

Representación matricial de algoritmos en paralelo de la transformada discreta de Fourier, la función discreta de ambigüedad y la distribución discreta de Cohen

por

Juan Pablo Soto Quirós y Domingo Rodríguez

RESUMEN. En este artículo se explica la representación matricial de tres transformadas discretas armónicas: la transformada discreta de Fourier, la función discreta de ambigüedad y la distribución discreta de Cohen. El desarrollo matricial se lleva a cabo utilizando un conjunto de operadores matriciales que contribuyen al análisis, diseño e implementación de algoritmos en paralelo.

1. INTRODUCCIÓN

El análisis armónico aplicado y numérico ha sido relevante en el área de la ingeniería eléctrica, donde la transformada discreta de Fourier, o DFT (del inglés *discrete Fourier transform*), ha tenido un gran papel en el estudio y la comprensión del comportamiento de señales unidimensionales discretas con dominio y rango finitos.

Sea $\mathbb{Z}_N = \{0, 1, 2, \dots, N-1\}$ y $\mathbb{C}$ el conjunto de los números complejos. Una señal unidimensional discreta con dominio y rango finitos es una función x con dominio $\mathbb{Z}_N$ y codominio $\mathbb{C}$, la cual representa a una cantidad física que varía con el tiempo:

$$\begin{array}{ccc} x : & \mathbb{Z}_N & \rightarrow \mathbb{C} \\ & n & \mapsto x[n] \end{array}$$

El espacio de señales unidimensionales discretas con dominio y rango finitos se representa con el símbolo $l^2(\mathbb{Z}_N)$,¹ el cual es isomorfo al conjunto $\mathbb{C}^N$ (matrices de N filas y 1 columna, cuyas entradas son números complejos), a través de la función identidad. En estructura $l^2(\mathbb{Z}_N)$ y $\mathbb{C}^N$ son iguales; la diferencia es conceptual. Por lo tanto podemos identificar un elemento en $l^2(\mathbb{Z}_N)$ como un elemento en $\mathbb{C}^N$.

Las señales nos ayudan a proporcionar una representación temporal de alguna situación particular, por ejemplo, del sonido de la voz. El sonido es una señal continua (o señal analógica). Realizando un muestreo de la señal continua se puede obtener una señal unidimensional discreta, con dominio y rango finitos.² A través de la DFT

¹Esta simbología hace referencia al espacio $L^2(\mathbb{R})$ (espacio de funciones cuadrado-integrables, también conocidas como funciones con energía finita); ver [22] para más detalles acerca de $L^2(\mathbb{R})$.

²A partir de este momento, en el transcurso del artículo, cuando se utilice el concepto de señal, se hará referencia a una señal unidimensional discreta con dominio y rango finitos.

se puede obtener una representación en frecuencia de dichas señales. Este tipo de señales son utilizadas en el área de procesamiento digital de señales.

Una señal x se puede representar en dos dimensiones a partir de una transformada discreta bidimensional $\mathcal{T}$ y de una señal y , conocida como señal eco:

$$\begin{aligned} \mathcal{T} : \{l^2(\mathbb{Z}_N) \times l^2(\mathbb{Z}_N), \mathbb{Z}_N \times \mathbb{Z}_N\} &\rightarrow \mathbb{C} \\ \{(x, y), (m, k)\} &\mapsto \mathcal{T}_{x,y}[m, k] \end{aligned}$$

donde $m, k \in \mathbb{Z}_N$. En este artículo, una transformada discreta bidimensional de las señales x, y se expresará como

$$\begin{aligned} \mathcal{T}_{x,y} : \mathbb{Z}_N \times \mathbb{Z}_N &\rightarrow \mathbb{C} \\ (m, k) &\mapsto \mathcal{T}_{x,y}[m, k] \end{aligned} \quad (1)$$

De esta manera, una transformada discreta bidimensional de x, y será una función con dominio $\mathbb{Z}_N \times \mathbb{Z}_N$ y codominio $\mathbb{C}$. Además, cada $\mathcal{T}_{x,y}$ se encontrará en el espacio $l^2(\mathbb{Z}_N \times \mathbb{Z}_N)$, el cual representa el espacio de señales bidimensionales discretas finitas. El espacio $l^2(\mathbb{Z}_N \times \mathbb{Z}_N)$ es isomorfo a $\mathbb{C}^{N \times N}$, a través de la función identidad; la diferencia es similar a la existente entre $l^2(\mathbb{Z}_N)$ y $\mathbb{C}^N$, anteriormente descrita.

Dos ejemplos de transformadas discretas bidimensionales son la función discreta de ambigüedad, o DAF (del inglés *discrete ambiguity function*) y la distribución discreta de Cohen, o DCD (del inglés *discrete Cohen distribution*). El concepto de representación bidimensional en tiempo y frecuencia es propio de la ingeniería eléctrica, donde se utiliza para resolver problemas de radares y sonares [5], secuencias CAZAC y el diseño de ondas [2, 11, 16] y criptografía [17], entre otros. Además, ha sido campo de estudio para matemáticos, puesto que aporta nuevos conocimientos en diversas áreas de la matemática como, por ejemplo, el análisis armónico [3] o la teoría de grupos [14]; pero tales estudios no han profundizado en el contexto matricial como el que aquí presentaremos.

El presente artículo desarrolla un marco computacional para la transformada discreta de Fourier, la función discreta de ambigüedad y la transformada discreta de Cohen. El marco computacional se basa en estructuras matriciales, utilizando un conjunto de operadores que contribuyen al análisis, diseño e implementación de algoritmos en paralelo. Entre los operadores matriciales empleados se encuentran los productos de Hadamard y de Kronecker, la suma directa de matrices, los operadores de permutación y el operador *vec*, entre otros.

