Signal Algebra Operators for SAR PSF Hardware Computations *

Domingo Rodriguez
Electrical and Computer Engineering Department
University of Puerto Rico at Mayaguez
Mayaguez, PR 00681-9042
E-mail:domingo@ece.uprm.edu

Abstract

Advances in information technologies are narrowing the gap that traditionally existed between computing methods for radar signal processing, digital communications, and digital signal processing in general. A new field is becoming more stable as new applications and challenges are met for on board, real time operations. This field is termed computing methods for radar communications signal processing, and it deals primarily with the analysis, design, and implementation of digital signal processing computing methods radar signal communications operations. Special attention is being given in this area to efficient implementation of signal operators for object domain to spectral domain transformations. This endeavor is somewhat straightforward when treating small-size one-dimensional signals. In dealing with multidimensional signals, we encounter a great many difficulties. These difficulties are compounded as the signals increase in size and dimension. New computing methods frameworks are sought in order to aid multidimensional digital signal processing and enhance radar signal communications implementation efforts. The work presented here deals with basic signal algebra concepts in the formulation of Kronecker Fourier factors as signal operators for discrete cross-ambiguity function processing in SAR point spread function (PSF) modeling.

*This work was supported in part by NSF Grant 9977071.

1 Introduction

This work deals with the fundamental issue of the fast and efficient treatment of microwave remote sensed data in order to extract information important to a surveillance user. Great advances in active sensor technology, communications, and signal processing technology are demanding new computational theories, methods, and techniques to improve our rapid awareness of our physical sensory reality. For the particular case of SAR systems, this implies fast and efficient means for image formation and rendering from raw data. The identification of enhanced raw data generation techniques will certainly contribute to improve at SAR image formation processes. The work presented here concentrates on the formulation Kronecker Fourier factors as signal operators for the algebraic modeling point target response functions using point estimates of discrete cross-ambiguity surface computations. Our work centers on the notion that enhancements and understanding of SAR point surface response functions processing greatly depends on the efficient computation of finite discrete cross-ambiguity functions. We, in turn, compute the cross-ambiguity function using two-dimensional discrete Fourier transform (DFT), cyclic shift, index reversal, modulation, and other signal operators. For the work presented here we use Kronecker products notation as a language to aid in the mapping of Fourier operators to DSP microprocessor units and to identify similarities and differences between commonly known formulations of two-dimensional fast Fourier transform (FFT) algorithms as compositions of basic functional expressions.

There exist many formulations of fast algorithms for computing the discrete Fourier transform (DFT). Kronecker array signal (KAS) algebra, a branch of finite dimensional multilinear algebra, has been used successfully as a language to identify similarities and differences among various fast Fourier transform (FFT) algorithm variants as well as for the creation of new variants. Each multidimensional DFT computation is expressed in matrix form. The multidimensional DFT matrix, in turn, is decomposed into a set of factors, called *functional primitives*, which are individually mapped to the underlying computing structure. It is in this mapping process where the language of KAS algebra becomes instrumental. For a given computing structure and multidimensional DFT matrix, there are many FFT algorithm variants which can map to this structure. The language of KAS algebra aids in this mapping effort by identifying the more computational efficient FFT variants and thus reducing the hardware computing effort.

2 Kronecker Signal Algebra

We introduce in this section some mathematical concepts which are useful in describing the work. First, the concept of tensor or Kronecker product of two matrices and then some basic ideas of Kronecker array signal (KAS) algebra, a branch of finite dimensional multilinear algebra. Let A and B be any two matrices. The Kronecker product of A and B is given by $A \otimes B = [a_{k\ell} \cdot B]_{k,\ell \in Z/R}, Z/R = \{0, 1, \cdots, R-1\}$. Here, we have assumed A to be a square matrix of order R. If B is also a square matrix of order, say, S, then the order of $A \otimes B$ is $R \cdot S = N$. Let A and C be $R \times R$ matrices, and B and D be $S \times S$ matrices. Next, form the Kronecker products $A \otimes B$ and $C \otimes D$. Through direct matrix multiplication we can show that $(A \otimes B)(C \otimes D) = AC \otimes BD$. If we denote I_R, I_S as identity matrices of order R and S, respectively, we have $(A \otimes B) = (A \otimes I_S)(I_R \otimes B)$. From this expression we can see that the action of computing with the matrix $(A \otimes B)$ can be performed in two stages: An action for the computation of $(I_R \otimes B)$, followed by an action for the computation of $(A \otimes I_S)$. Let the N-point discrete Fourier transform (DFT) of a one-dimensional discrete, complex, array signal $x[n]$, of length N, be defined by $(\widehat{x})[k] = \sum_{n \in Z/N} x[n]\omega_N^{kn}; \quad k \in Z/N$, where $w_N = e^{-j\frac{2\pi}{N}}$, and $j = \sqrt{-1}$. Written in matrix form, we have $(\widehat{x}) = F_N \cdot x$, $F_N = [\omega_N^{kn}]_{k,n \in Z/N}$. We call F_N a matrix representation of the DFT operator. In the same manner, the two-dimensional discrete Fourier transform of an $N_1 \times N_2$ discrete complex array signal $x[n_1, n_2]$ is defined by

