IEEE 4th Eurasip Conf. on Video, Image Processing and Multimedia Communications, 2-5 July 2003, Vol 2, pp. 607-612, Zagreb, Croatia

LINEAR PREDICTIVE SPEECH CODING USING FERMAT NUMBER TRANSFORM

G. Madre, E.H. Baghious, S. Azou and G. Burel

Laboratoire d'Electronique et Systèmes de Télécommunications - UMR CNRS 6165
6, avenue Le Gorgeu - BP 809 - 29285 BREST cedex - FRANCE
e-mail : guillaume.madre@univ-brest.fr

Abstract - *This paper is about the reduction of the computational complexity of a speech codec. A Linear Predictive Coding procedure is developed to allow its implementation with Number Theoretic Transforms. The use of Fermat Number Transform can reduce, in a significant way, the cost of Linear Predictive algorithm implantation on Digital Signal Processor.*

Key words - *Speech Coding, Linear Prediction, Autocorrelation, Number Theoretic Transform, Fermat Number Transform, CS-ACELP codec G.729.*

1. INTRODUCTION

An important aspect of modern telecommunications is speech coding, which is defined as the process of digitally representing an analogical speech signal in an efficient way. Its objective is to represent the speech signal with the lowest number of bits, while maintening a sufficient level of quality of the synthesized speech with reasonable computational complexity. Most current speech coders are based on Linear Predictive Coding (LPC) analysis due to its simplicity and high performance.

It is known that the significant part of coders complexity is due to the calculation of the Linear Prediction (LP) parameters. Therefore, in any application of real time speech signal analysis, the computation speed of LP coefficients must be increased.

To reduce the complexity of LP analysis and design an efficient speech coding in fixed-point arithmetic, different algorithms are implemented with Number Theoretic Transforms (NTT), which present the following advantages compared to Discrete Fourier Transforms (DFT) [1] :

- They require few or no multiplications.
- They suppress the use of floating point complex numbers.
- All calculations are performed on a finite ring of integers, which is interesting for implantation on Digital Signal Processor (DSP).

Hence, the use of NTT 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 the calculations of convolution and correlation can provide real benefits for low DSP implantation cost.

Then, we propose a procedure using FNT to determine the LP parameters, by computing the autocorrelation coefficients with a fast algorithm [2] and by implementing a simplified procedure based on [3].

The rest of the paper is organized as follows : In section 2, we will introduce the Linear Prediction Analysis used in many speech coders. In a third part, we will present an algorithm used to determine LP coefficients. Section 4 presents the concept of Number Theoretic Transform and details the Fermat Number Transform, which is implemented in LP modeling. In the final part, numerical results for the new procedure are given and discussed for its implantation in $CS - ACELP$ codec.

2. LINEAR PREDICTION ANALYSIS

Linear Predictive Coding exploits the redundancies of speech signal by modelling the vocal tract as a linear filter, known as an AutoRegressive (AR) model. Speech coders perform an LPC analysis on each frame (typical duration $10 - 20ms$) of speech to obtain LP coefficients and extract the *residual signal*, which is the output of analysis filter.

These LPC parameters are quantized and transmitted to the decoder, in which the *residual signal* becomes the excitation signal for the LP synthesis filter (Figure 2.1).

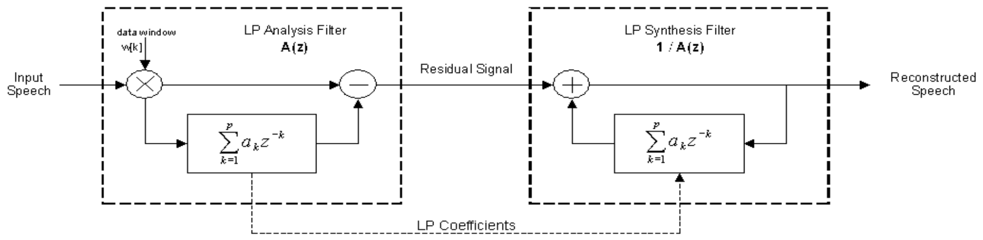

Fig. 2.1. *LP* analysis synthesis model

The most popular and efficient techniques for estimating the *LP* parameters are the *autocorrelation* or *covariance* methods. Both choose the short term filter coefficients $\{a_k\}$ by minimizing the energy E of residual signal by least square calculation :

