathbf{x}}$ is set approximately to the average speech level in voice telephone networks.

The residual echo vector $\epsilon_{\mathbf{k}}$ is defined as :

$$\epsilon_{\mathbf{k}} = \mathbf{D_k} - \mathbf{\hat{Y}_k} = \begin{bmatrix} e_{kN} & \dots & e_{(k+1)N-1} \end{bmatrix} \quad (7)$$

The sequence of the path echo for the iteration $\mathbf{k}$, is given by $\mathbf{D_k} = \begin{bmatrix} d_{kN} & \dots & d_{(k+1)N-1} \end{bmatrix}^{\mathbf{T}}$.

The vector weight updated equation of $\mathbf{RBPNLMS}$ algorithm is given by :

$$\mathbf{\hat{w}_{k+1}} = \mathbf{\hat{w}_k} + \mu \frac{\mathbf{sign}\left(\epsilon_{\mathbf{k}}\right) * \left(\mathbf{G_k X_{-k}}\right)}{\mathbf{X_k} * \left(\mathbf{G_k X_{-k}}\right) + \beta} \psi\left(\frac{|\epsilon_{\mathbf{k}}|}{\mathbf{s_k}}\right) \mathbf{s_k} \quad (8)$$

Here, $\mathbf{G_k} = \mathbf{diag}\begin{bmatrix} g_k(0) & \dots & g_k(L-1) \end{bmatrix}$ is a diagonal matrix with the elements of $\mathbf{G_k}$ calculated as follows:

$$\mathbf{g_k(n)} = \frac{\gamma_{\mathbf{k}}(\mathbf{n})}{\frac{\mathbf{1}}{\mathbf{L}}\mathbf{\Sigma_{m=0}^{L-1}}\gamma_{\mathbf{k}}(\mathbf{m})} \quad (9)$$

where $\gamma_{\mathbf{k}}(\mathbf{n}) = \mathbf{max}\left\{\ \rho\nu_k, \quad |\hat{w}_k(n)|\ \right\}$, $\mathbf{0} \le \mathbf{n} \le \mathbf{L}-\mathbf{1}$ and $\nu_{\mathbf{k}} = \mathbf{max}\left\{\ \delta, \quad |\hat{w}_k(0)|, \quad \dots, \quad |\hat{w}_k(L-1)|\ \right\}$. ρ and δ are typically chosen equal to $\frac{\mathbf{5}}{\mathbf{L}}$ and $\mathbf{10^{-2}}$respectively.

Then the alternation between $\mathbf{RBNLMS}$ (if $\mathbf{k}$ is even) and $\mathbf{RBPNLMS}$ (if $\mathbf{k}$ is odd) yields the $\mathbf{RBPNLMS}++$.

B. Post-Filter using noise reduction techniques

This approach is based on a general concept of disturbance reduction, the echo being one of these disturbances. Therefore a large number of methods similar to noise reduction can be used to compute the post-filter. Our approach is based on a method of spectral subtraction used in noise reduction techniques [10]. The post-filter transfer function can be expressed as :

$$\mathbf{\hat{W}_{opt}}\left(\mathbf{k},\mathbf{f}\right) = \frac{\mathbf{SER_{prio}}\left(\mathbf{k},\mathbf{f}\right)}{\mathbf{1}+\mathbf{SER_{prio}}\left(\mathbf{k},\mathbf{f}\right)} \quad (10)$$

where $\mathbf{SER_{prio}}\left(\mathbf{k},\mathbf{f}\right)$ stands for the a priori Signal to Echo Ratio:

$$\mathbf{SER_{prio}}(\mathbf{k},\mathbf{f}) = \eta \frac{\left|\mathbf{\hat{W}_{opt}}\left(\mathbf{k}-\mathbf{1},\mathbf{f}\right)\mathbf{E}\left(\mathbf{k}-\mathbf{1},\mathbf{f}\right)\right|^{\mathbf{2}}}{\hat{\gamma}_{\epsilon}\left(\mathbf{f}\right)} + \left(\mathbf{1}-\eta\right)\mathbf{SER_{post}}\left(\mathbf{k},\mathbf{f}\right) \quad (11)$$

