

Cyclic Convolution Algorithm Formulations Using Polynomial Transform Theory

Abraham H. Diaz-Perez

Popular University of the Cesar/ Electronic Department, Sabanas Campus, Valledupar-Cesar, Colombia
Email: abraham.diaz@gmail.com

Domingo Rodriguez

University of Puerto Rico/ Electrical and Computer Engineering Department, Mayagüez PR. 00681-9042
E-mail: domingo@ece.uprm.edu

Abstract— This work presents a mathematical framework for the development of efficient algorithms for cyclic convolution computations. The framework is based on the Chinese Remainder Theorem (CRT) and the Winograd's Minimal Multiplicative Complexity Theorem, obtaining a set of formulations that simplify cyclic convolution (CC) computations. In particular, this work focuses on the arithmetic complexity of a matrix-vector product when this product represents a CC computational operation or it represents a polynomial multiplication modulo the polynomial z^N-1 , where N represents the maximum length of each polynomial factor and it is set to be a power of 2. The proposed algorithms are compared against existing algorithms developed making use of the CRT and it is shown that these proposed algorithms exhibit an advantage in computational efficiency. They are also compared against other algorithms that make use of the Fast Fourier Transform (FFT) to perform indirect CC operations, thus, demonstrating some of the advantages of the proposed development framework.

Index Terms—Cyclic Convolution, Fast Fourier Transform, Circulant Matrix, Winograd's Theorem, Chinese Remainder Theorem.

I. INTRODUCTION

In digital signal processing, the design of fast and computationally efficient algorithms has been a major focus of research activity. The objective, in most cases, is the design of algorithms and their respective implementation in a manner that perform the required computations in the least amount of time. In order to achieve this goal, parallel processing has also received a lot attention in the research community [1].

This paper summarizes the main properties of the CC, or the polynomial multiplication modulo the polynomial z^N-1 , when they are represented by a vector-matrix multiplication operation using a circulant matrix [2].

Direct computation of a matrix-vector product takes N^2 complex multiplications; however, by exploiting the special structure of a circulant matrix, the computational effort could be substantially decreased for large matrices. One approach is to use the fast Fourier transform (FFT), which makes possible to compute a matrix-vector product in $(N)\log_2(N)$ complex multiplications for a matrix of order N . In this work algorithms are developed by means of a decimation in time approach, and the use of the roots of the unity (factoring the polynomial z^N-1 in the complex field). They require only N multiplications, with some advantages over FFT such as memory use and addressing techniques.

In the present work the CRT and the Winograd's theorem are used for the formulation of new and computationally efficient algorithms for matrix-vector products when they represent cyclic convolution operations. The mappings of the resulting operations onto signal flow diagrams that imply parallel computation are also remarked.

This document is organized as follows. First, mathematical foundations needed for the study of algorithms to compute the *discrete convolution* are summarized. Second, an identification is established between products of polynomials and the convolution operation. Third, the algorithm development for the basic problem of the multiplication of a circulant matrix by a vector, using the conceptual framework developed in the two previous sections, is explained. The section also presents several signal flow diagrams that may be implemented in diverse architectures by means of very large scale integration (VLSI) or very high speed integrated circuits hardware description language (VHDL) [3]-[8]. Conclusions, contributions, and future development of the present work are then summarized.

Based on the document "One Dimensional Cyclic Convolution Algorithms with Minimal Multiplicative Complexity by Abraham H. Diaz-Perez and Domingo Rodriguez," this appeared in the Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 2006. © 2006 IEEE.

II. THEORETICAL FRAMEWORK

A. Discrete Linear Convolution

A block diagram representing a basic discrete system is depicted in **Figure 1** below. A discrete system is defined here as an entity which acts, transforms, or operates on a input signal, termed the input signal in order to produce another signal, termed the output signal [9]. An important class of discrete systems is linear and shift-invariant systems known as discrete filters. A discrete filter is uniquely described by its impulse response signal, denoted by $h[n]$, $n \in Z$, where Z is the set of integers. A discrete filter's impulse response is obtained as a resulting output signal when the input to the filter is a delta function $\delta[n]$, $n \in Z$, with $\delta[n]=1$ for $n=0$ and $\delta[n]=0$ for $n \neq 0$. Consider an arbitrary input signal $x[n]$, $n \in Z$, to a discrete filter with an associated impulse response signal equal to $h[n]$, $n \in Z$. The output signal, say $y[n]$, $n \in Z$, of the discrete filter is given by the following general expression.

$$y[n] = \sum_{m=-\infty}^{m=+\infty} x[m]h[n-m] = \sum_{m=-\infty}^{m=+\infty} h[m]x[n-m], \quad n \in Z$$

This operation is commonly known as the *linear convolution sum operation* of the input signal $x[n]$, $n \in Z$ with the impulse response signal $h[n]$, $n \in Z$ and it is commutative operation.

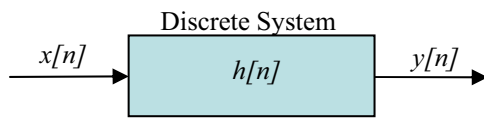


Figure 1

The set of all discrete complex signals of the type $x[n]$, $n \in Z$ becomes a linear space denoted by $l(Z)$. A subspace of this linear space, denoted by $l^2(Z)$, is the set of all discrete signals with finite energy. A discrete signal, say $x[n]$, $n \in Z$, is said to have finite energy if the condition $\sum_{n=-\infty}^{n=+\infty} x[n]x^*[n] = \langle x, x \rangle < \infty$ is satisfied. The symbol “*” in the expression above denotes complex conjugation. The expression $\langle x, y \rangle$ is termed an inner product of x and y , with, both, $x, y \in l^2(Z)$. The norm or length of a finite energy discrete signal $x \in l^2(Z)$ is denoted by $\|x\| = \langle x, x \rangle^{\frac{1}{2}}$.

A very important subset of the linear space $l^2(Z)$ is the set of finite discrete complex signals of length or order, say, N . This subset is denoted by $l^2(Z_N)$ and it is itself a linear space of vector space of dimension N . The set $Z_N = \{0, 1, 2, \dots, N-1\}$ is called the standard finite indexing set. Each element of the space $l^2(Z_N)$ is of the form $x: Z_N \rightarrow C$, where C is the set of complex numbers and the valuation $x[n]$ is a complex number. A signal x is represented in this work as a column vector.