$$E = \sum_{n=-\infty}^{+\infty} \left(s_w(n) - \sum_{k=1}^{p} a_k s_w(n-k) \right)^2 \tag{1}$$

where p is the order of LPC model and s_w denotes a windowed speech segment, which reduces the audible distortion of reconstructed speech. The *autocorrelation* method yields the LP coefficients by resolving the following p linear equations :

$$r(k) = -\sum_{n=1}^{p} r(|n-k|)\, a_n \qquad \text{with } 1 \leqslant k \leqslant p \tag{2}$$

where $r(i) = \sum_{n=i}^{N_w} s_w(n)\, s_w(n-i)$ with N_w the length of the analysis window and $0 \leqslant i \leqslant p$. The set of equations can be represented in the following form :

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_{p-1} \\ r_1 & r_0 & \cdots & r_{p-2} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p-1} & r_{p-2} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = R \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ \cdots \\ r_p \end{bmatrix} \tag{3}$$

The resulting matrix R is symmetric Toeplitz, which implies that the filter $A(z)$ is minimum phase and guarantees the stability of synthesis filter.

3. SOLUTION OF TOEPLITZ SYSTEM

The previous (pxp) Toeplitz system of linear equations may be solved by classical methods [4] (Gauss, Cholesky, ...) in order p^3 operations or by fast algorithms [4] [5] (Levinson-Durbin, Schur, Trench, ...) of reduced computational complexity in order p^2 flops.
Kumar [3] has presented a method, of complexity $\mathcal{O}(p(\log_2 p)^2)$ computations, using Fourier transform techniques, based on the inverse *Trench* algorithm [6]. Thanks to the symmetry of LP system, we propose to simplify the *Kumar* algorithm as presented below.

To compute the inverse of the (pxp) Toeplitz matrix R, an (LxL) circulant matrix $\overline{R}$ (shift to the right) is constructed, with $L = 2p - 1$. The first line (or column) is given by $\overline{R}_1 = [r_0 r_1 ... r_{p-1} r_{p-1} r_{p-2} ... r_1]$.

$$\overline{R} = \left[\begin{array}{ccc|ccc} & & & r_{p-1} & \cdot & r_1 \\ & \mathbf{R} & & \cdots & \cdot & \cdots \\ & & & \cdots & \cdot & \cdots \\ & & & r_1 & \cdot & r_{p-1} \\ \hline r_{p-1} & \cdot & r_1 & r_0 & \cdot & r_{p-2} \\ \cdots & \cdot & \cdots & \cdots & \cdot & \cdots \\ r_1 & \cdot & r_{p-1} & r_{p-2} & \cdot & r_0 \end{array} \right] \tag{4}$$

Then, we proceed in three steps to the desired solution directly from first row $\overline{R}_1$ by applying DFTs :

First step - The first column $\overline{Q}_1$ of the inverse matrix $\overline{R}$ is computed first. $\overline{R}$ being a cyclic shift matrix, the correlation of elements of $\overline{R}_1$ and $\overline{Q}_1$ is equal to the diagonal of the matrix identity. It is known that $\overline{Q}_1$ can be computed as $\overline{Q}_1 = DFT^{-1}\left(1./DFT\left(\overline{R}_1\right)\right)$ where $1./S_q$ denotes the term by term inverse of the sequence S_q.

This calculation requires about $2p\,(\log_2 p + 1)$ multiplications. However, the symmetry of LP system implies all values are real. Then, the transforms can be replaced by a p-sample cosinus table and the vector $\overline{Q}_1$ will be quickly computed with twice less multiplications.