where $\mathbf{0} < \eta < \mathbf{1}$, and $\hat{\gamma}_{\epsilon}$ is an estimation of the power spectral density of the noise. The $\mathbf{SER_{post}}\left(\mathbf{k},\mathbf{f}\right)$ is the a Posteriori Signal to Echo Ratio, is calculated by:

$$\mathbf{SER_{post}}\left(\mathbf{k},\mathbf{f}\right) = \frac{\left|\mathbf{E}\left(\mathbf{k},\mathbf{f}\right)\right|^{\mathbf{2}}}{\hat{\gamma}_{\epsilon}\left(\mathbf{f}\right)} - \mathbf{1} \quad (12)$$

The estimate $\mathbf{\hat{S}}\left(\mathbf{k},\mathbf{f}\right)$ in the frequency domain of the block $\mathbf{k}$ for the frequency $\mathbf{f}$ is given by:

$$\mathbf{\hat{S}}\left(\mathbf{k},\mathbf{f}\right) = \mathbf{\hat{W}_{opt}}\left(\mathbf{k},\mathbf{f}\right)\mathbf{E}\left(\mathbf{k},\mathbf{f}\right) \quad (13)$$

where $\mathbf{E}$, is the Short Time Fourier Transform $(\mathbf{STFT})$ of $\epsilon_{\mathbf{k}}$.

IV. GENERALIZED SLIDING FERMAT NUMBER TRANSFORM (GSFNT)

A. Principle of the FNT

Discrete transforms based on the $\mathbf{FNT}$ concept have been developed for efficient and error-free computation of finite convolutions [5]. An $\mathbf{FNT}$ of a discrete time signal $\mathbf{x}$ and its inverse are given respectively by:

$$\mathbf{X(j)} = \left\langle \mathbf{\Sigma_{n=0}^{N-1}x(n)}\alpha^{\mathbf{nj}} \right\rangle_{\mathbf{F_t}} \quad (14)$$

$$\mathbf{x(n)} = \left\langle \mathbf{N^{-1}\Sigma_{j=0}^{N-1}X(j)}\alpha^{\mathbf{-nj}} \right\rangle_{\mathbf{F_t}} \quad (15)$$

where the operator $\langle\cdot\rangle_{\mathbf{F_t}}$ denoting the modulo $\mathbf{F_t}$, $\mathbf{N}$ is the transform length, $\mathbf{F_t} = \mathbf{2^{2^t}+1}$ is the particular modulo equal to a Fermat number, with $\mathbf{t} \in \mathbb{N}$, and the basis α can be equal to a power of $\mathbf{2}$, thus allowing the replacement of multiplications by bit shifts. When the parameters of an $\mathbf{FNT}$ are chosen judiciously, an $\mathbf{FNT}$ defined over Galois Field $(\mathbf{GF}(\mathbf{F_t}))$ has several desirable properties in carrying out convolutions and correlations in comparison to the $\mathbf{FFT}$. For the parameters given in Table I, by $\mathbf{N} = \mathbf{2^{t+1-i}}$ and $\alpha = \mathbf{2^{2^i}}$ with $\mathbf{i} < \mathbf{t}$.

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

TABLE I

Possible combinations of FNT parameters

Some tests have shown that an $\mathbf{FNT}$-based convolution reduces the computation time by a factor of $\mathbf{3}$ to $\mathbf{5}$ compared to the $\mathbf{FFT}$ implementation [6].

B. principle of the GSFNT.

In a similar way to the $\mathbf{GSFFT}$, a $\mathbf{GSFNT}$ is proposed in order to reduce the input-output delay of finite ring convolvers and correlators. It refers to the computation of the $\mathbf{FNT}$ as the sequence slides over a time-limited rectangular window one sample at a time [9]. The main advantage is the reduction of complexity. Here, we consider a more general case of the $\mathbf{SFNT}$ when the data window hops $\mathbf{P}$ samples at a time. We call this the $\mathbf{GSFNT}$. This is best explained by considering an example. For this purpose, we may consider the $\mathbf{N}$-point $\mathbf{FNT}$ structure of Figure 2 $(\mathbf{N} = \mathbf{16})$. Next, we may assume that two new data samples are collected $(\mathbf{P} = \mathbf{2})$, and the $\mathbf{FNT}$ has to be performed on these two samples plus the last $\mathbf{14}$ samples of the previous inputs to the $\mathbf{FNT}$ structure.

