Realization of Block Adaptive Filters using Fermat Number Transforms

E.H. Baghious, G. Madre, H. Alaeddine 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
baghious@univ-brest.fr

Abstract - This paper is about an efficient implementation of adaptive filtering for echo cancelers. First, a realization of an improved Proportionate Normalized Least Mean Squares (PNLMS_{++}) adaptive filter using block structure is presented. Then, an efficient implementation of the block filtering process is proposed using Number Theoretic Transforms (NTT) which can significantly reduce the computation complexity of filter implantation on Digital Signal Processor (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 a conversation quality, the most popular echo canceler uses a Least Mean Square (LMS) adaptive filter. However, as the recent advances in telecommunication technology increase the voice delay and the echo cancellation problem, the development of more robust and fast converging Echo Cancelers (EC) remains an important issue. Hence, an improved adaptive algorithm, based on the standard Normalized-LMS (NLMS) and the original Proportionate-NLMS (PNLMS), has been proposed [2]. In order to gain computational advantage, a realization of this recent Proportionate Normalized Least Mean Squares (PNLMS_{++}) adaptive filter, with a blockwise processing [3], is introduced.

To design an efficient adaptive filtering into processor DSP in fixed-point arithmetic, we propose to realize the implementation of the previous Block-PNLMS_{++} (BPNLMS_{++}) algorithm with Number Theoretic Transforms (NTT), developed for fast error-free computation of finite digital convolutions [4][5]. These transforms present the following advantages compared to Fast Fourier Transform (FFT) [6]:

- 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 DSP

Hence, the use of Number Theoretic Transform will reduce the delay features by minimizing the computational complexity. The special case of Fermat Number Transforms (FNT), with arithmetic carried out modulo Fermat numbers, is particularly appropriate for digital computation. Its application to different functions, as filtering or correlation process, can provide real benefits for low computational complexity.

The rest of the paper is organized as follows. In section II, we will introduce the general diagram of echo cancellation by a filtering process and the formulation of the PNLMS_{++} adaptive algorithm. In a third part, a block processing of the previous procedure is given. Section IV presents the concept of Number Theoretic Transform and details, more particularly, the Fermat Number Transform, which will be implemented to the adaptive digital filter. In the final part, numerical results of the efficient adaptive filter realization using fast transforms are given.

II. Echo Cancellation Adaptive Filter

In standard Echo cancelers (EC), the Finite Impulse Response (FIR) filters, associated with Least Mean Squares (LMS) adaptive algorithm, are the most implemented systems. Recently, the Proportionate Normalized Least-Mean Squares (PNLMS_{++}) algorithm has been developed for efficient EC.

A. Echo Canceler Deployment

The conventional filtering model of an echo cancellation is given in Figure I-1. Here, an FIR filter estimates at best the echo signal $\{y\}$, by adapting its weights. The EC filters the far-end speech signal $\{x\}$ by an echo path image $\{\widehat{w}_k(n)\}_{n=0}^{N-1}$ to obtain an echo estimation $\{\widehat{y}\}$.

$$\widehat{y}_k = \sum_{n=0}^{N-1} \widehat{w}_k(n)\, x_{k-n} = \widehat{\mathbf{w}}_k^T \mathbf{x}_k \qquad (1)$$

where N denotes the coefficient order of the adaptive filter and k the sample iteration. Note that the bold terms, in the equation (1) *et seq.*, are vectors representation.

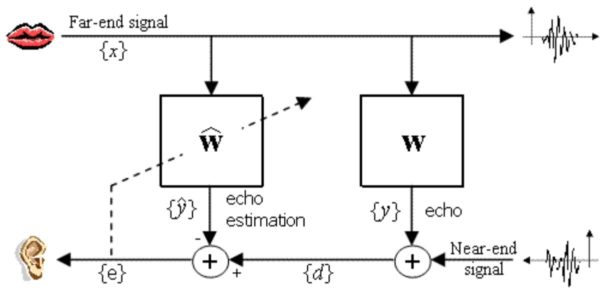

Figure I-1. Echo cancellation filter deployment

Some kind of LMS algorithms has been developed for echo cancellation as the standard stable Normalized-LMS (NLMS), which is tradionally used in echo canceler implementation and therefore serves as reference algorithm.