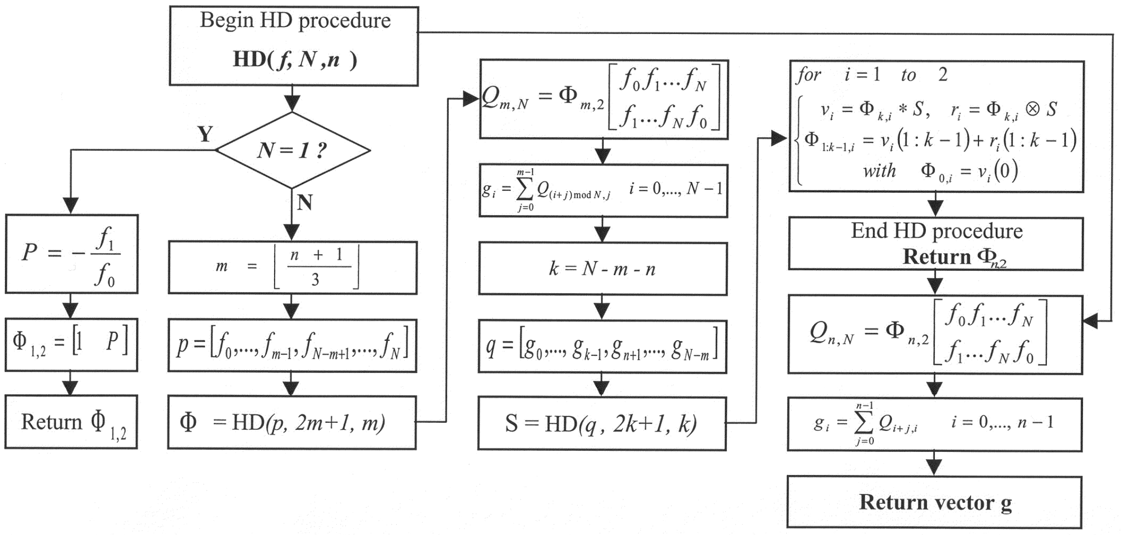
Begin HD procedure
HD(f, N, n)
N=1 ?
Y
N
$P = -\frac{f_1}{f_0}$
$\Phi_{1,2} = [1 \;\; P]$
Return $\Phi_{1,2}$
$m = \left\lfloor \frac{n+1}{3} \right\rfloor$
$p = [f_0,\ldots,f_{m-1},f_{N-m+1},\ldots,f_N]$
$\Phi = HD(p, 2m+1, m)$
$Q_{m,N} = \Phi_{m,2}\begin{bmatrix} f_0 f_1 \cdots f_N \\ f_1 \cdots f_N f_0 \end{bmatrix}$
$g_i = \sum_{j=0}^{m-1} Q_{(i+j) \bmod N, j} \quad i = 0,\ldots,N-1$
$k = N - m - n$
$q = [g_0,\ldots,g_{k-1},g_{n+1},\ldots,g_{N-m}]$
$S = HD(q, 2k+1, k)$
for $i = 1$ to 2
$v_i = \Phi_{k,i} * S, \quad r_i = \Phi_{k,i} \circledast S$
$\Phi_{1:k-1,i} = v_i(1:k-1) + r_i(1:k-1)$
with $\Phi_{0,i} = v_i(0)$
End HD procedure
Return $\Phi_{n,2}$
$Q_{n,N} = \Phi_{n,2}\begin{bmatrix} f_0 f_1 \cdots f_N \\ f_1 \cdots f_N f_0 \end{bmatrix}$
$g_i = \sum_{j=0}^{n-1} Q_{i+j,i} \quad i = 0,\ldots,n-1$
Return vector g

Fig. 3.1. Flowchart of HD procedure

Second step - By using the persymmetry of the Toeplitz matrix R, the first column Q_1 of R^{-1} is determined from column $\overline{R}_1$. The $(L\text{x}L)$ matrices $\overline{R}$ and $\overline{Q}$ are partitioned and the inverses of the submatrix of $\overline{R}$ are computed in a recursive manner until the order $n-1$ is reached $\left(\mathcal{Q}_{22}^{n-1} = Q = R^{-1}\right)$:

$$\mathcal{R}_{22}^{i-1} = \begin{bmatrix} \mathcal{R}_{11}^{i} & \mathcal{R}_{12}^{i} \\ \mathcal{R}_{21}^{i} & \mathcal{R}_{22}^{i} \end{bmatrix} \quad \text{and} \quad \mathcal{Q}_{22}^{i-1} = \begin{bmatrix} \mathcal{Q}_{11}^{i} & \mathcal{Q}_{12}^{i} \\ \mathcal{Q}_{21}^{i} & \mathcal{Q}_{22}^{i} \end{bmatrix} \quad \text{with } 1 \leqslant i \leqslant n-1 \tag{5}$$