El artículo está organizado de la siguiente manera: en la Sección 2 se introduce la notación matricial que se empleará. En la Sección 3 se definen los operadores matriciales. En la Sección 4 se define y desarrolla la representación matricial de la transformada discreta de Fourier, la función discreta de ambigüedad y la distribución discreta de Cohen. Finalmente, se presentan algunas conclusiones en la Sección 5.

La sección referente a la transformada discreta de Fourier se basa en el trabajo realizado por Van Loan [21] y Tolimieri *et al.* [18]. En el desarrollo de nuestro artículo se presentan dos resultados originales (Teoremas 5 y 6), con sus respectivas demostraciones, realizadas por los autores. Además, los algoritmos desarrollados en este

trabajo fueron implementados en MATLAB®, utilizando dos *toolbox* que permiten el cálculo en paralelo: *pMatlab Toolbox* ©, desarrollado por el Laboratorio Lincoln del MIT,³ y *Parallel Computing Toolbox* ©, diseñado por Mathworks.⁴

2. NOTACIÓN MATRICIAL

Sea $\mathbb{C}^{M \times N}$ el conjunto de matrices de M filas y N columnas tal que sus entradas son números complejos $\mathbb{C} = \{a + bi \mid a, b \in \mathbb{R}, i = \sqrt{-1}\}$. La matriz $I_N \in \mathbb{C}^{N \times N}$ es la matriz identidad de orden N . Para los propósitos del artículo, las filas y las columnas de cualquier matriz $A \in \mathbb{C}^{M \times N}$ se indexarán con elementos de $\mathbb{Z}_M$ y $\mathbb{Z}_N$, respectivamente. La entrada (m, n) de la matriz A se representa como $A_{m,n}$, con $m \in \mathbb{Z}_M$ y $n \in \mathbb{Z}_N$. Además, una columna n de la matriz A se representará como A_n o $A[:, n]$. La notación $A[:, n]$, similar a la de MATLAB®, se utilizará en el diseño de los algoritmos.

EJEMPLO 1. Sea $B \in \mathbb{C}^{3 \times 2}$ dada por

$$B = \begin{pmatrix} -2 & -i \\ i & 0 \\ 1 & 2 \end{pmatrix}.$$

La matriz B tiene 3 filas y 2 columnas:

- las filas están indexadas con los elementos de $\mathbb{Z}_3 = \{0, 1, 2\}$:

$$\text{Fila } 0 = (-2, -i), \quad \text{Fila } 1 = (i, 0) \quad \text{y} \quad \text{Fila } 2 = (1, 2);$$

- las columnas están indexadas con los elementos de $\mathbb{Z}_2 = \{0, 1\}$:

$$\text{Columna } 0 = (-2, i, 1)^T \quad \text{y} \quad \text{Columna } 1 = (-i, 0, 2)^T.$$

Además, las entradas de la matriz son

$$B_{0,0} = -2, \quad B_{1,0} = i, \quad B_{2,0} = 1, \quad B_{0,1} = -i, \quad B_{1,1} = 0 \quad \text{y} \quad B_{2,1} = 2. \quad \square$$

El conjunto de matrices de N filas y 1 columna, tal que sus entradas son números complejos, se representa con el símbolo $\mathbb{C}^N$. Los elementos de $\mathbb{C}^N$ se llaman vectores de entradas complejas, o simplemente vectores. Los elementos de este conjunto se representan con letras minúsculas; además, cada entrada de $x \in \mathbb{C}^N$ se indexará con elementos de $\mathbb{Z}_N$:

$$x = \begin{pmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{pmatrix}.$$

³<http://www.ll.mit.edu/mission/isr/pmatlab/pmatlab.html>

⁴<http://www.mathworks.com/products/parallel-computing>

Si $x \in \mathbb{C}^N$, entonces $x[m : n] \in \mathbb{C}^{n-m+1}$ representa al vector

$$x[m : n] = \begin{pmatrix} x[m] \\ x[m+1] \\ \vdots \\ x[n-1] \\ x[n] \end{pmatrix},$$

donde $0 \leq m \leq n \leq N-1$. Además, $\mathbf{1}_N \in \mathbb{C}^N$ representa al vector que cumple $\mathbf{1}_N[n] = 1$ para todo $n \in \mathbb{Z}_N$. A continuación se describen los operadores matriciales que se utilizarán en el transcurso del artículo.

3. OPERADORES MATRICIALES

Sean V y W dos espacios vectoriales. Una aplicación $\mathcal{O}$ de V a W , $\mathcal{O} : V \rightarrow W$, se denomina operador. En el caso que V y W sean iguales a $\mathbb{C}^{M \times N}$, o al producto cartesiano de P conjuntos $\mathbb{C}^{M_k \times N_k}$, $k \in \mathbb{Z}_P$, entonces el operador $\mathcal{O}$ se denomina un operador matricial. Entre los operadores matriciales más comunes se encuentran la suma y resta de matrices, el producto entre un escalar (número complejo) y una matriz, y el producto usual de matrices. Por ejemplo, la suma de matrices se puede considerar como el operador

$$\begin{aligned} + : \mathbb{C}^{M \times N} \times \mathbb{C}^{M \times N} &\rightarrow \mathbb{C}^{M \times N} \\ (A, B) &\mapsto A + B \end{aligned}$$

En este artículo también se utilizan otros operadores matriciales, los cuales van a contribuir al diseño e implementación de algoritmos en paralelo. Estos operadores son:

- | | |
|--------------------------|-------------------------------------|
| 1. Producto de Hadamard | 4. Operadores de Permutación |
| 2. Producto de Kronecker | 5. Operador de Acumulación |
| 3. Suma Directa | 6. Operador <i>vec</i> y su inverso |

3.1. PRODUCTOS DE KRONECKER Y DE HADAMARD

El producto de Hadamard, también conocido como producto de entradas o producto de Schur, y el producto de Kronecker, también conocido como producto directo o producto tensorial, son conceptos que tienen gran importancia en el desarrollo computacional de la ingeniería eléctrica, especialmente en el campo del procesamiento digital de señales [13]. Estos productos difieren de la forma ordinaria del producto de matrices.