An inner product of two finite sequences, say x and y , can be defined in the linear space or vector space $l^2(Z)$ through the expression $\langle x, y \rangle = \sum_{n=0}^{N-1} x[n]y^*[n] \neq \langle y, x \rangle$, with $\langle x, y \rangle \in C$ being a complex number. The length or norm of a finite signal $x \in l^2(Z_N)$ is then $\|x\| = \langle x, x \rangle^{\frac{1}{2}}$. An identification is made in this work between the linear space or vector space $l^2(Z)$, the vectors belonging to the complex Euclidean space C^N , and the finite dimensional polynomial algebra $C[x]/x^N - 1$, the ring of polynomials $C[x]$ modulo the monic polynomial $x^N - 1$. The inner product $\langle x, y \rangle$, $x, y \in l^2(Z_N)$ can be then identified with the standard Euclidean metric in C^N .

It is important to point out that the linear space $l^2(Z)$ can be turned into a finite dimensional linear algebra by introducing a vector multiplication operation [10]. There are many operations that can be used as a multiplication operation in $l^2(Z)$. One of the most useful multiplication operations utilized is the cyclic convolution operation modulo N . In the next section this cyclic convolution operation is introduced along with some of its most important properties.

B. Periodic or Cyclic Convolution

Let $x, h \in l^2(Z_N)$ be two arbitrary sequences, each of length N . The periodic or cyclic convolution modulo N of these two signals is denoted by the expression $x \otimes_N h$ and it is a new signal, say y , also of length N , defined by the following expression for any $n \in Z_N$:

$$y[n] = (x \otimes_N h)[n] = \sum_{m=0}^{N-1} x[m]h[\langle n-m \rangle_N]$$

or

$$y[n] = (h \otimes_N x)[n] = \sum_{m=0}^{N-1} h[m]x[\langle n-m \rangle_N]$$

Here, the notation $\langle p \rangle_N$, implies the remainder p after being divided by N .

The focus of this work is to develop fast and efficient algorithms for the computation of the circular or cyclic convolution operation, reaching the minimal number of multiplications according to the Winograd's theorem. Normally, two approaches are utilized to compute the cyclic convolution operation, namely, the direct approach and the transform approach. The direct approach evaluates the equation for the cyclic convolution of two N -point signals for each value $n \in Z_N$ resulting in a system of equations. The transform approach establishes a discrete Fourier transform (DFT) isomorphism between the cyclic convolution operation two signals in the object domain and the point-by-point multiplication operation or Hadamard product of each of the transformed signals.

C. Matrix Representation of the Cyclic Convolution

This section describes a cyclic convolution operator as a linear shift invariant (LSI) operator acting on the finite dimensional linear space $l^2(Z_N)$. In addition, the cyclic convolution operator is also described as a cyclic finite impulse response (FIR) system. Combining these two attributes allows for a deeper study of the properties of the cyclic convolution operator. A formal discussion follows, arriving at a matrix representation of a cyclic convolution operation [11].

Since each N -dimensional LSI-FIR system $T_h: l^2(Z_N) \rightarrow l^2(Z_N)$ describes a linear mapping on the space $L(Z_N)$, each T_h is uniquely determined by its action on a set of basis vectors (signals) spanning $l^2(Z_N)$. If the standard basis set $\{\delta_{\{j\}}: j \in Z_N\}$ is chosen as reference, then each signal $T_h(\delta_{\{j\}}) \in L(Z_N)$ can be uniquely expressed as a linear combination of the basis set:

$$T_h\{\delta_{\{k\}}\} = \sum_{j \in Z_N} h[j, k] \delta_{\{j\}},$$

where the set of scalars $\{h[j, k]: j \in Z_N, k \in Z_N\}$ actually represents the vector coordinates of the given signal $T_h\{\delta_{\{k\}}\}, k \in Z_N$, with respect to the standard basis set. The signal $T_h\{\delta_{\{k\}}\}$ can be written as follows:

$$T_h\{\delta_{\{k\}}\} = \sum_{j \in Z_N} T_h\{\delta_{\{k\}}\}[j] \delta_{\{j\}},$$

where

$$\begin{aligned} T_h\{\delta_{\{k\}}\}[j] &= \sum_{m \in Z_N} h[m] (S_N^m \{\delta_{\{k\}}\})[j] \\ T_h\{\delta_{\{k\}}\}[j] &= \sum_{m \in Z_N} h[m] \delta_{\{k+m\}}[j] = \\ &= \sum_{m \in Z_N} h[m] \delta_{\{j-k-m\}} = h[j-k] = (S_N^k \{h\})[j] \end{aligned}$$

Thus, the following formulation may be obtained

$$\begin{aligned} T_h\{\delta_{\{k\}}\} &= \sum_{j \in Z_N} h[j, k] \delta_{\{j\}} = \sum_{j \in Z_N} h[j-k] \delta_{\{j\}} \\ &= \sum_{j \in Z_N} (S_N^k \{h\})[j] \delta_{\{j\}} = T_{(S_N^k \{h\})} \{\delta_{\{j\}}\} = S_N^k \{h\} \end{aligned}$$

It is important to point out that the expression $S_N^m \{\delta_{\{k\}}\}$ represents the action of the *cyclic shift operation* over an the k -th element of the ordered standard basis set. This operator action is formally defined as follows:

$$\begin{aligned} S_N^m : l^2(Z_N) &\rightarrow l^2(Z_N) \\ \delta_{\{k\}} &\mapsto S_N^m \{\delta_{\{k\}}\} = \delta_{\{(k+m)_N\}} \end{aligned}$$

The notation $\langle \rangle_N$ denotes modulo N arithmetic, which we will omit most of the time for better readability.