where $\mathcal{R}_{11}^{i}$ and $\mathcal{Q}_{11}^{i}$ are scalar numbers, $\mathcal{R}_{21}^{i} = (\mathcal{R}_{12}^{i})^T$ and $\mathcal{Q}_{21}^{i} = (\mathcal{Q}_{12}^{i})^T$. By formulating $\mathcal{R}_{22}^{0} = \overline{R}$ and $\mathcal{Q}_{22}^{0} = \overline{Q} = \overline{R}^{-1}$, the submatrices $\mathcal{Q}_{22}^{i}$ are obtained as :

$$\mathcal{Q}_{22}^{i} = \mathcal{Q}_{22}^{i-1} - \tfrac{1}{\mathcal{Q}_{11}^{i}}\left(\mathcal{Q}_{21}^{i}\mathcal{Q}_{12}^{i}\right) \qquad \text{with } 1 \leqslant i \leqslant n-1 \tag{6}$$

Vector Q_1 is obtained by an efficient HD (Half Divisor) procedure, which is derived from $HGCD$ (Half Greatest Common Divisor) algorithm [7]. The procedure, described in Figure 3.1, is called with the following parameters $\left(f = \overline{Q}_1, N = 2p-1, n = p\right)$ and return the vector $g = Q_1$.

The matrices Φ are computed with convolutions and correlations, denoted $*$ and $\circledast$ respectively, and can be carried out Fast Fourier Transform (FFT) techniques. The presented HD algorithm requires about twice fewer operations than HD algorithm in [3].

Third step - The solution of the system is obtained after three stages :

- One L-dimensional new vector is defined as $h = [g_{p-1}g_{p-2}...g_1g_0g_1...g_{p-1}]$
- The convolution $\overline{u} = h * y$ is calculated to obtain a p-elements vector $u = [\overline{u}_p\overline{u}_{p+1}...\overline{u}_L]$
- Compute $\overline{v} = g \circledast y$ and $\overline{w} = g * y$ and determine two $(p-1)$-dimensional vectors :
$$\widetilde{v} = [\overline{v}_p\overline{v}_{p+1}...\overline{v}_{L-1}] \text{ and } \widetilde{w} = [\overline{w}_{2p}...\overline{w}_{L+p-1}]$$
- Calculate $v = g * \widetilde{v}$ and $w = g \circledast \widetilde{w}$ and deduce the solution of the system as :
$$a_k = u_{k-1} + \frac{(v_{k-2} - w_{k-1})}{g_0}$$
with $2 \leqslant k \leqslant p$ and the first values of LP coefficients $a_0 = 1$ and $a_1 = u_{k-1}$.

To implement the last step by Fourier transform techniques, 5 L-point Discrete Fourier Transforms, complex products and inverse $DFTs$ are required.

4. NUMBER THEORETIC TRANSFORM

To develop the whole LPC procedure in fixed-point arithmetic with low computational complexity, we propose to use Number Theoretic Transforms to compute the autocorrelation coefficients. Afterwards, we will implement the previous algorithm with FNT, which will replace the DFT in convolution and correlation computations.

An NTT [1] presents the same form as a DFT but is defined over finite rings. All arithmetic must be carried out modulo M, which may be equal to a prime number or to a multiple of primes, since NTT is defined over Galois Field $GF(M)$. Note that NTT admits the Cyclic Convolution Property (CCP) [5].

The 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 term α, which satisfy the equality $\langle \alpha^N = 1 \rangle_M$ where $\langle . \rangle_M$ denotes the modulo M operation. An NTT of a discrete time signal x and its inverse are given respectively by :

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\alpha^{nk} \right\rangle_M \qquad \text{with } k = 0, 1, ..., N-1 \tag{7}$$

$$x(n) = \left\langle N^{-1} \sum_{k=0}^{N-1} X(k)\alpha^{-nk} \right\rangle_M \qquad \text{with } n = 0, 1, ..., N-1 \tag{8}$$

where parameters α and N represent the generating term and the length of the tranform respectively.