DEFINICIÓN 1 (Producto de Hadamard). *El producto de Hadamard se define como el operador $\odot : \mathbb{C}^{M \times N} \times \mathbb{C}^{M \times N} \rightarrow \mathbb{C}^{M \times N}$ tal que, para $A, B \in \mathbb{C}^{M \times N}$, $A \odot B \in \mathbb{C}^{M \times N}$ cumple $(A \odot B)_{m,n} = A_{m,n} \cdot B_{m,n}$.*

DEFINICIÓN 2 (Producto de Kronecker). *El producto de Kronecker se define como el operador $\otimes : \mathbb{C}^{M \times N} \times \mathbb{C}^{P \times Q} \rightarrow \mathbb{C}^{MP \times NQ}$ tal que, para $A \in \mathbb{C}^{M \times N}$ y $B \in \mathbb{C}^{P \times Q}$, $A \otimes B \in \mathbb{C}^{MP \times NQ}$ es*

$$A \otimes B = \begin{pmatrix} A_{0,0}B & A_{0,1}B & \cdots & A_{0,(N-1)}B \\ A_{1,0}B & A_{1,1}B & \cdots & A_{1,(N-1)}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{(M-1),0}B & A_{(M-1),1}B & \cdots & A_{(M-1),(N-1)}B \end{pmatrix}.$$

El producto de Kronecker de dos matrices A y B consiste en una matriz por bloques, donde cada bloque representa el producto escalar de una entrada de la matriz A y la matriz B . Es fácil verificar que este tipo de producto no es conmutativo. Además, el producto de Kronecker es una manera compacta y práctica de expresar algoritmos del álgebra lineal que son esparcidos.⁵

Una ventaja de los productos de Hadamard y de Kronecker es que contribuyen al análisis, diseño e implementación de algoritmos en paralelo.

EJEMPLO 2. Sea $A \in \mathbb{C}^{R \times M}$, $x \in \mathbb{C}^{RN}$, $y \in \mathbb{C}^{MN}$. Consideremos la operación

$$x \odot (I_N \otimes A)y.$$

Este producto de matrices se puede descomponer de la siguiente manera:

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} \odot \begin{pmatrix} A & & \\ & A & \\ & & \ddots \\ & & & A \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix} = \begin{pmatrix} x_0 \odot Ay_0 \\ x_1 \odot Ay_1 \\ \vdots \\ x_{N-1} \odot Ay_{N-1} \end{pmatrix},$$

donde $x_m \in \mathbb{C}^R$ y $y_m \in \mathbb{C}^M$. La operación $x \odot (I_N \otimes A)y$ tiene una complejidad computacional de $O(MRN^2)$, pero al utilizar los productos de Hadamard y de Kronecker, se puede dividir en N operaciones $x_m \odot Ay_m$, para $m \in \mathbb{Z}_N$, por lo que la complejidad computacional pasaría a ser $O(RMN)$. La estructura de $x(I_N \otimes A)y$ permite su implementación utilizando cálculo en paralelo, ya que el cálculo de cada $x_m \odot Ay_m$ se realiza de manera independiente, para cada valor de m . $\square$

El cálculo de $I_N \otimes A$ se denomina operación paralela [18]. El Algoritmo 1 muestra la implementación del ejemplo anterior.

El siguiente teorema presenta algunas de las propiedades más importantes del producto de Kronecker. Estas propiedades se utilizan a lo largo del presente trabajo. El lector puede verificar los detalles de la demostración del teorema en [9, 18].

TEOREMA 1. Sea $A \in \mathbb{C}^{M \times N}$, $B \in \mathbb{C}^{R \times S}$, $C \in \mathbb{C}^{N \times P}$ y $D \in \mathbb{C}^{S \times T}$. Entonces

- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
- $(A \otimes B)(C \otimes D) = AC \otimes BD$.
- $A \otimes B = (A \otimes I_R)(I_N \otimes B)$.
- $I_R \otimes I_S = I_{RS}$.

⁵Un algoritmo se dice esparcido (o disperso) cuando la matriz que representa a este algoritmo es de gran tamaño y la mayor parte de sus entradas son iguales a cero.

Algoritmo 1: Cálculo operación $x \odot (I_N \otimes A)y$.

Entrada: $x \in \mathbb{C}^{RN}$, $y \in \mathbb{C}^{MN}$ y $A \in \mathbb{C}^{R \times M}$

Salida: $S \in \mathbb{C}^{RN}$

```

1: para  $n \leftarrow 0 : N - 1$  hacer
2:    $x_{aux} \leftarrow x[nR : (n+1)R - 1]$ 
3:    $y_{aux} \leftarrow y[nM : (n+1)M - 1]$ 
4:    $S[nR : (n+1)R - 1] \leftarrow x_{aux} \odot Ay_{aux}$ 
5: fin para