Next, the matrix H_N formally is defined as follows:

$$H_N = [h[j, k]]_{j, k \in Z_N} = [h[j-k]]_{j, k \in Z_N}$$

The matrix H_N , thus, have the following form

$$H_N = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \dots & h[1] \\ h[1] & h[0] & h[N-1] & \dots & h[2] \\ h[2] & h[1] & h[0] & \dots & h[3] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \dots & h[0] \end{bmatrix}$$

Notice that the columns of H_N are formed by cyclic shifted versions of the coordinate vector representation of the signal h with respect to the standard basis set; that is, the matrix H_N can be written as the following ordered action of the cyclic shift operator:

$$H_N = [I_N \{h\}, S_N \{h\}, S_N^2 \{h\}, \dots, S_N^{N-1} \{h\}]$$

To describe in more details how the matrix H_N , representing the system T_h , is obtained, the action of the operator over an element of the standard basis is formulated as follows:

$$T_h(\delta_{\{k\}}) = \sum_{j \in Z_N} h[j, k] S_N^j \{\delta_{\{j\}}\}, h[j, k] \in C$$

$$T_h(\delta_{\{k\}}) = h[0, k] \delta_{\{0\}} + h[1, k] \delta_{\{1\}} + \dots + h[N-1, k] \delta_{\{N-1\}}$$

Evaluating this expression at different values of $k \in Z_N$ results in the following set of identities:

$$\begin{aligned} T_h\{\delta_{\{0\}}\} &= h[0,0] \delta_{\{0\}} + \dots + h[N-1,0] \delta_{\{N-1\}} \\ T_h\{\delta_{\{1\}}\} &= h[0,1] \delta_{\{0\}} + \dots + h[N-1,1] \delta_{\{N-1\}} \\ T_h\{\delta_{\{N-1\}}\} &= h[0, N-1] \delta_{\{0\}} + \dots + h[N-1, N-1] \delta_{\{N-1\}} \end{aligned}$$

Rewriting these identities in array form results in:

$$\begin{bmatrix} T_h\{\delta_{\{0\}}\} \\ T_h\{\delta_{\{1\}}\} \\ \vdots \\ T_h\{\delta_{\{N-1\}}\} \end{bmatrix} = \begin{bmatrix} h[0,0] & h[N-1,0] \\ h[0,1] & h[N-1,1] \\ \vdots & \vdots \\ h[0, N-1] & h[N-1, N-1] \end{bmatrix} \cdot \begin{bmatrix} \delta_{\{0\}} \\ \delta_{\{1\}} \\ \vdots \\ \delta_{\{N-1\}} \end{bmatrix}$$

Thus, given a system T_h , and a signal $f \in L(Z_N)$, a response $g = T_h\{f\}$ is obtained as follows:

$$g = T_h\{f\} = T_h \left\{ \sum_{k \in Z_N} f[k] \delta_{\{j\}} \right\}$$

The linearity condition of the LSI-FI systems over the linear space $l^2(Z_N)$ becomes useful to simplify.

Taking advantage of the linearity of T_h results in:

$$g = T_h \{f\} = \sum_{k \in Z_N} f[k] T_h \{\delta_{\{k\}}\} = \sum_{j \in Z_N} g[j] \delta_{\{j\}}$$

Thus, $g = T_h \{f\} = \sum_{j \in Z_N} g[j] \delta_{\{j\}}$, $f \in l^2(Z_N)$, or

$$g = T_h \{f\} = \sum_{j \in Z_N} \left(\sum_{k \in Z_N} f[k] h[j, k] \right) \delta_{\{j\}}, \text{ where}$$

$$g[m] = T_h \{f\} = \left(\sum_{j \in Z_N} \sum_{k \in Z_N} f[k] h[j, k] \right) \delta_{\{j\}}[m]$$

$$g[m] = \sum_{k \in Z_N} f[k] h[m, k], \quad m \in Z_N$$

This last expression in coordinates notation with respect to the standard basis results in the following expression:

$$\begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[j] \\ \vdots \\ g[N-1] \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{N-1} f[k] h[0, k] \\ \sum_{k=0}^{N-1} f[k] h[1, k] \\ \vdots \\ \sum_{k=0}^{N-1} f[k] h[j, k] \\ \vdots \\ \sum_{k=0}^{N-1} f[k] h[N-1, k] \end{bmatrix}$$

Extracting the f from the above, results in the following matrix-vector representation

$$\begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[j] \\ \vdots \\ g[N-1] \end{bmatrix} = \begin{bmatrix} h[0,0] & \cdots & h[0, N-1] \\ h[1,0] & \cdots & h[1, N-1] \\ \vdots & \ddots & \vdots \\ h[j,0] & \cdots & h[j, N-1] \\ \vdots & \vdots & \vdots \\ h[N-1,0] & \cdots & h[N-1, N-1] \end{bmatrix} f$$

Recalling that $h[j, k] = h[j - k]$, $j, k \in Z_N$, produces the following expression for the matrix-vector formulation of the cyclic convolution operation:

$$\begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[j] \\ \vdots \\ g[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & \cdots & h[1] \\ h[1] & \cdots & h[2] \\ \vdots & \ddots & \vdots \\ h[j] & \cdots & h[j+1] \\ \vdots & \vdots & \vdots \\ h[N-1] & \cdots & h[0] \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ f[j] \\ \vdots \\ f[N-1] \end{bmatrix}$$

Thus, the matrix-vector computation $g = H_N(f)$ represents the cyclic convolution operation of the vectors f and h , or simply, $g = f \otimes_N h = T_h \{f\}$. Since the vector or signal f is an arbitrary vector, a general formulation for the matrix H_N is given, where commas are used to improve readability of the expressions:

$$H_N = [T_h \{\delta_{\{0\}}\}, T_h \{\delta_{\{1\}}\}, \dots, T_h \{\delta_{\{N-1\}}\}]$$

$$H_N = [T_{\delta_{\{0\}}} \{h\}, T_{\delta_{\{1\}}} \{h\}, \dots, T_{\delta_{\{N-1\}}} \{h\}]$$

$$H_N = [I_N T_h \{\delta\}, S_N T_h \{\delta\}, \dots, S_N^{N-1} T_h \{\delta\}]$$

The cyclic convolution operation can now be formulated at the vector space or linear space level as follows:

$$g = f \otimes_N h = T_h \{f\}, f, h \in L(Z_N)$$

$$g = T_h \{f\} = T_h \left(\sum_{k=0}^{N-1} f[k] \delta_{\{k\}} \right)$$

$$T_h \{f\} = T_h \left\{ \sum_{k \in Z_N} f[k] \delta_{\{k\}} \right\}$$

$$T_h \{f\} = \sum_{k \in Z_N} f[k] T_h \{\delta_{\{k\}}\}$$

$$T_h \{f\} = \sum_{k \in Z_N} f[k] \left(\sum_{j \in Z_N} h[j - k] \delta_{\{j\}} \right)$$

$$T_h \{f\} = \sum_{j \in Z_N} \left(\sum_{k \in Z_N} h[j - k] f[k] \right) \delta_{\{j\}}$$