The weigth update equation of NLMS algorithm is given by :

$$\widehat{\mathbf{w}}_{k+1} = \widehat{\mathbf{w}}_k + \mu \frac{\mathbf{x}_k \left(d_k - \widehat{y}_k \right)}{\mathbf{x}_k^T \mathbf{x}_k + \beta} = \widehat{\mathbf{w}}_k + \mu \frac{\mathbf{x}_k e_k}{\mathbf{x}_k^T \mathbf{x}_k + \beta} \quad (2)$$

where the adaptation step $\mu \in [0, 2]$, β is a regularization factor and $\{d\}$ the near-end signal plus the echo $\{y\}$. Typically, μ is about 0.1 and β corresponds to the variance of the signal $\{x\}$.

Recently the Proportionate-NLMS (PNLMS) algorithm has been developed for use in network echo cancelers [7]. This efficient adaptation achieves significantly faster convergence than the conventional NLMS algorithm:

$$\widehat{\mathbf{w}}_{k+1} = \widehat{\mathbf{w}}_k + \mu \frac{G_k \mathbf{x}_k e_k}{\mathbf{x}_k^T G_k \mathbf{x}_k + \beta} \quad (3)$$

Here, $G_k = diag \left[g_k\left(0\right), ..., g_k\left(N\text{-}1\right) \right]$ is a diagonal matrix that adjusts the step-size of the individual taps of the filter :

$$g_k\left(n\right) = \frac{\gamma_k(n)}{\frac{1}{N} \sum_{m=0}^{N-1} \gamma_k(m)} \quad (4)$$

where $\gamma_k(n) = \max \left\{ \rho \nu_k, \left| \widehat{w}_k\left(n\right) \right| \right\}$, $n \in [0, N\text{-}1]$ and $\nu_k = \max \left\{ \delta, \left| \widehat{w}_k\left(0\right) \right|, ..., \left| \widehat{w}_k\left(N\text{-}1\right) \right| \right\}$. The terms ρ and δ, which prevent worst updating, are typically chosen equal to $\frac{5}{N}$ and 10^{-2} respectively.

B. Proportionate-NLMS++ Adaptation

More recently a variant adaptation, called PNLMS_{++}, has been proposed in [2]. This improved proportionate adaptive algorithm for an efficient filtering process solves the divergence problem, when the impulse response is sparse or dispersive, by alternating the update process each sample period between NLMS and PNLMS algorithms (figure I-2). For odd steps the matrix G_k is calculed as above while for even steps it is chosen to be the identity matrix $G_k = I_{N-1}$, which results in an NLMS iteration.

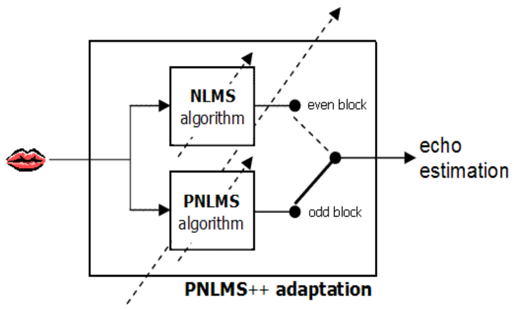

Figure I-2. Efficient proportionate adaptation

The adaptive PNLMS_{++} filter, which is less sensitive to a sparse impulse response of echo paths, gives an efficiently answer to the EC problem.

III. Blocks Adaptive Digital Filters

In this part, a PNLMS_{++} adaptation, using block structure, is defined [3]. Block adaptive filtering involves the computation of a block filter output from a block of input signal samples. The filter coefficients are adjusted once per each output block. Note that the previous LMS-type adaptation, which adjust parameters once per each signal sample, is in fact a special case with a unitary block length.

The filtering equation (1) is then written in matrix form as :

$$\widehat{\mathbf{y}}_k = \begin{bmatrix} x_{kN} & \cdots & x_{(k\text{-}1)N+1} \\ \cdots & \cdots & \cdots \\ x_{kN+L\text{-}1} & \cdots & x_{(k\text{-}1)N+L} \end{bmatrix} \begin{bmatrix} \widehat{w}_k\left(0\right) \\ \cdots \\ \widehat{w}_k\left(N\text{-}1\right) \end{bmatrix}$$

$$\widehat{\mathbf{y}}_k = \begin{bmatrix} \widehat{y}_{kN} & \cdots & \widehat{y}_{kN+L\text{-}1} \end{bmatrix}^T = X_k \widehat{\mathbf{w}}_k \quad (5)$$

where X_k is a (NxL)-Toeplitz matrix. Note that the term k is no longer the sample iteration but becomes the block index.