```

3.2. SUMA DIRECTA

DEFINICIÓN 3 (Suma Directa). *La suma directa matricial se define como el operador $\oplus : \mathbb{C}^{M \times N} \times \mathbb{C}^{P \times Q} \rightarrow \mathbb{C}^{(M+P) \times (N+Q)}$ tal que, para $A \in \mathbb{C}^{M \times N}$ y $B \in \mathbb{C}^{P \times Q}$, $A \oplus B \in \mathbb{C}^{(M+P) \times (N+Q)}$ es*

$$A \oplus B = \begin{pmatrix} A & \\ & B \end{pmatrix}.$$

Para un conjunto indexado de matrices $\{C_n\}_{n \in \mathbb{Z}_N}$, tal que $C_n \in \mathbb{C}^{M_n \times N_n}$, la suma directa se representa

$$C = \bigoplus_{n \in \mathbb{Z}_N} C_n = C_0 \oplus C_1 \oplus \cdots \oplus C_{N-1} = \begin{pmatrix} C_0 & & & \\ & C_1 & & \\ & & \ddots & \\ & & & C_{N-1} \end{pmatrix}$$

donde $C \in \mathbb{C}^{M \times N}$ con $M = \sum_{n \in \mathbb{Z}_N} M_n$ y $N = \sum_{n \in \mathbb{Z}_N} N_n$. Si el conjunto $\{C_n\}_{n \in \mathbb{Z}_N}$ cumple que $C_k = C_l$, para todo $k, l \in \mathbb{Z}_N$, entonces

$$\bigoplus_{n \in \mathbb{Z}_N} C_n = I_N \otimes C_n.$$

3.3. OPERADORES DE PERMUTACIÓN

DEFINICIÓN 4 (Operadores Lineales y Unitarios).

1. $\mathcal{O} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ es un operador lineal si $\mathcal{O}(\alpha v_1 + v_2) = \alpha \mathcal{O}(v_1) + \mathcal{O}(v_2)$, donde $v_1, v_2 \in \mathbb{C}^N$ y $\alpha \in \mathbb{C}$.
2. Un operador $\mathcal{U} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ es un operador unitario si es lineal y $\mathcal{U}\mathcal{U}^* = \mathcal{I}$, donde $\mathcal{U}^*$ es el operador adjunto de $\mathcal{U}$ y $\mathcal{I}$ es el operador identidad.

Todo operador unitario en $\mathbb{C}^N$ tiene una representación matricial. Por ejemplo, I_N es la representación matricial del operador identidad $\mathcal{I}$; además la representación matricial del operador adjunto $\mathcal{U}^*$ es la matriz transpuesta conjugada del operador $\mathcal{U}$. Un ejemplo de un operador unitario es el operador de permutación. El operador

de permutación, $\mathcal{L} : \mathbb{C}^N \rightarrow \mathbb{C}^N$, es un operador que realiza un re-acomodo de las entradas de un vector en $\mathbb{C}^N$. La representación matricial del operador de permutación, o simplemente matriz de permutación, es una matriz binaria cuadrada de orden N que tiene exactamente una entrada con el valor de 1 en cada fila y columna, y el valor de 0 en las otras entradas. La matriz de permutación se representa como $L \in \mathbb{C}^{N \times N}$.

EJEMPLO 3. Sea $v \in \mathbb{C}^4$. Una permutación para v es $\mathcal{L}\{v\} = (v[0], v[2], v[1], v[3])^T$. La representación matricial del operador $\mathcal{L}$ es la matriz $L \in \mathbb{C}^{4 \times 4}$ dada por

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad \square$$

Los operadores de permutación también pueden actuar sobre $\mathbb{C}^{M \times N}$, realizando permutaciones a las filas.

EJEMPLO 4. Sea $A \in \mathbb{C}^{4 \times 4}$. Utilizando el operador de permutación $\mathcal{L}$ del Ejemplo 3, aplicado a la matriz A , se obtiene

$$\mathcal{L}\{A\} = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}. \quad \square$$

Tres ejemplos de operadores de permutación son el operador de permutación de orden N y paso R , el operador de traslación y el operador de orden exponencial.

1. **Operador de Permutación de Orden N y Paso R :** Sea $N = RS$. El operador de permutación de orden N y paso R , $\mathcal{L}_R^N$, es el operador definido por la regla $\mathcal{L}_R^N\{v \otimes w\} = w \otimes v$, para $v \in \mathbb{C}^R$ y $w \in \mathbb{C}^S$. La representación matricial del operador $\mathcal{L}_R^N$ es la matriz $L_R^N \in \mathbb{C}^{N \times N}$. Además

$$L_R^N L_S^N = I_N. \quad (2)$$

Como se mencionó en la sección anterior, el producto de Kronecker no es conmutativo. A pesar de ello, este tipo de matrices de permutación tratan de dar una noción de conmutatividad, como se explica en el Teorema 2. El lector puede ver los detalles de la demostración en [9, 18].

TEOREMA 2 (Conmutación de productos de Kronecker). *Sea $N = RS$, $A \in \mathbb{C}^{R \times R}$ y $B \in \mathbb{C}^{S \times S}$. Entonces $A \otimes B = L_R^N (B \otimes A) L_S^N$.*

La implementación de este operador se puede hacer de tal manera que su complejidad computacional sea lineal. El Algoritmo 2 muestra tal implementación.

2. **Operador de Traslación:** Sea $m \in \mathbb{Z}_N$. El operador de traslación, $\mathcal{S}_m$, es el operador de permutación definido por la regla $\mathcal{S}_m\{v\}[n] = v[(n - m)_N]$, para

Algoritmo 2: Permutación de Orden N y Paso R ($N = RS$).**Entrada:** $x \in \mathbb{C}^N$ **Salida:** $y \in \mathbb{C}^N$

```

1: para  $r \leftarrow 0 : R - 1$  hacer
2:   para  $s \leftarrow 0 : S - 1$  hacer
3:      $y[r + Rs] \leftarrow x[Rs + s]$ 
4:   fin para
5: fin para

```

$v \in \mathbb{C}^N$, donde $\langle a \rangle_N = a$ módulo N . La constante m se denomina constante de traslación. La constante de traslación es un número en el conjunto $\mathbb{Z}_N$, por lo que cuando se utilice la notación $\mathcal{S}_{-m}$, el valor $-m$ es una representación módulo N , es decir, $\mathcal{S}_{-m} = \mathcal{S}_{\langle -m \rangle_N}$. Además, si $m \in \mathbb{Z}_N$, entonces $\mathcal{S}_{-m} = \mathcal{S}_{N-m}$, donde $N - m \in \mathbb{Z}_N$. La representación matricial del operador $\mathcal{S}_m$ es la matriz $S_m \in \mathbb{C}^{N \times N}$. La implementación de este operador se puede hacer de tal manera que su complejidad computacional sea lineal. El Algoritmo 3 muestra tal implementación.