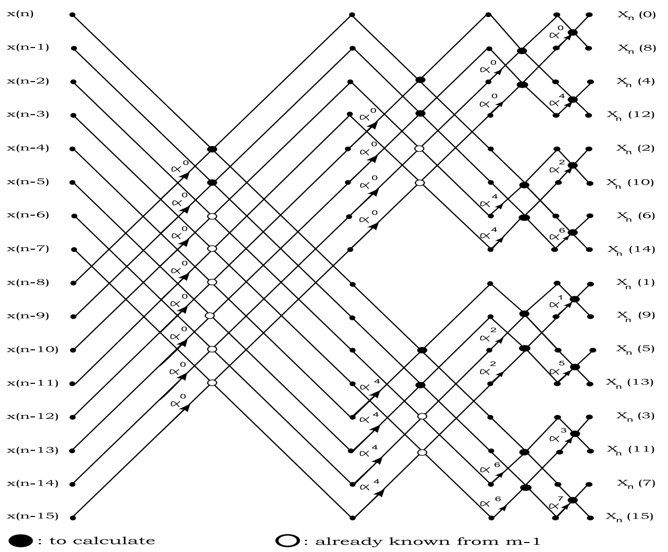

Fig. 2. *Generalized Sliding FNT of $N = 16$ and $P = 2$*

The computation complexity of performing the $\mathbf{GSFNT}$ after every $\mathbf{P}$ new data samples, in terms of the number of butterflies to be calculated, is:

$$\mathbf{C_{GSFNT}} = \frac{\mathbf{N}}{\mathbf{2}}(\mathbf{\log_2 P + 2}) - \mathbf{P} \tag{16}$$

Minimum input-output delay is obtained when $\mathbf{P = 1}$, the $\mathbf{GSFNT}$ reduces to the $\mathbf{SFNT}$, whereas the case $\mathbf{P = N}$ corresponds to the ordinary $\mathbf{FNT}$ ($\mathbf{C_{GSFNT}}$ is seen to vary from $\mathbf{N-1}$ for $\mathbf{P = 1}$ to $\frac{\mathbf{N}}{\mathbf{2}}\mathbf{\log_2 N}$ for $\mathbf{P = N}$).

C. Algorithm of the GSFNT

We propose the algorithm of the $\mathbf{GSFNT}$ similar to the $\mathbf{GSFFT}$ [9], based on computing the value of the variables in the $\mathbf{FNT}$ structure that are not computed in previous iterations. We define the $\mathbf{GSFNT}$ input vector as:

$$\mathbf{X_m} = \begin{bmatrix} x_n & \dots & x_{n-N+1} \end{bmatrix}^{\mathbf{T}} = \begin{bmatrix} Y_m^T & \dots & Y_{m-M+1}^T \end{bmatrix} \tag{17}$$

with $\mathbf{Y_m}$ is the block of the new samples of iteration $\mathbf{m}$:

$$\mathbf{Y_m} = \begin{bmatrix} x_n & x_{n-1} & \dots & x_{n-2^l+1} \end{bmatrix}^{\mathbf{T}} \tag{18}$$

with $\mathbf{P} = \mathbf{2^l}$, $\mathbf{M} = \mathbf{2^q}$ and $\mathbf{N} = \mathbf{P}.\mathbf{M}$. The $\mathbf{GSFNT}$ is based on the fact that in $\mathbf{FNT}$ a large part of the calculations is available from the previous iterations, i.e., we calculate the elements of $\mathbf{Y_m}$. The state vector is defined as:

$$\mathbf{U_m} = \begin{bmatrix} U_{0,m}^T & U_{1,m}^T & \dots & U_{(l+q),m}^T \end{bmatrix} \tag{19}$$

with $\mathbf{U_{0,m}} = \mathbf{Y_m}$, and $\mathbf{FNT}(\mathbf{X_m}) = \mathbf{U_{(l+q),m}}$. The state vector can be expressed in iterative form, when $\mathbf{0 \le i \le q-1}$:

$$\mathbf{U_{i+1,m}} =$$

$$\left\langle (\mathbf{R_i} \otimes \mathbf{I_{2^l}}) . \left(\mathbf{I_{2^{l+i}}} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} U_{i,m} \\ V_{i,l} U_{i,m-2^{q-i-1}} \end{bmatrix} \right) \right\rangle_{\mathbf{F_t}} \tag{20}$$

$\mathbf{R_i} = \begin{bmatrix} e_1 & e_{1+2^i} & e_{2+2^i} & \dots & e_{2^{i+1}} \end{bmatrix}$ is a $\mathbf{2^{i+1}}$ by $\mathbf{2^{i+1}}$ permutation matrix, $\mathbf{e_r}$ is $\mathbf{r}$'th column of $\mathbf{2^{i+1}}$ by $\mathbf{2^{i+1}}$ identity matrix and the operator $\otimes$ denoting the Kroneker product.

The last element of the state vector is given by:

$$\mathbf{V_{i,l} U_{i,m-2^{q-i-1}}} = (\mathbf{V_i} \otimes \mathbf{I_{2^l}}) \mathbf{U_{i,m-2^{q-i-1}}} =$$

$$\left(\mathbf{diag} \begin{pmatrix} 1 & \alpha^{\sigma(1)} & \dots & \alpha^{\sigma(2^i-1)} \end{pmatrix} \otimes \mathbf{I_{2^l}} \right) \mathbf{U_{i,m-2^{q-i-1}}} \tag{21}$$

with the basis α equal to a power of $\mathbf{2}$. $\sigma(\mathbf{r})$ is the bit reverse of $\mathbf{r}$, where $\mathbf{r} = \mathbf{0}, \mathbf{1}, \dots, \mathbf{2^i - 1}$.

for $\mathbf{q \le i < l + q}$

$$\mathbf{U_{i,m}} = \begin{bmatrix} U_{i,1,m} & U_{i,2,m} & \dots & U_{i,N,m} \end{bmatrix}^{\mathbf{T}} \tag{22}$$

$$\begin{bmatrix} U_{i+1,t,m} \\ U_{i+1,t+c,m} \end{bmatrix} = \left\langle \begin{bmatrix} 1 & V_{i,t} \\ 1 & -V_{i,t} \end{bmatrix} \begin{bmatrix} U_{i,t,m} \\ U_{i,t+c,m} \end{bmatrix} \right\rangle_{\mathbf{F_t}} \tag{23}$$

in which $\mathbf{V_{i,t}} = \alpha^{\sigma_\mathbf{i}(\mathbf{t})}$, $\mathbf{c} = \mathbf{2^{l+q-i-1}}$ and $\sigma_\mathbf{i}(\mathbf{t})$ is the bit reverse of $\langle(\mathbf{t-1})\rangle_{\mathbf{N/2^i}}$ such as $\mathbf{0} \le \langle\mathbf{t}\rangle_{\mathbf{N/2^i}} < \mathbf{N/2^{i+1}}$ and $\mathbf{0 \le t < N}$.

The inverse $\mathbf{FNT}$ using the sliding technique is obtained in a similar manner to the inverse $\mathbf{GSFFT}$ [8].

The total complexity of the $\mathbf{GSFNT}$ in terms of number of butterflies is given in equation 16. The complexity of the $\mathbf{GSFNT}$ butterfly processors on the other hand is dominated by the adders since all coefficients multiplications are powers of α.

V. Numerical results

A full numerical simulations have been conducted to evaluate the performances of the general combined system. In this section the simulations are organized as follows.