$$(\widehat{x})[k_1, k_2] = \sum_{n_1 \in Z/N_1} \sum_{n_2 \in Z/N_2} x[n_1, n_2]\omega_{N_1}^{k_1 n_1}\omega_{N_2}^{k_2 n_2}; \quad k_1 \in Z/K_1,\ k_2 \in Z/K_2.$$

Also, $\omega_{N_1} = e^{-j\frac{2\pi}{N_1}}$ and $\omega_{N_2} = e^{-j\frac{2\pi}{N_2}}$. Let $F_{N_1 \times N_2}$ denote a matrix representation of the two-dimensional discrete Fourier transform operator acting on an $N_1 \times N_2$ complex signal array $x[n_1, n_2]$. Through direct matrix multiplication we can show that

$$F_{N_1 \otimes N_2} = (F_{N_1} \otimes F_{N_2}) = (I_{N_1} \otimes F_{N_2})(F_{N_1} \otimes I_{N_2})$$

$$F_{N_1 \otimes N_2} = (F_{N_1} \otimes F_{N_2}) = (F_{N_1} \otimes I_{N_2})(I_{N_1} \otimes F_{N_2})$$

If U_1, U_2, V_1, and V_2 are linear spaces over the complex field $\mathbb{C}$, and $\mathcal{T}_i : U_i \to V_i$, $i = 1, 2$, are linear operators acting over these spaces; then, $(\mathcal{T}_1 \otimes \mathcal{T}_2) : U_1 \otimes U_2 \to V_1 \otimes V_2$, termed the Kronecker product of the transformations $\mathcal{T}_1$ and $\mathcal{T}_2$, is the linear transformation satisfying the following condition:

$$(\mathcal{T}_1 \otimes \mathcal{T}_2)\{u_1 \otimes u_2\} = \mathcal{T}_1\{u_1\} \otimes \mathcal{T}_2\{u_2\}$$

for all $u_i \in U_i$, $i = 1, 2$. $T_1 \otimes T_2$ is a Kronecker product of matrices $A \otimes B$; where, A and B are the matrix representations of the operators $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, conditioned on bases selection criteria[3]. We call the elements of the linear spaces by the names of vector array signals, array signals, or, simply, signals. The linear spaces are turned into linear algebras by introducing a vector array signal binary multiplication operation invoked by the circular or cyclic convolution. Thus, we are interested in the linear spaces of the form $V = M_{R,S}(\mathbb{C})$, as well as linear $\mathcal{T}$, such that

$$\mathcal{T} : M_{R,S}(\mathbb{C}) \to M_{R,S}(\mathbb{C}),$$

In general, we can say that if we have $V = M_{R,S}(\mathbb{C})$; then, the linear space $M_{R,S}(\mathbb{C})$ can then be represented as the Kronecker product

$$M_{R,S}(\mathbb{C}) = M_{R,1}(\mathbb{C}) \otimes M_{S,1}(\mathbb{C})$$

in which the Kronecker mapping is the dyad mapping $x \otimes y = xy^T$. The signal algebras manifest themselves when we invoke multidimensional array cyclic convolutions as array binary multiplication operations.