Evaluating $g \in l^2(Z_N)$ at $j \in Z_N$ results in

$$g[j] = T_h \{f\}[j] = \sum_{j \in Z_N} \left(\sum_{k \in Z_N} h[j - k] f[k] \right) \delta_{\{j\}}[j]$$

$$g[j] = \sum_{j \in Z_N} \left(\sum_{k \in Z_N} h[j - k] f[k] \right) \delta$$

$$g[j] = \sum_{k \in Z_N} h[j - k] f[k].$$

For the rest of this section a slight change of notation is utilized to conform a more common notation in the area of discrete signal processing. In this context, an input to an LSI-FIR filter is normally denoted with the symbol x , while the output is denoted by the symbol y . The finite sequence representing the impulse response function of the LSI-FIR filter is usually denoted by the symbol h .

Consider the N equations for cyclic convolution:

$$y[n] = \sum_{m=0}^{N-1} h[m] \cdot x[\langle n-m \rangle_N] = \sum_{m=0}^{N-1} x[m] \cdot h[\langle n-m \rangle_N],$$

$$n = 0, 1, 2, \dots, N-1$$

In matrix form these equations may be expressed as:

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} x[0] & x[N-1] & \cdots & x[1] \\ x[1] & x[0] & \cdots & x[2] \\ \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[N-2] & \cdots & x[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[N-1] \end{bmatrix}$$

Note: A matrix A is a *Toeplitz-like matrix* if the elements along the diagonals are the same. That is, if:

$$a_{ij} = a_{pq} \quad \text{for } i-p = j-q$$

If the matrix A is of size $N \times N$ and each row is the preceding row circularly rotated right, then the matrix A is termed *circulant* [12]. The matrix representation of the CC is a circulant matrix.

D. The Discrete Fourier Transform (DFT)

Given a discrete signal $x \in l^2(Z_N)$, then the discrete Fourier transform (DFT) pair is established as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W^{kn} \leftrightarrow x[n] = -\frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot W^{-nk}$$

Here, N is the number of samples in one period of $x \in l^2(Z_N)$, and $W = \exp(-\frac{j2\pi}{N})$.

A *cyclic convolution property* relates the resulting cyclic convolution signal y of two periodic discrete signals $x, y \in l^2(Z_N)$ by means of their DFT in the following manner:

$$Y[k] = H[k] \cdot X[k] \quad k = 0, 1, \dots, N-1$$

Here, $Y[k]$, $H[k]$ and $X[k]$ are the DFT's of y , h , and x , respectively. Thus, an alternative formulation of the CC is as follows:

$$y[n] = -\frac{1}{N} \sum_{k=0}^{N-1} H[k] \cdot X[k] \cdot W^{-nk} \quad n = 0, 1, 2, \dots, N-1$$

$H[k]$ and $X[k]$ can be computed in parallel and then calculate the N products $H[k] \cdot X[k]$ $k \in Z_N$.

E. Polynomial Congruences.

An n th degree polynomial $P[z]$ over some field F is formulated by the following expression:

$$P[z] = \sum_{i=0}^n a_i \cdot z^i \quad n > 0,$$

where the fixed elements $\{a_i\}$ are drawn from the field F , normally a *Galois field* $GF(p)$ or an isomorphic subset of the complex numbers. Some of the elements can be zero but a_n cannot. If a_n is the unity, then the polynomial is termed a *monic* polynomial.

The *degree* or *order* of the polynomial is denoted by $Deg[P[z]]$. Assume two polynomials $P[z]$ and $Q[z]$ meet this definition. If there exist a third polynomial $D[z]$ such that $P[z] = Q[z] \cdot D[z]$, then $D[z]$ is termed a divisor of $P[z]$ and the division is denoted by $D[z] | P[z]$. If $P[z]$ can only be divided by polynomials of degree 0 (that is numbers) or polynomials of degree n , then $P[z]$ is termed *irreducible* over the field F or *prime* [5]. The structure of the monic polynomials $p_i[z]$ is strongly dependent on the field F . The classic example is $(z^2 + 1)$ which is irreducible in the real numbers field, in the rational field, and the integer field; however, it has factors $(z + i)$ and $(z - i)$ in the complex field. The nature of field, then, plays an important role in the minimal algorithms for products of polynomials [13], as will be demonstrated in the next sections.

Associated with a N -point discrete signal or sequence $y[n]$, $n \in Z_N$, or simply $y[n]$, there exist an $(N-1)$ -th degree polynomial in the indeterminate z :

$$y(z) = y_0 + y_1 z + \dots + y_{N-1} z^{N-1}$$

If a sequence $y[n]$ is the convolution of two sequences $h[n]$ and $x[n]$, then a well-known property of z transform is expressed as follows:

$$Y[z] = H[z] \cdot X[z]$$

Where $Y[z]$, $H[z]$ and $X[z]$ are the z transform of $y[n]$, $h[n]$ and $x[n]$ respectively. Thus, efficient methods for convolving sequences are also efficient methods for multiplying two polynomials, and vice versa. Consider the general polynomial congruence:

$$Y[z] = \langle H[z] \cdot X[z] \rangle_{P[z]}$$

where $H[z]$ and $X[z]$ are two polynomials defined over the same field as $Y[z]$. Given the inequality:

$$Deg(P[z]) > Deg(H[z]) + Deg(X[z]),$$

then this describes a discrete convolution operation. If all three polynomials have degree of N and $P[z]$ is the polynomial $(z^N - 1)$, then the cyclic convolution can be represented as a multiplication of polynomials modulo a monic polynomial through the expression:

$$y(z) = \langle x(z)h(z) \rangle_{(z^N - 1)}$$

F. Winograd's Minimal Complexity Theorem.

Let F be an arbitrary field and let $\{F\}[z]$ represent the linear space of all polynomials in the indeterminate z , with degree less than N . Let $X[z], H[z] \in \{F\}[z]$ two polynomials defined over the field F . Then, the operation:

$$Y[z] = \langle H[z] \cdot X[z] \rangle_{P[z]}$$

requires at least $2N - k$ multiplications, where k is the number of irreducible factors of $P[z]$ over the field F . If $P[z]$ is prime, then k is 1. If $P[z] = (z^N - 1)$ (as in the CC), then $k = N$ and the minimal number of multiplications is $2N - k = 2N - N = N$. [14], [15].