Algoritmo 3: Permutación de Traslación con constante m .**Entrada:** $m \in \mathbb{Z}_N$ y $x \in \mathbb{C}^N$ **Salida:** $y \in \mathbb{C}^N$

```

1: para  $n \leftarrow 0 : N - 1$  hacer
2:    $y[n] \leftarrow x[\langle n - m \rangle_N]$ 
3: fin para

```

3. Operador de Orden Exponencial: El operador de orden exponencial se define para elementos de $\mathbb{C}^N$, donde N es un número primo. Consideremos el conjunto $\mathbb{U}_N = \{1, 2, \dots, N - 1\}$. Si N es un número primo, entonces existe un elemento $a \in \mathbb{U}_N$ que genera, mediante potencias, todos los elementos de $\mathbb{U}_N$. Es decir, tal que el conjunto $\{1, a, \langle a^2 \rangle_N, \dots, \langle a^{N-2} \rangle_N\}$ sea una permutación de los elementos de $\mathbb{U}_N$.

Consideremos el conjunto $\pi_a = \{0, 1, a, \langle a^2 \rangle_N, \dots, \langle a^{N-2} \rangle_N\}$. El conjunto π_a es una permutación del conjunto $\mathbb{Z}_N$. Por lo tanto, si N es un número primo y a es un generador de $\mathbb{U}_N$, el operador de orden exponencial, $\mathcal{L}_a$, es el operador de permutación definido por la regla

$$\mathcal{L}_a\{v\}[n] = \begin{cases} v[0], & \text{si } n = 0, \\ v[\langle a^{n-1} \rangle_N], & \text{si } n \in \mathbb{U}_N, \end{cases}$$

para $v \in \mathbb{C}^N$. La representación matricial del operador de permutación exponencial es la matriz $L_a \in \mathbb{C}^{N \times N}$. La implementación de este operador se puede hacer de tal manera que su complejidad computacional sea lineal. El Algoritmo 4 muestra tal implementación.

Algoritmo 4: Permutación de Orden Exponencial, con generador a .

Entrada: $x \in \mathbb{C}^N$ y $a \in \mathbb{U}_N$ (generador)

Salida: $y \in \mathbb{C}^N$

```

1:  $v = 0$ 
2: para  $n \leftarrow 0 : N - 1$  hacer
3:    $y[n] \leftarrow x[v]$ 
4:    $v = \langle a^n \rangle_N$ 
5: fin para

```

3.4. OTROS OPERADORES

1. **Operador de Acumulación de Matrices:** Sea $\{A_n\}_{n \in \mathbb{Z}_N}$ un conjunto de N matrices tal que $A_n \in \mathbb{C}^{M_n \times P}$. El operador de acumulación de matrices, $\sqcup : \prod_{n \in \mathbb{Z}_N} \mathbb{C}^{M_n \times P} \rightarrow \mathbb{C}^{M \times P}$, donde $M = \sum_{n \in \mathbb{Z}_N} M_n$, se define como

$$\sqcup \{\{A_n\}\} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{N-1} \end{pmatrix}.$$

La única condición que deben tener los elementos de $\{A_n\}$ es coincidir en el número de columnas. Con el fin de simplificar la notación, el operador de acumulación se representará como

$$\sqcup \{\{A_n\}\} = \bigsqcup_{n \in \mathbb{Z}_N} A_n.$$

2. **Operador Vec:** El operador $vec, \mathcal{V} : \mathbb{C}^{M \times N} \rightarrow \mathbb{C}^{MN}$, es un operador que actúa sobre $\mathbb{C}^{M \times N}$ y da como resultado un vector formado por la acumulación de las columnas de las matrices. Si $B \in \mathbb{C}^{M \times N}$ y B_n representa la columna n de la matriz B , entonces

$$\mathcal{V}\{B\} = \begin{pmatrix} B_0 \\ B_1 \\ \vdots \\ B_{N-1} \end{pmatrix}.$$

3. **Operador Vec Inverso:** El inverso del operador $vec, \mathcal{R}_{M,N} : \mathbb{C}^{MN} \rightarrow \mathbb{C}^{M \times N}$, también conocido como operador *re-shape*, es el operador que transforma un vector de dimensión MN a una matriz de dimensión $M \times N$. Sea $v \in \mathbb{C}^{MN}$ dado por

$$v = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix}, \quad v_n \in \mathbb{C}^M;$$

entonces $\mathcal{R}_{M,N}\{v\} = (v_0 \ v_1 \ \dots \ v_{N-1})$, donde cada v_n representa la columna n de la matriz resultante.

Detalles de la implementación de estos operadores se aportan en la Sección 4.2 y en la Sección 4.3, aplicados a la representación matricial de la función discreta de ambigüedad y de la distribución discreta de Cohen, respectivamente. La Proposición 1 describe una relación de los operadores $\mathcal{V}$ y $\mathcal{R}_{N,N}$ con la matriz de permutación $L_N^{N^2}$.

PROPOSICIÓN 1. Sean $v \in \mathbb{C}^{N^2}$ y $A \in \mathbb{C}^{N \times N}$. Entonces

$$1. (\mathcal{R}_{N,N}\{v\})^T = \mathcal{R}_{N,N}\{L_N^{N^2} v\}. \quad 2. \mathcal{V}\{A^T\} = L_N^{N^2} \mathcal{V}\{A\}.$$

DEMOSTRACIÓN.