- We propose to realize the implementation of the previous $\mathbf{BPNLMS++}$ algorithm with $\mathbf{GSFNT}$.
- We will compare the robust versus the non-robust algorithms (convergence and divergence when double-talk occurs)
- We propose to realize the implementation of the general combined system with $\mathbf{GSFNT}$ to reduce the complexity compared to Fermat Number Transforms $\mathbf{FNT}$.

A. Implementation of GSFNT-based BPNLMS++

The $\mathbf{BPNLMS++}$ algorithm proposed in [4], presents a faster adaptation convergence than $\mathbf{BNLMS}$ or $\mathbf{BPNLMS}$ algorithms. Since the $\mathbf{GSFNT}$ has several desirable properties mentioned earlier, a realization of $\mathbf{BPNLMS++}$ filters using the $\mathbf{GSFNT}$ compared to the $\mathbf{FNT}$ is also considered. An implementation of $\mathbf{BPNLMS++}$ filters using the $\mathbf{GSFNT}$ reduces the complexity of a filter using a block structure.

The $\mathbf{BPNLMS++}$ adaptative filter algorithm, with blockwise processing, is investigated in single-talk situation where empirical values for the parameters are chosen as $\beta = \mathbf{100}$, $\rho = \frac{\mathbf{1}}{\mathbf{L}}$, $\delta = \mathbf{10^{-2}}$, and the step size $\mu = \mathbf{0.8}$. The dimension of the computed block in filter processing is taken $\mathbf{N = 64}$, $\mathbf{L = 193}$ and $\mathbf{P = 64}$. In Figure 3-a, The far-end signal $\{\mathbf{x}\}$ is $\mathbf{16}$ bit $\mathbf{PCM}$ coded at a $\mathbf{8kHz}$ sampling rate and the performance of the adaptation for echo cancellation is measured by means of the Echo Return Loss Enhancement of the compensator $\mathbf{\hat{w}_k}$ $(\mathbf{ERLEC})$ (Figure 3-b).

$$\mathbf{ERLEC}\,(\mathbf{k}) = \mathbf{10log_{10}}\left(\frac{\Sigma_{\mathbf{n=(k-1)N+1}}^{\mathbf{kN}}\,(\mathbf{y}\,(\mathbf{n}))^{\mathbf{2}}}{\Sigma_{\mathbf{n=(k-1)N+1}}^{\mathbf{kN}}\,(\mathbf{y}\,(\mathbf{n}) - \hat{\mathbf{y}}\,(\mathbf{n}))^{\mathbf{2}}}\right) \tag{24}$$

where $\mathbf{k}$ is the index of block.

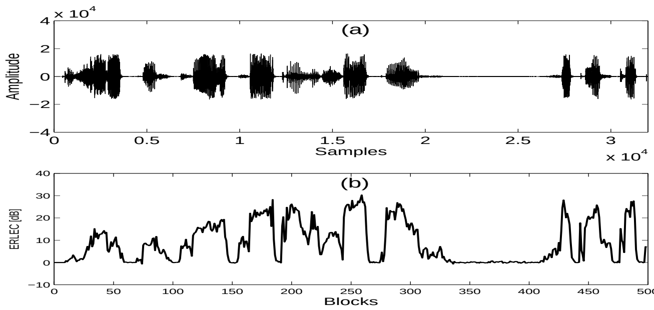

Fig. 3. *Performance for BPNLMS++ algorithm. Far − end (a) and ERLEC (b)*

In Figure 3-b, we present the attenuation of the echo brought by algorithm $\mathbf{BPNLMS++}$. In this Figure the attenuation of the echo is about $\mathbf{25\ dB}$.