This last number N is the number of complex multiplications obtained by the application of the proposed algorithms demonstrated in the next section.

G. The Chinese Remainder Theorem (CRT).

Optimal algorithms for one dimensional cyclic convolution have been constructed by Winograd [6] by means of the CRT. Consider the expression for polynomial multiplication modulo a polynomial:

$$y(z) = \langle x(z)h(z) \rangle_{p(z)}$$

With,

$$p(z) = \prod_{i=1}^k p_i(z)$$

The polynomial product can be reduced to the following product of polynomial of smaller degree:

$$y_i(z) = \langle x_i(z) \cdot h_i(z) \rangle_{p_i(z)},$$

and the total product can be reconstructed using the Chinese Remainder Theorem (CRT) [7] by:

$$y(z) = \sum_{i=1}^k \langle y_i(z) \cdot R_i(z) \rangle_{p(z)}$$

The polynomials $R_i(z)$ are defined as:

$$R_i(z) = 1 \text{ mod } p_i(z), \\ 0 \text{ mod } p_j(z), i \neq j$$

In [5] is demonstrated that the products $y_i(z)$ are disjoint. It is also proved that if optimal algorithms for the products $y_i(z)$ exist, then the algorithm together with the polynomial reductions modulo $p_i(z)$ and the CRT reconstruction form an overall optimal algorithm for the computation of the cyclic convolution [8].

III. ALGORITHM DEVELOPMENT

This section shows the process to obtain recursive algorithms in order to perform the polynomial product of length N modulo the polynomial $p(z) = (z^N - 1)$ when N is a power of 2. In the development a matrix representation is used to formulate a matrix-vector product and take advantage of the structure of the circulant matrix and the roots of unity. In this manner a lower bound is obtained in the number of multiplications as established by the Winograd theorem.

Let $\mathbf{y} = \mathbf{X} \cdot \mathbf{h}$, where $N = 2$, and $N - 1 = 1$ is the degree of the associated polynomials, the polynomial product module $(z^2 - 1)$ can be represented in matrix form as:

$$\mathbf{y} = \begin{bmatrix} x_0 & x_1 \\ x_1 & x_0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} x_0 h_0 + x_1 h_1 \\ x_0 h_1 + x_1 h_0 \end{bmatrix} \quad (1)$$

Straightforward computation is done by means of 4 multiplications and 2 sums. Winograd in [6], [7] shows that, for computing this matrix-vector product, the following algorithm can be used with only 2 multiplications and 6 sums:

$$\mathbf{y} = \begin{bmatrix} x_0 & x_1 \\ x_1 & x_0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 \\ m_1 - m_2 \end{bmatrix}, \text{ where:}$$

$$m_1 = \frac{1}{2}(x_0 + x_1)(h_0 + h_1); \quad m_2 = \frac{1}{2}(x_0 - x_1)(h_0 - h_1) \quad (2)$$

In the reduction of the algorithm's complexity in terms of multiplications, it is not necessary to take into consideration the multiplications by $\frac{1}{2}$, 1 and -1 . They are elements of the field of constants or ground set G , which will be in the field of complex numbers [6], [7]. The following observation can be made: the constants employed in this algorithm are the roots of unit of the polynomial $z^N - 1$ in this case $z^2 - 1$ (1 and -1), and the value $\frac{1}{2}$ is simply $\frac{1}{N}$. The relation between this algorithm for performing cyclic convolution and the approach using the DFT is now evident, since the constants used in both algorithms are the same.

The following figure shows, through a numerical example, how the first algorithm can be mapped into a parallel hardware structure:

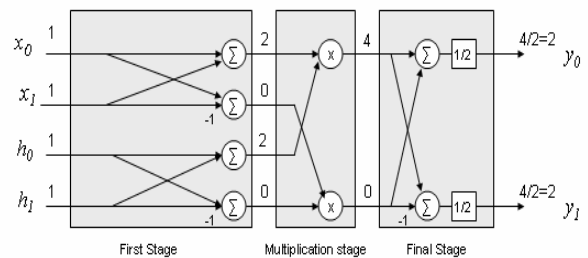


Figure 2. Flow diagram for cyclic convolution - $N=2$.

The Figure 2 above shows three stages in the algorithm for $N = 2$. The first stage can be associated with the DFT. In fact, the values of the coefficients at the output of this stage are the same that those obtained by applying the DFT. The multiplication stage can be associated with the Hadamard multiplication product of the two transformed sequences, and the last stage can be related to the inverse transform. The number of stages of our algorithm is given by the order of the sequences. In this case, for $N = 2^1$, the algorithm shows $S_1 = \log_2(N) = 1$ multiplication stages by the field of constants or ground set G (the roots of the polynomial $z^2 - 1$), and $S_2 = \log_2(N) = 1$ multiplications stages by the roots of the polynomial $z^2 - 1$. It is necessary only one stage to multiply the sequences and it is of size N , which is the limit established by the Winograd theorem. Now, increasing the order of the polynomials to the next power of two, which is $N = 2^2 = 4$, the matrix representation is then:

$$\mathbf{X} = \begin{bmatrix} x_0 & x_3 & x_2 & x_1 \\ x_1 & x_0 & x_3 & x_2 \\ x_2 & x_1 & x_0 & x_3 \\ x_3 & x_2 & x_1 & x_0 \end{bmatrix}, \text{ and } \mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}, \quad (3)$$

The matrix \mathbf{X} and the vector \mathbf{h} are represent as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 \\ \mathbf{X}_1 & \mathbf{X}_0 \end{bmatrix}, \text{ and } \mathbf{h} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \end{bmatrix}, \text{ then} \tag{4}$$

$$\mathbf{y} = \mathbf{Xh} = \begin{bmatrix} \mathbf{X}_0\mathbf{h}_0 + \mathbf{X}_1\mathbf{h}_1 \\ \mathbf{X}_0\mathbf{h}_1 + \mathbf{X}_1\mathbf{h}_0 \end{bmatrix}$$