1. Sea $v \in \mathbb{C}^{N^2}$ dado por

$$v = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix},$$

con $v_n \in \mathbb{C}^N$. Entonces $\mathcal{R}_{N,N}\{v\} = (v_0 \ v_1 \ \dots \ v_{N-1})$. Pero

$$(\mathcal{R}_{N,N}\{v\})_{m,n} = v_j[i] \Rightarrow (\mathcal{R}_{N,N}\{v\})_{m,n}^T = v_m[n],$$

para $m, n \in \mathbb{Z}_N$. Por otro lado

$$L_N^{N^2} v = \begin{pmatrix} v'_0 \\ v'_1 \\ \vdots \\ v'_{N-1} \end{pmatrix},$$

donde $v'_j[i] = v_i[j]$. Entonces $(\mathcal{R}_{N,N}\{L_N^{N^2} v\})_{m,n} = v'_n[m] = v_m[n]$.

2. Se sabe que si $\mathcal{V}\{A\} = a$, con $a \in \mathbb{C}^{N^2}$, entonces $A = \mathcal{R}_{N,N}\{a\}$. Del primer punto de esta proposición se deduce que $(\mathcal{R}_{N,N}\{a\})^T = \mathcal{R}_{N,N}\{\mathcal{L}_N^{N^2} a\}$. Aplicando el operador $\mathcal{V}$ a ambos lados de la ecuación anterior, se obtiene

$$\mathcal{V}\{(\mathcal{R}_{N,N}\{a\})^T\} = \mathcal{V}\{\mathcal{R}_{N,N}\{\mathcal{L}_N^{N^2} a\}\} \Rightarrow \mathcal{V}\{A^T\} = L_N^{N^2} \mathcal{V}\{A\}. \quad \square$$

4. TRANSFORMADAS DISCRETAS ARMÓNICAS

Una transformada discreta armónica de una señal $x \in l^2(\mathbb{Z}_N)$ es una aplicación $\mathcal{T}_x$ tal que dicha transformada realiza una representación de x como una superposición de ondas, conocidas como armónicos. Un ejemplo de una onda es la señal $y[n] = \omega_N^n$, donde $\omega_N = e^{2\pi i/N}$ es la N -ésima raíz de la unidad. A continuación se explicarán tres tipos de transformadas discretas armónicas: la transformada discreta de Fourier

o DFT (del inglés *discrete Fourier transform*), la función discreta de ambigüedad o DAF (del inglés *discrete ambiguity function*) y la distribución discreta de Cohen o DCD (del inglés *discrete Cohen distribution*).

La DFT de una señal $x \in l^2(\mathbb{Z}_N)$ es una transformada unidimensional:

$$\begin{array}{ccc} \mathcal{F}_x : & \mathbb{Z}_N & \rightarrow \mathbb{C} \\ & n & \rightarrow \mathcal{F}_x[n] \end{array}$$

En cambio, la DAF y la DCD de la señal $x \in l^2(\mathbb{Z}_N)$, con la señal eco $y \in l^2(\mathbb{Z}_N)$, son transformadas bidimensionales:

$$\begin{array}{ccc} \mathcal{A}_{x,y} : & \mathbb{Z}_N \times \mathbb{Z}_N & \rightarrow \mathbb{C} \\ & (m, k) & \mapsto \mathcal{A}_{x,y}[m, k] \end{array} \quad \begin{array}{ccc} \mathcal{C}_{x,y} : & \mathbb{Z}_N \times \mathbb{Z}_N & \rightarrow \mathbb{C} \\ & (m, k) & \mapsto \mathcal{C}_{x,y}[m, k] \end{array}$$

4.1. TRANSFORMADA DISCRETA DE FOURIER

DEFINICIÓN 5 (Transformada Discreta de Fourier). *Sea $x \in l^2(\mathbb{Z}_N)$. La transformada discreta de Fourier de orden N de la señal x , o simplemente DFT, se define como la aplicación $\mathcal{F}_x : \mathbb{Z}_N \rightarrow \mathbb{C}$ dada por*

$$\mathcal{F}_x[k] = \sum_{n \in \mathbb{Z}_N} x[n] \omega_N^{-nk}, \quad k \in \mathbb{Z}_N,$$

donde $\omega_N = e^{2\pi i/N}$.

Ya que el conjunto $l^2(\mathbb{Z}_N)$ es isomorfo al conjunto $\mathbb{C}^N$, a través de la función identidad, entonces una señal $l^2(\mathbb{Z}_N)$ se puede interpretar como un elemento de $\mathbb{C}^N$ y, mediante esta identificación, realizar operaciones matriciales a la señal x . La DFT tiene una representación matricial a partir de la matriz finita de la transformada de Fourier [18, 21], también conocida como matriz DFT.

DEFINICIÓN 6 (Matriz DFT). *La matriz DFT se define como $F_N \in \mathbb{C}^{N \times N}$ con $(F_N)_{m,n} = \omega_N^{-mn}$, para $m, n \in \mathbb{Z}_N$. La representación matricial de la DFT de x es*

$$\mathcal{F}_x = F_N x.$$

4.1.1. TRANSFORMADA RÁPIDA DE FOURIER

J. W. Cooley y J. W. Tukey [8] fueron los primeros en desarrollar un algoritmo para realizar el cálculo de la DFT de una manera rápida. Este algoritmo es conocido como la transformada rápida de Fourier de Cooley-Tukey o FFT Cooley-Tukey (del inglés *fast Fourier transform*). La FFT reduce significativamente el coste computacional de la DFT para señales de longitud N , siendo N un número compuesto [18]. Sea $x \in l^2(\mathbb{Z}_N)$ tal que N es un número compuesto: $N = RS$. El algoritmo

de Cooley-Tukey plantea que la DFT de x , de orden N , se puede calcular utilizando R transformadas discretas de Fourier de orden S , es decir,