Let $Z_N = Z/N = \{0, 1, 2, \cdots, N-1\}$. A one-dimensional array signal x, of length N, is said to be periodic, modulo R if R is a divisor of N; that is, $N = R \cdot S$ and $x[a + bR] = x[a], a \in Z_R, b \in Z_S$. A one-dimensional array signal x, of length N, is decimated modulo R if R is a divisor of N; that is, $N = R \cdot S$ and $x[a] = 0, x[a + b \cdot R] = x[a] = 0, a \in Z_R; b \in Z_S$. These observations are very important when considering the additive group theoretic properties (coset decompositions) of the input/output indexing sets of the matrix representation of *unitary* operators $\mathcal{T}$ in linear algebras V. The matrix representation of the operators can be decomposed into a set of factors which we term *Kronecker functional primitives* and are, basically, sparse matrices. This decomposition process usually leads to efficient algorithms for the action of operators. Of particular importance to us is the ubiquitous discrete Fourier transform operator.

2.1 Operators on $L(Z/N)$

The set of all one-dimensional array signals $f : Z/N \to \mathbb{C}$ forms a linear space which we denote by $L(Z/N)$. The set $L(Z/N)$ is isomorphic to the N-dimensional complex linear space $\mathbb{C}^N$. The set of N, N-point array signals $\left\{\delta_{\{k\}} : k = 0, 1, \ldots, n-1\right\}$, where $\delta_{\{k\}}[j] = 1, k = j$ forms a basis for the space $L(Z/N)$ which we call the standard basis. We introduce the shift operator S_N over the space $L(Z/N)$. This operator is the central component in the characterization of shift-invariant, finite impulse response (FIR) operators commonly used for filtering operations. Let the operator S_N over the space $L(Z/N)$ be defined in the following manner. $S_N : L(Z/N) \longrightarrow L(Z/N)$, where $\delta_{\{k\}} \longmapsto S_N\delta_{\{k\}} = \delta_{\{k+1\}}$. Using $\langle f, \delta_{\{k\}}\rangle = f[k] = f_k$ as an orthogonal projection operation, we write $f = \sum_{0\leq k<N}\langle f, \delta_{\{k\}}\rangle\delta_{\{k\}}$. Allowing F_N to operate on f gives $F_N(f) \equiv \widehat{f} = F_N(\sum_{j\in Z/N} f_j\delta_{\{j\}})$. After linearity, $\sum_{j\in Z/N} f_jF_N\delta_{\{j\}} = \sum_{j\in Z/N} f_j\chi_j^*$. To characterize cyclic, finite impulse response (FIR) operators, we start by identifying the vector array signal obtained by letting the FIR operator T_h act on the unit sample array signal δ. Since any N-th order vector array signal f can be written as a linear combination of shifted versions of δ, knowing the response $T_h(\delta)$ will help in determining $T_h(f)$. We call the unit sample response or impulse response of the system T_h the result obtained by applying T_h to the unit sample sequence δ, which sometimes is called the impulse signal. Thus, we have $T_h(\delta) = \sum_{0\leq m<M} h[m]S_M^m\delta_{\{0\}}$ or $\sum_{0\leq m<M} h[m]\delta_{\{m\}} = h$. The unit sample response of an FIR operator T_h is the array signal h. For any given vector signal $f \in L(Z/M)$, we can always write $f = \sum_{k\in Z/M} f[k]S_M^k\delta$ or $\sum_{k\in Z/M} f[k]\delta_{\{k\}}$. Evaluating f at $j \in Z/M$ results in $f[j] = \sum_{k\in Z/M} f[k]\delta_{\{k\}}[j]$ or

$\sum_{k\in Z/M} f[k]\delta[j-k]$. The indexing set $A = Z/N = \{0, 1, \ldots, N-1\}$ forms an abelian group with modulo N addition as the internal binary operation. Its dual is $\widehat{A} = \{^N\chi_{\{k\}} : k \in Z/N\}$, with $\chi_{\{k\}} : Z/N \longrightarrow \mathbb{C}$, with $[m] \longmapsto {}^N\chi_{\{k\}}[m] = e^{+2\pi jk\cdot(m)/N}$, $j = \sqrt[2]{-1}$. When no ambiguities arise, we drop the superscript N from the expression ${}^N\chi_{\{k\}}$. The value ${}^N\chi_{\{1\}}[1]$ is usually written as $\omega_N = e^{-2\pi j/N}, j = \sqrt[2]{-1}$. The functions $\chi_{\{k\}}$ are usually termed exponential sequences, characteristic sequences, or, simply, characters. Given an N-point impulse response signal, $h \in L(Z/N)$, and an input vector array signal, $x = \chi_{\{k\}}$, the output y, after acting with T_h, becomes $y = \sum_{j\in Z/N} h[j]S_N^j\chi_{\{k\}}^*$ or $T_h\{\chi_k^*\}$. Another important operator is the cyclic reflection operator, denoted by the symbol R_N. Its action on the linear space $L(Z/N)$ is described by $R_N : L(Z/N) \longrightarrow L(Z/N)$, with $(f) \longmapsto R_Nf = f^{(-)}$. Here,$(R_Nf)[k] = f^{(-)}[k]$ or f_{N-k}, Modulo N, and $k \in Z/N$.