4.1. Fermat Number Transform

For an efficient implantation on processor (DSP), the choice of parameters is crucial. The choice of modulo equal to a Fermat number [1], $F_t = 2^{2^t} + 1$ with $t \in \mathbb{N}$, implies the values of N and α equal to a power of 2 (Table I), hence allowing the replacement of multiplications by bit shifts. The values of N and α associated to a Fermat Number Transform (FNT) defined over the Galois Field $GF(F_t)$ are given by $N = 2^{t+1-i}$ and $\alpha = 2^{2^i}$ with $i < t$.

Table I. Possible combinations of FNT parameters

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

A Fermat Number Transform needs about $N \log_2 N$ simple operations (bit shifts, additions) but no multiplication, while a DFT requires in order $N \log_2 N$ multiplications. Note that fast FNT-type computational structure ($FFNT$) similar to the Fast Fourier Transform exists. Available FFT $VLSI$ hardware structure for real-time implementation of the $FFNT$ may be adopted. About 30% duration reduction to convolve two sequences using $FFNT$ instead of FFT implementation can be envisaged [1].

4.2. Fast autocorrelation

The classical method for computing correlation uses the Cyclic Convolution Property. The autocorrelation of a sequence of N samples generates $2N - 1$ output numbers and requires the use of Number Theoretic Transform of length $2N$ (the sequences are extended by zeros). To avoid adding zeros, a fast computation of the autocorrelation coefficients is implemented. We described an efficient algorithm [2], modified to allow more flexible segmentation of the speech signal frame.

To compute the autocorrelation $r_{xx}(k) = \sum_{j=0}^{N-1} x(j)x(k+j)$ to the p^{th} order, the presented method, based on the segmentation of the frame, computes n_i-points autocorrelation analysis. Let us determine the lengths n_i, applicable to NTT, of such that $N \leqslant \sum_{i=1}^{i_{\max}} n_i - p$ with $0 < n_i < n_{i+1}$. We will be able to choose $n_i = n_{i+1}$ if $n_{i+1} = n_{i_{\max}}$. Let us form new frames for $i = 1, ..., i_{\max}$:

- $x_{i,2}(j) = \begin{cases} x(j + d_i) & \text{with } d_i = (i-1)(n_{i-1} - p) \text{ and } 0 \leqslant j < n_i - p \\ 0 & \text{for } n_i - p \leqslant j < n_i \end{cases}$
- $\begin{cases} \text{if } i \neq i_{\max} : & x_{i,1}(j) = x(j + d_i) & \text{for } 0 \leqslant j < n_i \\ \text{if } i = i_{\max} : & x_{i,1}(j) = x_{i,2}(j) & \text{for } 0 \leqslant j < n_i \end{cases}$

The Number Theoretic Transforms are used afterwards. The autocorrelation is written as :

$$r_{xx}(k) = \sum_{i=1}^{i_{\max}} \sum_{j=0}^{n_i - 1} x_{i,1}(j)x_{i,2}(k+j) \qquad \text{with } 0 \leqslant k \leqslant p-1 \tag{9}$$

For $i = 1, ..., i_{\max} - 1$, we calculate $X_{i,1}$ and $X_{i,2}$, the respective NTTs of sequences $x_{i,1}$ and $x_{i,2}$. Next, we compute $Y_i(j) = X_{i,1}(j)X_{i,2}(N_i - j)$ where $X_{i,2}(N_i) = X_{i,2}(0)$ with $j = 0, ..., N_i - 1$. Afterwards, the inverse transform y_i of Y_i makes it possible to obtain the p^{th} order autocorrelation coefficients, which are equal to $r_{xx}(k) = \sum_{i=1}^{i_{\max}} y_i(k)$ with $k = 0, ..., p-1$.

Using the FNT reduces considerably the computational complexity of speech signal frame autocorrelation function. Compared to method using CCP with NTT, this fast autocorrelation computation requires twice fewer multiplications.