$$\mathcal{F}_x[k] = \begin{cases} \mathcal{F}_{x_0}[m], & \text{si } k = Rm, \\ \mathcal{F}_{x_1}[m], & \text{si } k = Rm + 1, \\ \vdots & \\ \mathcal{F}_{x_{R-1}}[m], & \text{si } k = Rm + (R-1), \end{cases}$$

con $k \in \mathbb{Z}_N$ y $m \in \mathbb{Z}_S$, donde

$$\mathcal{F}_{x_r}[m] = \sum_{s \in \mathbb{Z}_S} x_r[s] \omega_S^{-sm} \quad \text{y} \quad x_r[s] = \omega_N^{-sr} \sum_{r' \in \mathbb{Z}_R} x[s + r'S] \omega_R^{-r'r}$$

para $r \in \mathbb{Z}_R$. Importantes investigadores como J. Johnson *et al.* [10] o Tolimieri *et al.* [18] han obtenido una factorización de la matriz DFT utilizando el algoritmo FFT como referencia. El Teorema 3 explica dicha factorización y utiliza el producto de Kronecker, permitiendo su implementación en paralelo. El lector puede verificar los detalles de la demostración en [18].

TEOREMA 3. *Sea $N = RS$. La matriz DFT se puede factorizar de la siguiente manera:*

$$F_N = L_S^N (I_R \otimes F_S) L_R^N T_R^N (I_S \otimes F_R) L_S^N, \quad (3)$$

donde $T_R^N \in \mathbb{C}^{N \times N}$ es conocido como la matriz «twiddle factor», la cual es una matriz diagonal por bloques tal que

$$T_R^N = \bigoplus_{s \in \mathbb{Z}_S} (D_R^N)^s,$$

con $D_R^N = \text{diag}(1, \omega_N^{-1}, \dots, \omega_N^{-(R-1)})$.

La ecuación (3) muestra que la matriz DFT se puede factorizar utilizando factores que contengan el producto de Kronecker, por lo que se puede realizar una implementación en paralelo. El Algoritmo 5 muestra la implementación del cálculo de $F_N x$, basada en el teorema anterior.

En el Algoritmo 5 se puede observar que los pasos 3–4 y 17–18, que representan el cálculo de las DFT de orden S y R , respectivamente, son independientes en cada iteración, por lo que pueden implementarse en paralelo. Los pasos 9–13 realizan el producto de la matriz «twiddle factor»; al ser dicha matriz diagonal y por bloques, dichos pasos son también independientes en cada iteración, por lo que se pueden implementar también en paralelo.

Otro caso a considerar es aquel en el que N es un número primo. Este trabajo fue desarrollado por S. Winograd [23]. El Teorema 4 expone los detalles de la factorización en este supuesto.

TEOREMA 4. *Sea N un número primo y sea $a \in \mathbb{U}_N$ un generador del conjunto $\mathbb{U}_N$. La matriz DFT puede factorizarse de la siguiente manera:*

$$F_N = L_a^{-1} (1 \oplus F_{N-1}) (1 \oplus D) (1 \oplus F_{N-1}) A L_a,$$

Algoritmo 5: Cálculo de $F_N x$, donde N es compuesto ($N = RS$).

Entrada: $x \in \mathbb{C}^N$

Salida: $y \in \mathbb{C}^N$

<pre> 1: $y \leftarrow \mathcal{L}_S^N\{x\}$ 2: para $n \leftarrow 0 : S - 1$ hacer 3: $y_{aux} \leftarrow y[nR : (n+1)R - 1]$ 4: $y[nR : (n+1)R - 1] \leftarrow \mathcal{F}_{y_{aux}}$ 5: fin para 6: $u \leftarrow \mathbf{1}_R$ 7: $d \leftarrow (1, \omega_N^{-1}, \dots, \omega_N^{-(R-1)})^T$ 8: para $s \leftarrow 1 : S - 1$ hacer 9: $y_{aux} \leftarrow y[sR : (s+1)R - 1]$ 10: para $n \leftarrow 1 : s$ hacer 11: $u \leftarrow u \odot d$ 12: fin para 13: $y[sR : (s+1)R - 1] \leftarrow d \odot y_{aux}$ 14: fin para </pre>	<pre> 15: $y \leftarrow \mathcal{L}_R^N\{x\}$ 16: para $n \leftarrow 0 : R - 1$ hacer 17: $y_{aux} \leftarrow y[nS : (n+1)S - 1]$ 18: $y[nS : (n+1)S - 1] \leftarrow \mathcal{F}_{y_{aux}}$ 19: fin para 20: $y \leftarrow \mathcal{L}_S^N\{x\}$ </pre>
---	---

donde

- $D \in \mathbb{C}^{(N-1) \times (N-1)}$ es una matriz diagonal tal que $D = F_{N-1}^{-1} W F_{N-1}^{-1}$, donde $W \in \mathbb{C}^{(N-1) \times (N-1)}$ es el núcleo de Winograd: $W_{m,n} = (\omega_N)^{\langle a^{m+n} \rangle_N}$;
- $A \in \mathbb{C}^{N \times N}$ es

$$A = \begin{pmatrix} 1 & \mathbf{1}_{N-1}^T \\ -\mathbf{1}_{N-1} & I_{N-1} \end{pmatrix}.$$

La idea del teorema anterior es factorizar la matriz F_N de tal manera que los factores contengan la matriz F_{N-1} . Como N es primo, entonces $N - 1$ es compuesto, si $N > 3$, por lo que se puede utilizar el Teorema 3 para su cálculo. El Algoritmo 6 muestra la implementación del cálculo de $F_N x$, basada en el teorema anterior.

Algoritmo 6: Cálculo de $F_N x$, donde N es primo.