3 Point Surface Response

Kronecker array signal (KAS) algebra has been instrumental in the analysis, design and implementation of different classes of algorithms for signal processing computing methods. In this work we concentrated on the design of variants of algorithms for the computation of the finite, discrete, radar cross-ambiguity functions, and their software and hardware realizations. The algorithms implementation methodology is an improvement over existing formulations. Enhancements on the methodology concentrate on group theoretic techniques applied to input/output data indexing sets in a KAS algebra and linear operator setting, on modified re-sampling techniques, and on the efficient computation of two-dimensional fast Fourier transforms. The algorithms have been tested in MATLAB and are currently being ported to TI 6711 DSP computing units. As it was point out above, multidimensional FFT algorithms can be expressed as Kronecker products of lower dimensional FFT's. We used this approach throughout this work. We proceed to define the basic formulation for the finite, discrete, radar cross-ambiguity used throughout this work. Let f, g be functions on $L\left(Z/N\right)$, the linear, complex space of all N-point, one-dimensional, vector array signals. The finite, discrete, radar cross-ambiguity function $A\left(f,g\right)\left[a,b\right]$ is defined on the cartesian product indexing set $Z/N \times Z/N$ as $A\left(f,g\right)\left[m,k\right] = \sum\limits_{n\in Z/n} f[n]\cdot g^{*}[n+m]e^{-j\frac{2\pi}{N}\cdot k\cdot n}$.We use $g_{m,k}[n] = g[n+m]e^{j\frac{2\pi}{N}\cdot k\cdot n}$ to write $A\left(f,g_{m,k}\right)\left[p,q\right] = e^{-j\frac{2\pi}{N}\cdot k\cdot n}A\left(f,g\right)\left[m+p,k+q\right]$. This expression introduces the study of various additive group theoretic techniques on the input output indexing sets of the computation as well as time-frequency analyses. We proceed in the next sections with the introduction of Fourier expressions as factors in the composition of Fourier transforms so essential for cross-ambiguity function processing.

3.1 $(I_R \otimes F_S)$: Parallel Fourier Factor Operation

The action of this functional primitive on the input sequence x is as follows:

$$\left(I_R \otimes F_S\right)x = \left(\underbrace{\begin{bmatrix} 1\,0\cdots 0\\ 0\,1\cdots 0\\ \vdots\,\vdots\,\ddots\,\vdots\\ 0\,0\cdots 1\end{bmatrix}}_{R-times} \otimes F_S\right)x$$

The Kronecker product $(I_R \otimes F_S)$ is obtained multiplying each element of the identity matrix I_R by the Fourier matrix F_S. Thus, it has the following

$$\left(I_R \otimes F_S\right)x = \underbrace{\begin{bmatrix} F_S & 0 & \cdots & 0\\ 0 & F_S & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & F_S\end{bmatrix}}_{R-times} x$$

From the structure of the operation $(I_R \otimes F_S)\, x$ can be observed that each F_S will act on an S-point segment of the input sequence x. That is, if x is divided in R segments of length S, the operation $(I_R \otimes F_S)\, x$ can be seen as follows:

$$(I_R \otimes F_S)\, x = \underbrace{\begin{bmatrix} F_S & 0 & \cdots & 0 \\ 0 & F_S & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_S \end{bmatrix}}_{R-times} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{R-1} \end{bmatrix}$$

$$\text{where } x_0 = \begin{bmatrix} x\,[0] \\ x\,[1] \\ \vdots \\ x\,[S-1] \end{bmatrix}, \; \cdots, \; x_{R-1} = \begin{bmatrix} x\,[(R-1)\,S] \\ x\,[(R-1)\,S+1] \\ \vdots \\ x\,[RS-1] \end{bmatrix}$$

Also, it can be observed that each F_S acts on an independent segment of data of length S. This means that if it has available a parallel computing unit with at least R processing units, the operation $(I_R \otimes F_S)\, x$ could be performed in parallel fashion, where each processing unit would execute the operation $F_S x_i$, $i \in Z_R$.