5. NUMERICAL RESULTS

Numerical simulations have been conducted to evaluate the performances of Linear Prediction Coding using FNT. This LP analysis has been implemented in the $8kbit/s\ CS-ACELP$ Coder, described in International Telecommunication Union (ITU) recommendation $G.729$ [8]. The coder operates on speech frames of $10ms$ duration, which corresponds to 80 samples at a sampling rate of $8kHz$. The input speech signal is analyzed within each frame to extract the 10^{th} order linear prediction filter coefficients, which are converted to Line Spectral Frequencies (LSF) and quantized. Afterwards, the excitation parameters (Pitch delay, fixed codebook indices and gains) are estimated on a $5ms$ subframe basis.

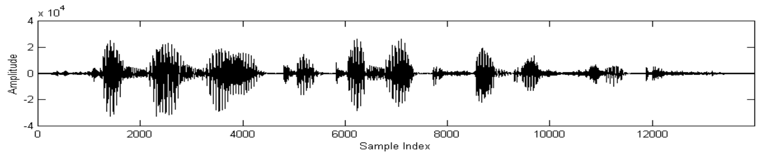

Fig. 5.1. Input sentence pronounced by Male Talker

In these tests, we compare the standard analysis using Levinson-Durbin algorithm [4] [8] to the proposed LPC procedure. Both methods are compared, for the same sentence (Figure 5.1), by computing the *prediction gain, Signal to Noise Ratio* (SNR), *spectral distortion* and *euclidean LSF distance* measures.

5.1. Implementation of FNT-based LPC Model

Although DSP becomes more and more powerful, implantation of multiplications or divisions remains more complicated than additions or shifts. Hence, to evaluate the computational cost of our LPC procedure, we will consider the number of multiplications only.
Linear Predictive Coding computes predictive parameters using autocorrelation coefficients. To calculate the 11^{th} order autocorrelation coefficients of a 240-sample windowed signal, we propose to use the previous fast algorithm with one 256-sample segment. The coefficients $\{a_i\}$ will be computed with two 256-point $FNTs$. Only 256 multiplications are required to obtain the autocorrelation coefficients.

Afterwards, to determine LP parameters, the simplified *Kumar* algorithm, which already requires fewer multiplications than other methods, can still be improved using FNT. A fast similar algorithm is developed to operate with 32-bit numbers.
For the dimension $p = 10$, the HD procedure, where the vectors g can be computed with fast matrix multiplication algorithm using FNT [9], is called 17 times every frames. All arithmetic is carried out modulo F_5 and the FNT replaces DFT in convolution and correlation computations. However, to avoid the number representation problems met in [10], we accept to round off Q_1 in the first step and P in HD algorithm by calculating 32-bit integer divisions, instead of modulo divisions defined over $GF(F_5)$. These approximations do not introduce significant degradation in the algorithm performance.
Thanks to the use of FNT framework, the proposed algorithm of complexity $\mathcal{O}\left(p\log_2 p\right)$ fixed-point operations gives other advantages and can be computed, through a parallel processing the procedure, in less than $7p$ clock cycles.

5.2. LPC Analysis Comparisons

After presenting a procedure allowing low costs of implantation and computation, we are going to verify that the different implementations of LPC analysis do not induce a significant distortion in the synthesized signal. To compare the coefficient values and the speech signal degradation, the LP modelling procedures have been progammed into *MatLab* software. The *SNR* and *euclidean LSF distance* measures reveal a slight difference between floating-point $G.729$ method and 32-bit fixed-point proposed procedure.

In Figure 5.2, the solid plot represents the SNR obtained through codec using the new procedure and the dashed curve is computed with $G.729$ LPC analysis :

$$SNR = 10\log_{10}\left(\frac{\sum_{i=0}^{79} s_i^2}{\sum_{i=0}^{79}(s_i-\widehat{s}_i)^2}\right) \tag{10}$$

where s_i and $\widehat{s}_i$ are frame samples of input speech signal and of reconstructed speech respectively.