As the block-PNLMS_{++} algorithm can be written with convolution computation from equations (2) (3) and (5), the block algorithm can improve filter implementation, using fast convolution method based on FFT techniques, in computing the filter outputs and in updating its weights, given respectively in equations (6) (7) and (8).

$$\widehat{\mathbf{y}}_k = \widehat{\mathbf{w}}_k * \mathbf{x}_k \quad (6)$$

$$\widehat{\mathbf{w}}_{k+1} = \widehat{\mathbf{w}}_k + \mu \frac{\mathbf{e}_k * \mathbf{x}_{-k}}{\mathbf{x}_k * \mathbf{x}_{-k} + \beta} \text{ if } k \text{ is even} \quad (7)$$

$$\widehat{\mathbf{w}}_{k+1} = \widehat{\mathbf{w}}_k + \mu \frac{\mathbf{e}_k * \left(G_k \mathbf{x}_{-k} \right)}{\mathbf{x}_k * \left(G_k \mathbf{x}_{-k} \right) + \beta} \text{ if } k \text{ is odd} \quad (8)$$

where $\widehat{\mathbf{w}}_k^T = \left[\widehat{w}_k\left(0\right) ... \widehat{w}_k\left(N\text{-}1\right) \right]$ and $\mathbf{x}_k^T = \left[x_{(k\text{-}1)N+1} ... x_{kN+L\text{-}1} \right]$ are respectively N and (N+L-1)-dimensional vectors. Here, an empirical value will be chosen for the parameter β. As the operator $*$ denotes the linear convolution, the elements of each L-dimensional vector $\mathbf{e}_k = \mathbf{d}_k - \widehat{\mathbf{y}}_k$ are defined as :

$$e_{kN+l} = d_{kN+l} - \widehat{y}_{kN+l} = d_{kN+l} - \sum_{n=0}^{N-1} \widehat{w}_k\left(n\right) x_{kN+l-n} \quad (9)$$

with $l \in [0, L]$. Moreover, the denominators of both $\widehat{\mathbf{w}}_{k+1}$ have been redefined as a correlation computation. Thus, the elements of N-dimensional weights vectors are given by :

$$\widehat{w}_{k+1}\left(n\right) = \widehat{w}_k\left(n\right) + \mu \frac{\sum_{l=0}^{L-1} e_{kN+l} x_{kN+l-n}}{\beta + \sum_{l=0}^{L-1} x_{kN+l} x_{kN+l-n}} \quad (10)$$

$$\widehat{w}_{k+1}\left(n\right) = \widehat{w}_k\left(n\right) + \mu \frac{\sum_{l=0}^{L-1} e_{kN+l} g_k\left(n\right) x_{kN+l-n}}{\beta + \sum_{l=0}^{L-1} x_{kN+l} g_k\left(n\right) x_{kN+l-n}} \quad (11)$$

where $n \in [0, N]$ and the elements g_k of G_k are similarly obtained as the PNLMS_{++} algorithm in the previous part.

Analyses of convergence properties and computational complexity show that the block adaptive filter permits fast implementations while maintening good performances.

IV. Number Theoretic Transform

To compute the expressions of equations (6) (7) and (8), a block filter realization can incorporate fast convolution algorithm using the Fast Fourier Transform (FFT). Then, to develop an adaptive filtering process in fixed point arithmetic with low computational complexity, we propose to replace all FFT by an under-utilized Number Theoretic Transform (NTT).

A. Definitions

Discrete transforms based on the number theoretic concept have been developed for efficient and error-free computation of finite convolutions [5]. These transforms, called Number Theoretic Transform, present the same form as a Discrete Fourier Transform but are defined over finite rings [6]. All arithmetics must be carried out modulo M, which may be equal to a prime number or to a multiple of primes, since an NTT is defined over the Galois Field $GF(M)$. An NTT of a discrete time signal x and its inverse are given respectively by :

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\alpha^{nk} \right\rangle_M \quad (12)$$

$$x(n) = \left\langle N^{-1} \sum_{k=0}^{N-1} X(k)\alpha^{-nk} \right\rangle_M \quad (13)$$

where $n, k = 0, ..., N-1$. The DFT N^{th} root of the unit in $\mathbb{C}$, $e^{j\frac{2\Pi}{N}}$, is replaced by the N^{th} root of the unit over $GF(M)$ represented by the generating term α, which satisfies the equality $\left.\alpha^N = 1\right\rangle_M$ where N is the length of the tranform and $\langle . \rangle_M$ denotes the modulo M operation.

Note that an NTT has similar properties to the DFT such as the periodicity, symmetry or shift properties. Moreover, an NTT admits the Cyclic Convolution Property [8] [9] :

$$U * V = T^{-1}\{T(U) \bullet T(V)\} \quad (14)$$

where U and V represent both sequences to be convolved, T and T^{-1} are the forward and inverse NTT respectively. The operator $\bullet$ denotes the term by term multiplication.