3.2 $(F_R \otimes I_S)$: Vector Fourier Factor Operation

The action of this functional primitive on the input sequence x is as follows:

$$(F_R \otimes I_S)\, x = \left(\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_S & \omega_S^2 & \cdots & \omega_S^{S-1} \\ 1 & \omega_S^2 & \omega_S^4 & \cdots & \omega_S^{2(S-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_S^{S-1} & \omega_S^{2(S-1)} & \cdots & \omega_S^{(S-1)(S-1)} \end{bmatrix} \otimes I_S \right) x$$

The Kronecker product $(F_R \otimes I_S)$ is obtained multiplying each element of the Fourier matrix F_R by the identity matrix I_S. That is,

$$(F_R \otimes I_S)\, x = \begin{bmatrix} I_S & I_S & I_S & \cdots & I_S \\ I_S & \omega_S I_S & \omega_S^2 I_S & \cdots & \omega_S^{S-1} I_S \\ I_S & \omega_S^2 I_S & \omega_S^4 I_S & \cdots & \omega_S^{2(S-1)} I_S \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_S & \omega_S^{S-1} I_S & \omega_S^{2(S-1)} I_S & \cdots & \omega_S^{(S-1)(S-1)} I_S \end{bmatrix} x$$

From the structure of the operation $(F_R \otimes I_S)\, x$ can be observed that each element of $(F_R \otimes I_S)$ will act on an S-point segment of the input sequence x. That is, if x is divided in R segments of length S, the operation $(F_R \otimes I_S)\,x$ can be seen as follows:

$$(F_R \otimes I_S)\,x = \begin{bmatrix} I_S & I_S & I_S & \cdots & I_S \\ I_S & \omega_S I_S & \omega_S^2 I_S & \cdots & \omega_S^{S-1} I_S \\ I_S & \omega_S^2 I_S & \omega_S^4 I_S & \cdots & \omega_S^{2(S-1)} I_S \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_S & \omega_S^{S-1} I_S & \omega_S^{2(S-1)} I_S & \cdots & \omega_S^{(S-1)(S-1)} I_S \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{R-1} \end{bmatrix}$$

$$\text{where } x_0 = \begin{bmatrix} x\,[0] \\ x\,[1] \\ \vdots \\ x\,[S-1] \end{bmatrix},\ \cdots,\ x_{R-1} = \begin{bmatrix} x\,[(R-1)\,S] \\ x\,[(R-1)\,S+1] \\ \vdots \\ x\,[RS-1] \end{bmatrix}$$

Each element of $(F_R \otimes I_S)$ acts on a segment of data of length S. This means that if it has available a computing unit with vector processor architecture with at least R vector registers, whose vector length is at least S, the operation $(F_R \otimes I_S)\,x$ could be performed in a vector format.

3.3 $(I_M \otimes (F_R \otimes I_Q))$: Mixed Parallel-Vector Fourier Factor Operation

This operation can be seen as follows:

$$(I_M \otimes (F_R \otimes I_Q))\,x = \left(\underbrace{\begin{bmatrix} 1\,0 \cdots 0 \\ 0\,1 \cdots 0 \\ \vdots\,\vdots \ddots \vdots \\ 0\,0 \cdots 1 \end{bmatrix}}_{M-times} \otimes (F_R \otimes I_Q) \right) x$$

The Kronecker product $(I_M \otimes (F_R \otimes I_Q))$ is obtained multiplying each element of I_M by $(F_R \otimes I_Q)$. Thus, the operation $(I_M \otimes (F_R \otimes I_Q))\,x$ can be expressed as follows:

$$(I_M \otimes (F_R \otimes I_Q))\,x = \underbrace{\begin{bmatrix} (F_R \otimes I_Q) & 0 & \cdots & 0 \\ 0 & (F_R \otimes I_Q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (F_R \otimes I_Q) \end{bmatrix}}_{M-times} x$$