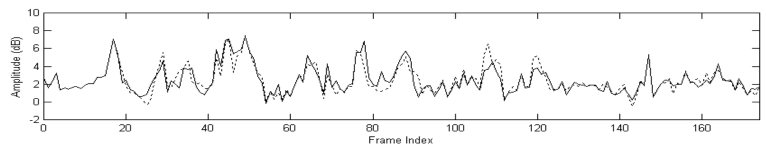

Fig. 5.2. SNR between input and output signals of $G.729$ codec

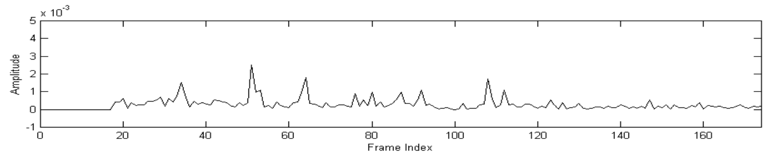

Fig. 5.3. Euclidean LSF distance measure

The proposed procedure involves minuscule audible distortion in the synthesized signal. We confirm that both methods give very close results by computing a *euclidean LSF distance* (Figure 5.3) defined as :

$$d_{LSF} = \sum_{i=1}^{p=10} \left(w_i^{(1)} \left(w_i^{(1)} - w_i^{(2)} \right) \right)^2 \tag{11}$$

where $w_i^{(1)}$ and $w_i^{(2)}$ are 10^{th} order quantized LSF coefficients obtained using LPC analysis, with Levinson-Durbin and proposed procedure respectively, and the Vector Quantization described in [8].

6. CONCLUSION

In this paper, an efficient implementation of a Linear Predictive Coding procedure in speech signal processing has been presented. Fast algorithms have been implemented to determine autocorrelation and Linear Prediction coefficients. Then, a fixed-point HD procedure has been developed to resolve symetric Toeplitz system of equations.
To limit the implantation cost of the LP analysis, the use of Number Theoretic Transforms has been proposed. In particular, it is shown that Fermat Number Transforms reduce the computational complexity of various algorithms involved in the LPC modelling. The FNT, which may be implemented with fast hardware structure for real-time application, allows a LP analysis much faster than the existing methods.

Next stages would be to implant the procedure into Digital Signal Processor. A subjective comparison of speech quality should be made in the form of Mean Opinion Score (MOS).

Although the selected application is the $CS-ACELP$ $G.729$ coder, the proposed LPC procedure can be adapted to other types of speech coder or LP system of different order. Other areas where fast computations are required could also take benefit of a FNT implementation.

REFERENCES

[1] R.C. Agarwal, C.S. Burrus, "Fast convolution using Fermat number transform with application to digital filtering", IEEE Trans. on Acoustics, Speech and Signal Proc., ASSP-22, N°2, pp. 87-97, 1974.
[2] S. Xu, L. Dai, S.C. Lee, "Autocorrelation Analysis of Speech Signals Using Fermat Number Transform", IEEE Transactions on Signal Processing, vol. 40, N°8, August 1992.
[3] R. Kumar, "A fast algorithm for solving a Toeplitz system of equations", IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-33, pp. 254-267, Feb. 1985.
[4] G. Golub, C Van Loan, "Matrix computations", North Oxford Academic, 1983.
[5] R. Blahut, "Fast algorithms for digital signal processing", Addison-Wesley Publishing Company, 1985.
[6] W. Trench, "An algorithm for the inversion of finite Toeplitz matrices", J. Soc. Indust. Appl. Math., vol. 12, N°3, pp. 515-522, 1964.
[7] A.V. Aho, J.E. Hopcroft, J.D. Ullman, "The design and analysis of computer algorithms", Addison-Wesley Publishing Company, 1974.
[8] ITU-T Rec. G.729, "Coding of Speech at 8 kbits/s using Conjugate Algebraic Code-Excited Linear Prediction (CS-ACELP)", 1996.
[9] A.E. Yagle, "Fast Algorithms for Matrix Multiplication using Pseudo-Number-Theoretic Transforms", IEEE Trans. on Signal Processing, vol. 43, N°1, January 1995.
[10] J.-J. Hsue, A.E. Yagle, "Fast algorithms for solving Toeplitz systems of equations using number-theoretic transforms", Signal Proc., vol. 44, pp. 89-101, 1995.