B. Fermat Number Transform

The particular modulo equal to a Fermat number, $F_t = 2^{2^t} + 1$ with $t \in \mathbb{N}$, involves the highly composite transform lengths N and the basis α can be equal to a power of 2, hence allowing the replacement of multiplications by bit shifts. When the parameters of a Fermat Number Transform (FNT) are chosen judiciously, an FNT defined over $GF(F_t)$ has several desirable properties in carrying out convolutions and correlations in comparison to the FFT.

For the parameters given in Table I, by $N = 2^{t+1-i}$ and $\alpha = 2^{2^i}$ with $i < t$, multiplications in the FNT can be realized with shifts and additions whereas complex multiplications are required in the FFT. Also, an FNT computation needs only on the order of $N \log_2 N$ basic operations such as bit shifts and additions but no multiplication, while an FFT requires about $N \log_2 N$ multiplications.

Table I : *Possible Combinations of FNT Parameters*

t	modulo F_t	possible N values for $\alpha = 2$	$\alpha = 4$	$\alpha = \sqrt{2}$	$\alpha = \sqrt[4]{2}$
2	2^4+1	8	4	16	
3	2^8+1	16	8	32	64
4	$2^{16}+1$	32	16	64	128
5	$2^{32}+1$	64	32	128	256

Moreover, as a fast FNT-type computational structure similar to the Fast Fourier Transform exists, available FFT VLSI hardware structure for real-time implementation of the FNT may be adopted. 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].

V. Numerical Results

Numerical simulations have been conducted to evaluate the performances of the proposed FNT-based BPNLMS++ adaptive filtering. The computational complexities of the Fermat Number Transform realization and the fixed-point Fast Fourier Transform implementation are compared and the different adaptations convergence is studied through MatLab simulations.

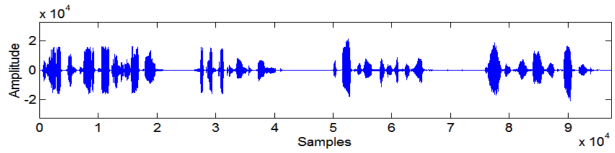

Figure V-1. Far-end speech signal

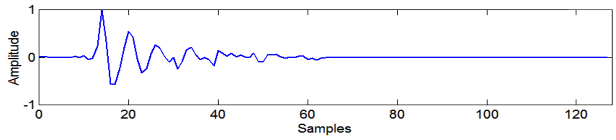

Figure V-2. Impulse response of the echo path

Throughout these experiments, input and output signals are assumed to be real, the far-end signal $\{x\}$ is 16 bit PCM coded at a 8kHz sampling rate (figure V-1). The impulse response of the filter $\mathbf{W}$ used in the simulations is given in figure V-2.

A. Performance Comparisons

The NLMS, PNLMS and PNLMS++ adaptative filter algorithms, with blockwise processing, are investigated in single talk situation where empirical values for the parameters are chosen as $\beta = 10^2$, $\delta = \frac{1}{N}$, $\rho = 10^{-2}$ and the step size $\mu = 0.8$. The dimension of the computed block in filter processing is taken with $N = L = 64$. In the following Figure V-3, the performances of three adaptations for echo cancellation are measured by means of the filter weight error convergence N_m :

$$N_m = 10\log_{10}\left(\frac{\|w - \widehat{w}\|^2}{\|w\|^2}\right) \quad (15)$$

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

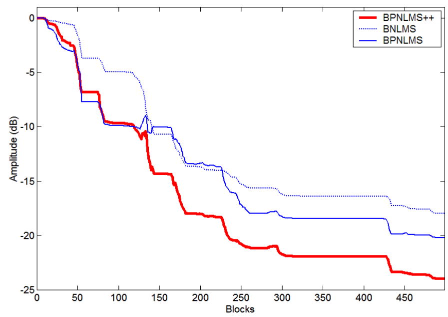

Figure V-3. Adaptation normalized misalignment

We can note that the BPNLMS++ adaptation, represented with bold plot, presents the faster adaptation convergence than Block-NLMS (BNLMS) or Block-PNLMS (BPNLMS) algorithms given with dotted and solid plots respectively. Moreover, it maintains a good quality of signal and involves only a modest increase in computational complexity.

B. FNT-Based Block Algorithm

As a realization of block adaptive filters incorporates fast convolution algorithms and appropriate sectioning of data, an implementation of BPNLMS_{++} filters using the FNT, instead of the FFT, could be considered for real data [10]. 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 [5]. The only difference in the FNT computation comes from the use of finite arithmetic modulo a Fermat number F_t, which can be implemented using conventional binary arithmetic.