In Figure 4, the performance of the $\mathbf{BPNLMS++}$ algorithm is mesured by means of the filter weight error convergence $\mathbf{N_m}$:

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

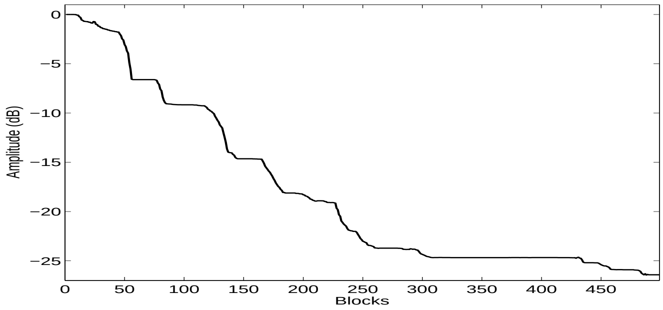

Fig. 4. *Adaptation normalized misalignment*

The overall computational efficiency of a realization is directly associated to the computational complexity of both transforms ($\mathbf{FNT}$ and $\mathbf{GSFNT}$). In Figure 5, the operation number, required for the convolution of the $\mathbf{BPNLMS++}$ filter, corresponds to $\mathbf{GSFNT}$ and $\mathbf{FNT}$-based realizations.

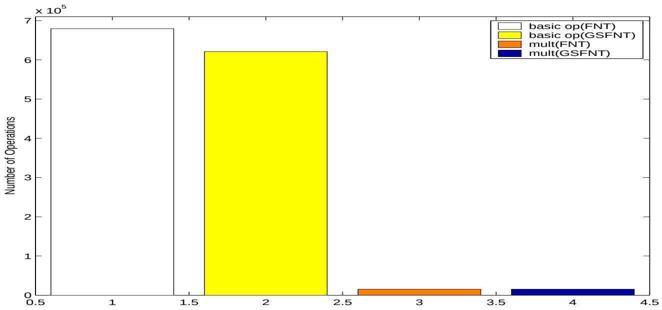

Fig. 5. *Operations required for BPNLMS++ filtering for* 10 *blocks*

This later implementation of the $\mathbf{BPNLMS++}$ filter is computationally more efficient due to $\mathbf{GSFNT}$ efficiency.

B. Comparisons between robust and non-robust algorithms

We compare the robust and the non-robust algorithms ($\mathbf{BNLMS}$, $\mathbf{BPNLMS}$, and $\mathbf{BPNLMS++}$) using the $\mathbf{VAD}$ presented in [12].

The parameters chosen for the following simulations are as follows:

- Parameters for the non-robust $\mathbf{BPNLMS++}$ algorithm are given in subsection V.A.
- Parameters for the robust $\mathbf{BPNLMS++}$ algorithm are, $(\lambda, \mathbf{k_0}) = (\mathbf{0.995}, \mathbf{1.1})$, $\mathbf{a = 0.6}$ and $\mathbf{s_{-1} = 1000}$.

In Figure 6 the misalignment of the robust and non-robust algorithms ($\mathbf{BNLMS}$, $\mathbf{BPNLMS}$, $\mathbf{BPNLMS++}$) is measured by means of the filter weight error convergence $\mathbf{N_m}$ given in equation 25.

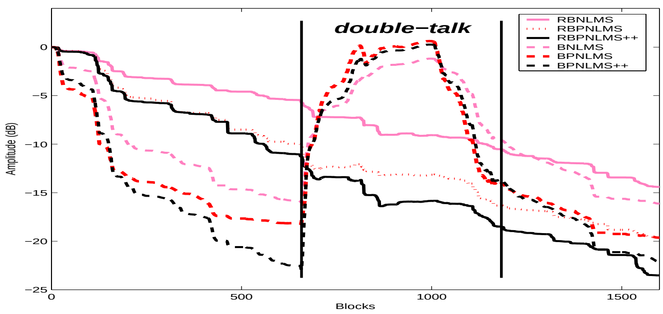

Fig. 6. *Adaptation normalized misalignment*

Initial convergence rate of $\mathbf{BNLMS}$, $\mathbf{BPNLMS}$, and $\mathbf{BPNLMS++}$ are clearly superior to that of $\mathbf{RBNLMS}$,

$\mathbf{RBPNLMS}$ and $\mathbf{RBPNLMS++}$. While the non-robust algorithms (with detector double-talk) diverge, the robust algorithms are much less affected and never perform worse than $-\mathbf{10}$ $\mathbf{dB}$ misalignment during double-talk. We can note also that the $\mathbf{RBPNLMS++}$ presents the faster adaptation convergence.