From the structure of the operation $(I_M \otimes (F_R \otimes I_Q))\,x$ can be observed that each element $(F_R \otimes I_Q)$ will act on a $R \cdot Q$-point segment of the input sequence. That is, if x is divided in M segments of length $R \cdot Q$, the operation $(I_M \otimes (F_R \otimes I_Q))\, x$ can be seen as follows:

$$\underbrace{\begin{bmatrix} (F_R \otimes I_Q) & 0 & \cdots & 0 \\ 0 & (F_R \otimes I_Q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (F_R \otimes I_Q) \end{bmatrix}}_{M-times} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{M-1} \end{bmatrix} x$$

$$\text{where } x_0 = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[RQ-1] \end{bmatrix},\ \cdots,\ x_{M-1} = \begin{bmatrix} x[(M-1)\,RQ] \\ x[(M-1)\,RQ+1] \\ \vdots \\ x[MRQ-1] \end{bmatrix}$$

Each $(F_R \otimes I_Q)$ acts on an independent segment of data of length $R \cdot Q$. Therefore, if it has available a parallel computing unit with at least M processing units, the operation $(I_M \otimes (F_R \otimes I_Q))\, x$ could be performed in parallel way, where each processing unit would execute the operation $(F_R \otimes I_Q)\, x_i$, $i \in Z_M$.

4 Summary and Conclusions

Kronecker-core array signal algebra, a branch of finite dimensional multilinear algebra, was utilized as a mathematical tool-language for formulations of multidimensional fast Fourier transform (FFT) algorithms, prevalent in all cross-ambiguity functions as well as multidimensional correlation computations. An interactive Java-based stand-alone utility was designed and developed to assist in this work through automatic software source code generation of FFT algorithms from Kronecker algebra formulations.

This alternative modality of using Kronecker algebra for mapping multidimensional FFT's to advanced hardware computing structures is showing promising results for allowing to establish identifications between parallel-distributed computing constructs and the mathematical expressions named by us *functional primitives*. Algorithms were formulated in this work as factored compositions of functional primitives.

This method will, hopefully, contribute to make inferences in estimating computing performance results of certain classes of large-scale multidimensional signal processing algorithms from their mathematical formulations in Kronecker products form, effecting, this way, a potential impact at the essential hardware implementation scales needed when dealing with fundamental understandings of planetary surface energetics and dynamics.

Acknowledgements I would like to acknowledge the assistance provided by my ex-graduate student, Ms. Dilia Rueda, and my current graduate student, Mr. Alberto Quinchanegua, in all the software-hardware implementation efforts required for the preparation of this work.

References

[1] D. Rodríguez, "SAR Point Spread Signals and Earth Surface Property Characteristics (Invited Paper)," *Proceedings of the SPIE's 44th Annual Meetings*, Denver, Colorado, July 18-23, 1999.

[2] R. Tolimieri, M. An, *Time-Frequency Representations,* Birkhäuser, Boston, 1998.

[3] R. A. Horn, C. R. Johnson, *Matrix analysis,* Cambridge Univ. Press, New York, 1985.

[4] R. Blahut, "Theory of Remote Surveillance Algorithms," *Radar and Sonar, Part I,* Springer Verlag, New York, 1991.

[5] G. Franceschetti, R. Lanari, *Synthetic Aperture Radar Processing,* CRC Press, Boca Raton, 1999.

[6] M. Soumekh, *Synthetic Aperture Radar Signal Processing,* John Wiley, New York, 1999.

[7] G. Franceschetti, et al., "An Efficient SAR Parallel Processor," *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 27, No. 2, pp. 343-353, March 1991.

[8] J. Johnson, R. Johnson, D. Rodríguez, R. Tolimieri, "A Methodology for Designing, Modifying, and Implementing Fourier Transform Algorithms on Various Architectures," *Journal of Circuits, Systems and Signal Processing, Birkhäusers*, Vol. 9, No. 4, 1990.

[9] R. Tolimieri, M. An, C. Lu, *Mathematics of Multidimensional Fourier Transform Algorithms,* Springer Verlag, New York, 1997.

[10] D. Rodríguez, D. Rueda "Kronecker-based algorithms for SAR imaging kernel formation," *SPIE's Aerosense*, Orlando, Florida,April 2000.

[11] M. Rodríguez, D. Rodríguez, "Java Based Tool Environment for Automatic Multidimensional FFT Code Generation," *International Conference on Signal Processing, Applications and Technology*, Orlando, Florida, Nov. 1999.

[12] G. Mirchandani, et al., "A Wreth Product Group Approach to Signal and Image Processing: Part II: Convolution, Correlation, and Applications," *IEEE Transactions on Signal Processing,* " Vol. 48, No. 3, pp. 749-767, March 2000.