However, as the fixed-point FFT of a b-bit real sequence is computed with both real and imaginary parts of a $\frac{b}{2}$-bit complex word, the computational complexities and convergence performances of the BPNLMS_{++} filtering realized by using b-bit FNT will be hardware computationally comparable to those implementing $\frac{b}{2}$-bit FFT. Then, a filter weight evolution for a b-bit FNT (bold plot) and a $\frac{b}{2}$-bit FFT (solid plot) implementations are compared in Figure V-4, where the results are shown for the word length b equal to 16 bits and the floating-point FFT (dotted plot) computation is given for reference of infinite precision arithmetic. In this test, all implementation types are simulated using the same filter parameters given in the previous part.

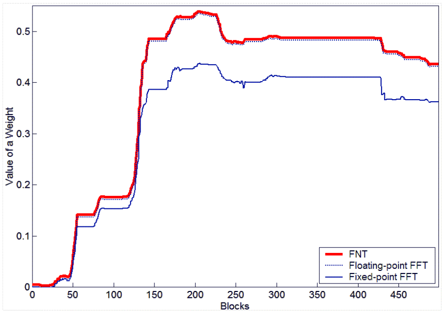

Figure V-4. Weight evolution of BPNLMS++ filter implementations

The weights evolution of the BPNLMS_{++} using 16-bit FNT are comparable to those using the floating-point FFT, whereas the performances of the 8-bits FFT-based realization will be not always acceptable. However, careful attention has been paid in our computer simulation to avoid overflows in arithmetic operations since they may cause serious performance degradations.

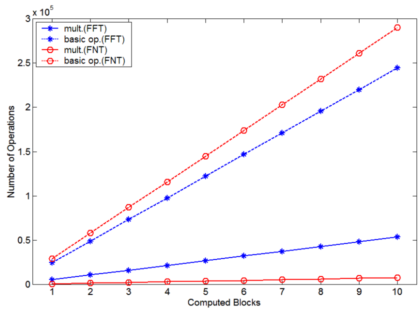

Figure V-5. Operations required for BPNLMS++ filtering

Although the recent processor DSP becomes more and more powerful, the multiplications remain more complicated than basic operations. Hence, the usual way to evaluate the different procedures computational cost is to consider the number of required multiplications. The computation gain being more significant for the filtering process as the computational complexity of various operations needed in implementing FFT or FNT-based BPNLMS_{++} filterings is about the same when the five convolutions are excluded, the overall computational efficiency of a realization is directly associated to the computational complexity of both different transforms. In Figure V-5, the operation number, required for the convolutions of the BPNLMS_{++} filter, is represented in dotted lines for simple operations (additions, bit shifts) and in solid lines for multiplications. The o and $*$ marker correspond respectively to FNT and FFT-based realizations.

In summary, the convergence properties of the FFT-based implementation degrade rapidly due to the finite word length effects instead of FNT-based realization. Moreover, this latter implementation of the BPNLMS_{++} filter is computationally more efficient due to the computational efficiency of the Fermat Number Transform.

VI. Conclusion

In this paper, we briefly show the proportionate adaptation and propose an algorithm class PNLMS_{++} with block structure. Here, our application is focused on acoustic echo cancellation problem, however the algorithms are valid for other scenarios as network echo cancellation where the double-talk should be considered [1]. Afterwards, a block adaptive filter has been derived to allow fast implementation while maintaining good performances. A BPNLMS_{++} filter has been developed to involve less computational complexity using fast convolution algorithms.

Moreover, an implementation using Fermat Number Transforms, which improve and simplify the fixed-point convolution procedures compared with FFT-based method, has allowed to realize a BPNLMS_{++} adaptive filtering with a considerably reduced computation complexity. The convergence properties of the FNT-based BPNLMS_{++} filter have been evaluated through computer simulations. Note that the FNT implementation could be beneficial to other adaptation types in filter processing.

VII. 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] S.L. Gay, "An efficient fast converging adaptive filter for network echo cancellation", Asilomar Conference on Signal, Systems and Computers, USA, 1998

[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] 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°2, 1974

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

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

[8] R. Blahut, "Fast algorithms for digital signal processing", Addison -Wesley Publishing Company, 1985.

[9] W. Shu, Y. Tianren, "Algorithm for linear Convolution using Number Theoretic Transforms", Electronics Letters, vol. 24, N°5, March 1988.

[10] J.C. Clark, B.K. Min, M. Suk, "Realization of Adaptive Digital Filters using the Fermat Number Transform", IEEE Trans. on Acoust., Speech, Signal Proc., vol. ASSP-33, N°4, August 1985