C. Implementation of GSFNT-based combined system

We insert a new block given in Figure 1. Numerical simulations have been conducted to evaluate the performance of the $\mathbf{GSFNT}$-based general combined system. There are three cases to study (Figure 7):

The parameters chosen for the following simulations are given in subsection V.B.

Figure 7-a represents the far-end signal, Figure 7-b represents the near-end signal and Figure 7-c represents the signal restored.

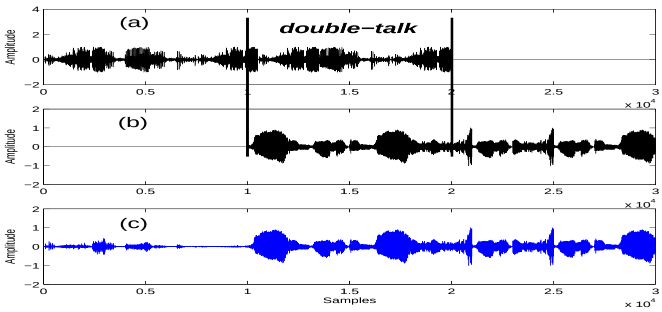

Fig. 7. $a)\ Far-end\ speech, b)\ Near-end\ speech\ and\ c)\ Near-end\ speech\ estimation\ \{\hat{s}\}$

In Figure 8-a, the performance of the general combined system is measured by means of the Echo Return Loss Enhancement ($\mathbf{ERLE}$) (case-single-talk) and Figure 8-b represents the $\mathbf{SNR}$ obtained by the general combined system (case-double-talk) :

$$\mathbf{SNR}\left(\mathbf{k}\right)=\mathbf{10log_{10}}\left(\frac{\mathbf{\Sigma_{n=(k-1)N+1}^{kN}}\left(\mathbf{s}\left(\mathbf{n}\right)\right)^{\mathbf{2}}}{\mathbf{\Sigma_{n=(k-1)N+1}^{kN}}\left(\mathbf{s}\left(\mathbf{n}\right)-\hat{\mathbf{s}}\left(\mathbf{n}\right)\right)^{\mathbf{2}}}\right) \tag{26}$$

where $\mathbf{s}(\mathbf{n})$ and $\hat{\mathbf{s}}(\mathbf{n})$ are the near-end speech and the reconstructed speech respectively.

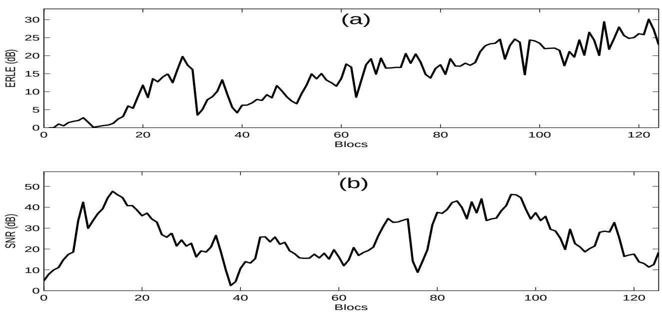

Fig. 8. $Echo\ Return\ Loss\ Enhancement\ (ERLE)\ and\ SNR\ between\ \{s\}\ and\ \{\hat{s}\}\ signals$

The computation gain being more significant for the filtering process, as the computational complexity of various operations needed in implementing $\mathbf{FNT}$ or $\mathbf{GSFNT}$-based general combined system is about the same when the convolutions are excluded, the overall computational efficiency of a realization is directly associated to the computational complexity of both transforms.