By the use of the same algorithm employed in (2) it is possible to calculate the vector \mathbf{y} as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 \\ \mathbf{X}_1 & \mathbf{X}_0 \end{bmatrix} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 + \mathbf{m}_2 \\ \mathbf{m}_1 - \mathbf{m}_2 \end{bmatrix}, \text{ where:} \tag{5}$$

$$\mathbf{m}_1 = \frac{1}{2}(\mathbf{X}_0 + \mathbf{X}_1)(\mathbf{h}_0 + \mathbf{h}_1); \mathbf{m}_2 = \frac{1}{2}(\mathbf{X}_0 - \mathbf{X}_1)(\mathbf{h}_0 - \mathbf{h}_1)$$

The vectors \mathbf{m}_1 and \mathbf{m}_2 are found by:

Calling for \mathbf{m}_1 :

$$\begin{bmatrix} x'_0 & x'_1 \\ x'_1 & x'_0 \end{bmatrix} = \begin{bmatrix} x_0 + x_1 & x_3 + x_1 \\ x_1 + x_3 & x_0 + x_2 \end{bmatrix}, \text{ then}$$

$$\begin{bmatrix} h'_0 \\ h'_1 \end{bmatrix} = \begin{bmatrix} h_0 + h_2 \\ h_1 + h_3 \end{bmatrix}, \text{ where we have 4 sums.} \tag{6}$$

Now, the matrix - vector product to obtain \mathbf{m}_1 is :

$$\mathbf{m}_1 = \frac{1}{2} \begin{bmatrix} x'_0 & x'_2 \\ x'_1 & x'_0 \end{bmatrix} \begin{bmatrix} h'_0 \\ h'_1 \end{bmatrix}$$

It has the same properties as shown in (2); thus, it can be computed using 2 multiplications and 6 sums. For \mathbf{m}_2 , a similar procedure is followed:

$$\begin{bmatrix} x'_2 & -x'_3 \\ x'_3 & x'_2 \end{bmatrix} = \begin{bmatrix} x_0 - x_2 & x_3 - x_1 \\ x_1 - x_3 & x_0 - x_2 \end{bmatrix}, \text{ then}$$

$$\begin{bmatrix} h'_2 \\ h'_3 \end{bmatrix} = \begin{bmatrix} h_0 - h_2 \\ h_1 - h_3 \end{bmatrix}, \text{ There are 4 additional sums.} \tag{7}$$

Now, the matrix - vector product to obtain \mathbf{m}_2 :

$$\mathbf{m}_2 = \frac{1}{2} \begin{bmatrix} x'_2 & -x'_3 \\ x'_3 & x'_2 \end{bmatrix} \begin{bmatrix} h'_2 \\ h'_3 \end{bmatrix}$$

Here, a different structure appears; however, it is closely related to block circulants. In order to solve this matrix-vector product, a new algorithm suggested by S. Winograd is employed:

$$\mathbf{m}_2 = \frac{1}{2} \begin{bmatrix} x'_2 & -x'_3 \\ x'_3 & x'_2 \end{bmatrix} \begin{bmatrix} h'_2 \\ h'_3 \end{bmatrix}; \mathbf{m}_2 = \frac{1}{2} \begin{bmatrix} (r_1 + r_2) \\ (r_1 - r_2)/i \end{bmatrix}, \text{ where} \tag{8}$$

$$r_1 = \frac{1}{2}(x'_2 + ix'_3)(h'_2 + ih'_3); r_2 = \frac{1}{2}(x'_2 - ix'_3)(h'_2 - ih'_3)$$

The computations of the matrix sums $(\mathbf{X}_0 + \mathbf{X}_1)$, and $(\mathbf{X}_0 - \mathbf{X}_1)$ are done using only two sums. The same occurs with the computation of vectors sums $(\mathbf{h}_0 + \mathbf{h}_1)$ and $(\mathbf{h}_0 - \mathbf{h}_1)$. Also is evident the multiplication by the roots of unit $1, -1, i,$ and $-i$. The vectors sums to calculate the general results are then:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 + \mathbf{m}_2 \\ \mathbf{m}_1 - \mathbf{m}_2 \end{bmatrix}, \text{ calling:}$$

$$\mathbf{m}_1 = \begin{bmatrix} x''_0 \\ x''_1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x'_0 & x'_1 \\ x'_1 & x'_0 \end{bmatrix} \begin{bmatrix} h'_0 \\ h'_1 \end{bmatrix};$$

$$\mathbf{m}_2 = \begin{bmatrix} x''_2 \\ x''_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x'_2 & -x'_3 \\ x'_3 & x'_2 \end{bmatrix} \begin{bmatrix} h'_2 \\ h'_3 \end{bmatrix}. \text{ Then,} \tag{9}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 + \mathbf{m}_2 \\ \mathbf{m}_1 - \mathbf{m}_2 \end{bmatrix} = \begin{bmatrix} x''_0 + x''_2 \\ x''_1 + x''_3 \\ x''_0 - x''_2 \\ x''_1 - x''_3 \end{bmatrix}$$

is computed employing 4 additional sums. The multiplication by the constant $\frac{1}{2}$ was realized twice. It is possible to change these constants by the constant $\frac{1}{4}$ or in other words $\frac{1}{N}$. The other constants employed are the roots of $z^4 - 1$. These outcomes can be observed with a numerical example depicted in the **Figure 3**.

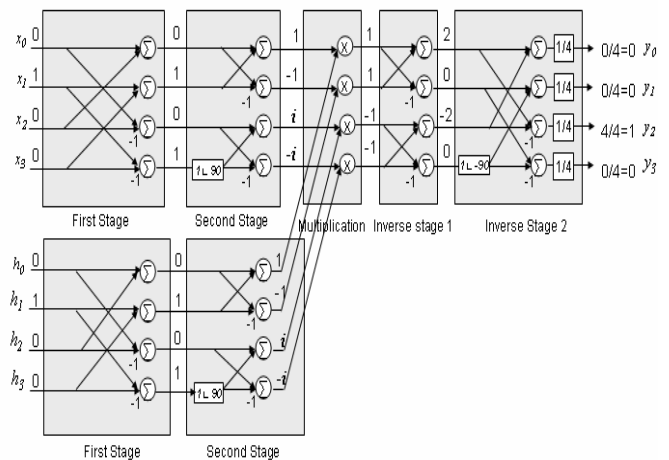


Figure 3. Flow diagram for cyclic convolution - $N=4$.