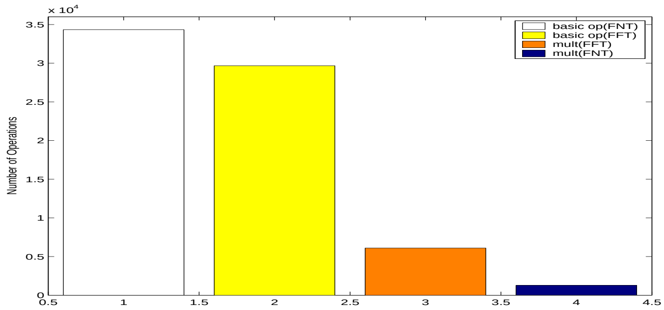

Fig. 9. $Operations\ required\ for\ the\ post-filter\ for\ 10\ blocks$

The operation number required for the convolutions of the $\mathbf{RBPNLMS++}$ is given in Figure 5. In the calculation of the post-filter there is no overlap between the consecutive input signals. The complexity of both different transforms ($\mathbf{FNT}$ and $\mathbf{GSFNT}$) are identical. In Figure 9 is presented the number of operations required for the $\mathbf{FFT}$ and $\mathbf{FNT}$-based post-filter implementation.

VI. CONCLUSIONS

In this paper, we consider the adaptation filter algorithm $\mathbf{BPNLMS++}$ and its implementation $\mathbf{GSFNT}$. A high convergence rate is usually accompanied by a high divergence rate in presence of double-talk. To avoid this problem of divergence, a general combined system is proposed, $\mathbf{RBPNLMS++}$ and the post-filter. The procedure is to use a system based on the double-talk detector formed by two Voice Activity Detectors $(\mathbf{VAD's})$. We realize the implementation of the general combined system with $\mathbf{GSFNT}$ which reduced the computational complexity of the implantation on Digital Signal Processors.

REFERENCES

[1] J. Benesty, T. Gansler, D.R. Morgan, M.M. Sondhi, S.L. Gay, “Advances in Network and Acoustic Echo Cancellation”, Springer-Verlag, 2001.

[2] D.L. Duttweiler, “Proportionate normalized least means square adaptation in echo cancelers”, IEEE Trans. Speech Audio Processing, vol. 8, pp. 508-518, September 2000.

[3] G. A. Clark, S.K. Mitra, and S.R. Parker, “Block implementation of adaptive digital filters”, IEEE Trans. on Acoust., Speech and Signal Processing, vol. ASSP-29, pp. 744-752, June 1981.

[4] E. H. Baghious, G. Madre, H. Alaeddine, 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, *accepted*.

[5] R.C. Agarwal, C.S. Burrus, “Number theoretic transform to implement fast digital convolution”, Proc. IEEE, vol. 63, pp.550-560, April 1975.

[6] R.C. Agarwal, C.S. Burrus, “Fast convolution using Fermat number transform with application to digital filtering”, IEEE trans. Acoust., Speech and Signal Proces., ASSP-22, N^o2, 1974.

[7] G.A. Julien, “Number Theoretic Techniques in Digital Signal Processing”, Academic Press, vol. 80, Chap. 2,pp. 69-163, 1991.

[8] B. Farhang-Boroujeny and S. Gazor, “Generalized sliding FFT and its application to implementation of block LMS adaptive filters,” IEEE Trans. Signal Processing, vol. 42. pp. 532-538, Mar, 1994.

[9] S. Gazor and B. Farhang-Boroujeny, “A state space approach for efficient implementation of block LMS adaptive filters,” Proc. Singapore int. Conf. Commun. Syst. ICCS/ISITA’92, (Singapore), Nov, 1992, pp. 808-812.

[10] Turbin V., Gilloire A., Scalart P., “Comparison of three post-filtering algorithms for residual acoustic echo reduction”, Processings of ICASSP’97, Vol. 1, pp. 307-310, Munich, Germany, April 1997.

[11] T. Gänsler, S.L. Gay, M.M. Sondhi, and J. Benesty, “Double-talk robust fast converging algorithms for network echo cancellation”, IEEE Transactions on Speech and Audio Processing, 8, 656-663, November 2000.

[12] ITU-T Recommendation. G.729, “Coding of Speech at 8 kbits/s using Conjugate Algebraic Code-Excited Linear Predictions (CS-ACELP)”, June 1995.