It is possible to appreciate in **Figure 3** the stages of the algorithm. For $N = 2^2$, the algorithm shows $S_1 = \log_2 N = 2$ stages of multiplication by the roots of the polynomial $z^4 - 1$. We can observe also $S_2 = \log_2 N = 2$ stages that are similar to the process for the inverse Fourier transform, and one multiplication stage of the “transformed” sequences of size $N = 4$. The total numbers of operation are 4 multiplications and 24 sums. A similar process to the one described above is done with the roots of the polynomial $z^N - 1$, for convolutions of order $N = 2^s$. Since the generalization is straightforward there is no need to follow with a detailed description.

The general algorithm uses a field of constants that are organized according to its use in the mapping of the convolution process. **Figure 4** below shows an easy manner to find constants for different order sequences.

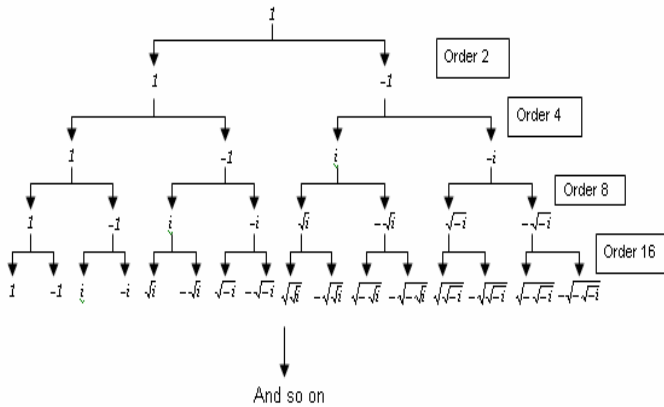


Figure 4. Roots of unit for different order sequences.

Now, the mathematical foundation for the algorithm is described by means of a general example; consider $y(z) = \langle x(z)h(z) \rangle_{(z^4-1)}$ for the polynomials:

$$\begin{aligned}
 x(z) &= x_0 + x_1z + x_2z^2 + x_3z^3 \\
 h(z) &= h_0 + h_1z + h_2z^2 + h_3z^3 \\
 y(z) &= x_0h_0 + x_3h_1 + x_2h_2 + x_1h_3 + \\
 &\quad (x_1h_0 + x_0h_1 + x_3h_2 + x_2h_3)z + \\
 &\quad (x_2h_0 + x_1h_1 + x_0h_2 + x_3h_3)z^2 + \\
 &\quad (x_3h_0 + x_2h_1 + x_1h_2 + x_0h_3)z^3
 \end{aligned}$$

The above product can be computed, by direct computation, using 16 multiplications and 12 sums. By means of the Chinese remainder theorem this polynomial product can be realized following three steps:

1) The two polynomials are divided way long division by the roots of the monic polynomial $(z^4 - 1)$ obtaining the following remainders:

$$\begin{aligned}
 &(x_0 + x_1 + x_2 + x_3) \text{ and } (h_0 + h_1 + h_2 + h_3) \text{ for } (z - 1) \\
 &(x_0 - x_1 + x_2 - x_3) \text{ and } (h_0 - h_1 + h_2 - h_3) \text{ for } (z + 1) \\
 &(x_0 + ix_1 - x_2 - ix_3) \text{ and } (h_0 + ih_1 - h_2 - ih_3) \text{ for } (z - i) \\
 &(x_0 - ix_1 - x_2 + ix_3) \text{ and } (h_0 - ih_1 - h_2 + ih_3) \text{ for } (z + i)
 \end{aligned}$$

2) Now, by multiplying those remainders we can obtain:

$$\begin{aligned}
 m_1 &= (x_0 + x_1 + x_2 + x_3) (h_0 + h_1 + h_2 + h_3) / 2 \\
 m_2 &= (x_0 - x_1 + x_2 - x_3) (h_0 - h_1 + h_2 - h_3) / 2 \\
 m_3 &= (x_0 + ix_1 - x_2 - ix_3) (h_0 + ih_1 - h_2 - ih_3) / 2 \\
 m_4 &= (x_0 - ix_1 - x_2 + ix_3) (h_0 - ih_1 - h_2 + ih_3) / 2
 \end{aligned}$$

3) By means of a linear combination we obtain:

$$\begin{aligned}
 y_0 &= (m_1 + m_2 + m_3 + m_4) / 2 \\
 y_1 &= ((m_1 - m_2) + (m_3 - m_4) / i) / 2 \\
 y_2 &= ((m_1 + m_2) - (m_3 + m_4)) / 2 \\
 y_3 &= ((m_1 - m_2) - (m_3 - m_4) / i) / 2
 \end{aligned}$$

The great advantage of the matrix representation proposed in this paper is that it is not necessary to

calculate the polynomials remainders. Other minimal multiplicative complexity algorithms based on CRT such as Winograd's algorithms need to realize this decomposition. For this reason, the whole algorithm complexity is reduced. It is very interesting to compare the new algorithm with those that use the DFT to compute cyclic convolution. The pre-Hadamard stages are compared in **Figure 4** and **Figure 5** for $N = 8$.

A comparison between these figures shows a high correlation in the mapping of the two algorithms into a signal flow diagram. Even though their signal flow diagrams are slightly different, the two figures show the same number of computations ($O(N \cdot \log(N))$). An advantage of the present algorithm, is that it does not need a stage of bit reversal operation. For this reason, this will improve the time necessary to realize the convolution operation with respect to the radix 2 and radix 4 FFT approaches. This is due to the fact that they need to do twice this process (at the beginning and in the post Hadamard multiplication) in order to obtain the output data in the required organized manner.

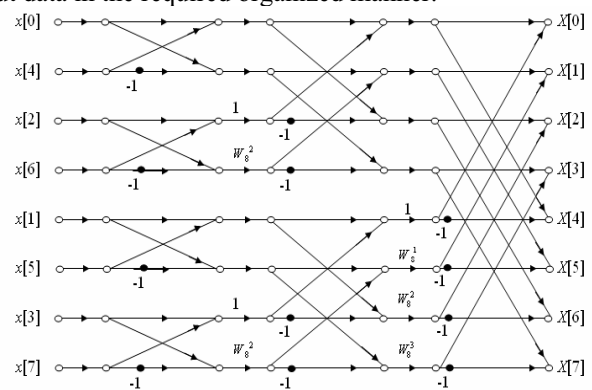


Figure 5. Signal flow diagram for FFT-2 order $N=8$.

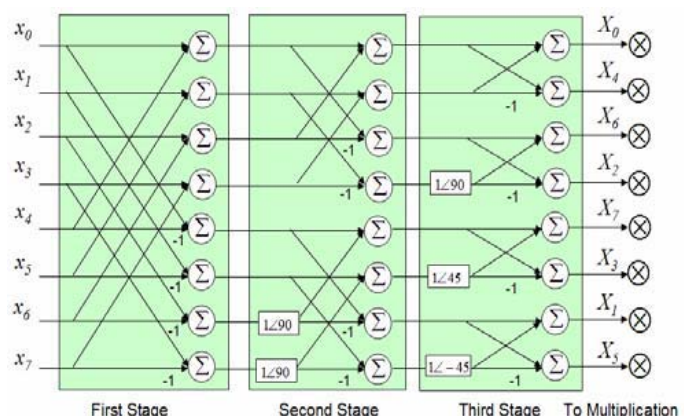


Figure 6. Signal flow diagram for the new algorithm order $N=8$.

IV. CONCLUSION

This document presents a novel algorithm for the fast and efficient computation of the CC with minimal mathematical complexity, based on the product of a circulant matrix by a vector, and the use of the CRT.

The principal goal was to obtain a recursive algorithm, easy to implement, with the advantage of not needing to realize the polynomial divisions by the roots of unity in order to obtain less number of multiplications. The algorithm was also compared with the algorithm that uses the Fourier transform approach and the present algorithm has the comparative advantage that it does not require to realize the bit reversal operation in order to exhibit an in-place computation.

This work changes the conceptual framework of the computation of the CC using the FFT and locates it in a structure of minimum mathematical complexity. The algorithm obtained is limited to polynomials with length a power of 2, and its flow diagram suggests the possible use of Kronecker or Tensor products as a tool to improve their performance by means of parallel processing.

Future work includes the use of the methodology employed here for the problem of 2 dimensional cyclic convolution operations, and the mapping to parallel hardware structure by use of VHDL and low power complex multiplication operations.

ACKNOWLEDGMENT

The authors would like to thank Dr. Shmuel Winograd for his helpful suggestions and comments during the design and development stages of this work.

REFERENCES

- [1] H. Krishna, B. Krishna, K. Y. Lin, J. D. Sun, *Computational Number Theory and Digital Signal Processing* (CRC Boca Raton, Florida, 1994).
- [2] Abraham H. Díaz-Pérez, *Análisis y Diseño de Algoritmos Para la Computación con Estructuras Circulantes*, MS Thesis, ECE Dept., UPRM, Mayagüez, Puerto Rico, May 2004.
- [3] D. Chiper, M.N.S. Swamy, M. Omair Ahmad, "An Efficient Unified Framework for Implementation of a Prime-Length DCT/IDCT with High Throughput," *IEEE Trans. on Signal Processing*, vol. 55, no. 6, June 2007.
- [4] C. Cheng, K. K. Parhi, "Low-Cost Fast VLSI Algorithm for Discrete Fourier Transform," *IEEE Trans. on Circuits and Systems*, vol. 54, no. 4, April 2007.
- [5] C. Cheng, K. K. Parhi, "Hardware Efficient Fast DCT Based on Novel Cyclic Convolution Structures," *IEEE Trans. On Signal Processing*, vol. 54, no. 11, Nov. 2006.
- [6] Y. Guo, J. Zhang, D. McCain, J. R. Cavallaro, "Structured Parallel Architecture for Displacement MIMO Kalman Equalizer in CDMA Systems," *IEEE Trans. on Circuits and Systems*, vol. 54, no. 2, Feb. 2007.
- [7] H. Zhang, M. Xia, G. Hu, "A Multiwindow Partial Buffering Scheme for FPGA-Based 2-D Convolvers," *IEEE Trans. on Circuits and Systems*, vol. 54, Feb. 2007.
- [8] D. Zhu, Z. Zhu, "Range Resampling in the Polar Format Algorithm for Spotlight SAR Image Formation Using the Chirp z-Transform," *IEEE Trans. on Signal Processing*, vol. 55, no. 3, March 2007.
- [9] D. G. Myers, *Efficient Convolution and Fourier Transform Techniques* (Prentice Hall, Australia, 1990).
- [10] K. Hoffman, R. Kunze, *Linear Algebra*, (Prentice Hall, Inc., New Jersey, 1961).
- [11] D. Rodriguez, *Computational Signal Processing and Sensor Array Signal Algebra: A Representation Development Approach*, Book Draft, (UPRM, PR, 2002).
- [12] J. Davis, *Circulants Matrices* (John Wiley, New York, 1979).
- [13] J. McClellan and C. Rader, *Number Theory in Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall 1979.
- [14] S. Winograd, *Arithmetic complexity of computations* (Society for Industrial and Applied Mathematics, 1980).
- [15] M. Heideman, *Multiplicative complexity, convolution, and the DFT* (Springer Verlag, New York 1988)
- [16] J. Cooley, *Some applications of computational complexity Theory to Digital Signal Processing*. 1981 Joint Automatic Contr. Conf. University of Virginia, June 17-19 1981.

Abraham H. Diaz-Perez was born in Barranquilla-Colombia. He received the B.S. in Electrical Engineer degree from the Technologic University of Bolivar at Cartagena-Colombia in 1997 and the M.Sc. in Electrical Engineer from the University of Puerto Rico at Mayaguez-Puerto Rico in 2004. He is currently working in the Popular University of the Cesar at Valledupar-Colombia, the work fields cover fast algorithms for multidimensional cyclic convolution, adaptive filter theory and acoustic and image processing applications. Prof. Diaz-Perez is an international judge for different IEEE conferences as MWCASS and IASTED, and he is recipient of different grants as the PR Louis Stroke Alliance for Minorities Participation and Colciencias.

Domingo Rodriguez received his doctoral degree in Engineering from the City University of New York in 1988. He conducted post-doctoral work at the CUNY Center for Large Scale Computation in New York and at the Bell Laboratories in New Jersey. Since 1988 he has been with the Department of Electrical and Computer Engineering of the University of Puerto Rico at Mayaguez, where he is currently a professor of Communications and Signal Processing and the Director of the Institute for Computing and Informatics Studies.