Entrada: $x \in \mathbb{C}^N$ y $a \in \mathbb{U}_N$ (generador)

Salida: $y \in \mathbb{C}^N$

<pre> 1: $y \leftarrow \mathcal{L}_a\{x\}$ 2: $const \leftarrow \sum_{n \in \mathbb{Z}_N} y[n]$ 3: para $n \leftarrow 1 : N - 1$ hacer 4: $y[n] \leftarrow y[n] - y[0]$ 5: fin para 6: $y[0] \leftarrow const$ </pre>	<pre> 7: $y_{aux} \leftarrow y[1 : N - 1]$ 8: $y_{aux} \leftarrow \mathcal{F}_{y_{aux}}$ 9: $y_{aux} \leftarrow D \cdot y_{aux}$ 10: $y_{aux} \leftarrow \mathcal{F}_{y_{aux}}$ 11: $y[1 : N - 1] \leftarrow y_{aux}$ 12: $y \leftarrow \mathcal{L}_a^{-1}\{y\}$ </pre>
---	---

A partir del Teorema 3 y del Teorema 4 pueden considerarse variaciones de las factorizaciones de la matriz F_N , que tratan de aumentar la rapidez del cálculo de la

DFT. Por ejemplo, podemos mencionar los algoritmos de Pease, Good-Thomas, el algoritmo *mixed Radix* Cooley-Tukey o el algoritmo *mixed Radix* Agarwal-Cooley, entre otros. Ver [18, 21] para más detalles.

4.1.2. TRANSFORMADA DISCRETA DE FOURIER EN 2 DIMENSIONES

La Transformada Discreta de Fourier en 2 dimensiones es una transformada bidimensional, que extiende el concepto de la DFT para señales en $l^2(\mathbb{Z}_N \times \mathbb{Z}_N)$.

DEFINICIÓN 7. Sea $x \in l^2(\mathbb{Z}_N \times \mathbb{Z}_N)$. Entonces la transformada de Fourier en 2 dimensiones de x , o 2D-DFT, se define como la aplicación $\mathcal{F}_x^{2D} : \mathbb{Z}_N \times \mathbb{Z}_N \rightarrow \mathbb{C}$ dada por

$$\mathcal{F}_x^{2D}[m, k] = \sum_{\tau \in \mathbb{Z}_N} \sum_{\nu \in \mathbb{Z}_N} x[\tau, \nu] \omega_N^{-(m\tau + k\nu)}.$$

Al ser bidimensional, la 2D-DFT de x puede ser interpretada como una matriz $F_x^{2D} \in \mathbb{C}^{N \times N}$ tal que $(F_x^{2D})_{m,k} = \mathcal{F}_x^{2D}[m, k]$. Tolimieri *et al.* [19] y Kumhom *et al.* [12] explican que la matriz F_N^{2D} puede ser calculada de la siguiente forma:

$$F_x^{2D} = \mathcal{R}_{N,N}\{(F_N \otimes F_N)\mathcal{V}\{X\}\}, \quad (4)$$

donde $X \in \mathbb{C}^{N \times N}$ tal que $X_{m,k} = x[m, k]$. Los detalles de la implementación del algoritmo para la 2D-DFT se describen en la Sección 4.3.

4.2. FUNCIÓN DISCRETA DE AMBIGÜEDAD

En 1953, P. M. Woodward definió la función continua de ambigüedad [24] como un mecanismo formulado para describir el efecto Doppler en los receptores de filtro apareados [3]. Woodward reconoció la influencia de la teoría de la comunicación de Shannon, a partir de 1948, en sus ideas, y explicó la importancia de la «ambigüedad» en el procesamiento de señales de radares, tal vez el mejor concebido en términos de una forma del principio de incertidumbre.

DEFINICIÓN 8 (Función Continua de Ambigüedad). Sea $f, g \in L^2(\mathbb{R})$. Entonces la función continua de ambigüedad de f , g se define como la aplicación $\mathcal{A}_{f,g} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{C}$ dada por

$$\mathcal{A}_{f,g}(t, \theta) = \int_{\mathbb{R}} f\left(x + \frac{t}{2}\right) \bar{g}\left(x - \frac{t}{2}\right) e^{2i\pi x\theta} dx.$$

Realizando un cambio de variable, la función de ambigüedad se puede reescribir como

$$\mathcal{A}_{f,g}(t, \theta) = e^{i\pi t\theta} \int_{\mathbb{R}} f(x+t) \bar{g}(x) e^{2i\pi x\theta} dx. \quad (5)$$

Existen muchas variantes de esta definición de ambigüedad, como el lector puede consultar en [1, 2, 11, 15, 20]. Pero todas estas variaciones coinciden a través de la función módulo de los números complejos. A nivel de implementación, Richman *et al.* [14] definen la función de ambigüedad de tal manera que admita una representación con dominio discreto y finito.

DEFINICIÓN 9 (Función Discreta de Ambigüedad). Sean $x, y \in l^2(\mathbb{Z}_N)$. Entonces la función discreta de ambigüedad de x, y , o simplemente DAF, se define como la aplicación $\mathcal{A}_{x,y} : \mathbb{Z}_N \times \mathbb{Z}_N \rightarrow \mathbb{C}$ dada por

$$\mathcal{A}_{x,y}[m, k] = \lambda_{m,k} \sum_{n \in \mathbb{Z}_N} x[\langle n + m \rangle_N] \bar{y}[n] \omega_N^{nk},$$

donde $\bar{y} \in l^2(\mathbb{Z}_N)$ es la señal conjugada de la señal y , y $\lambda_{m,k} \in \mathbb{C}$ es⁶

$$\lambda_{m,k} = \begin{cases} \omega_N^{qmk}, & \text{si } N \text{ es impar, donde } q \in \mathbb{Z}_N \text{ tal que } 2q \equiv 1 \pmod{N}, \\ \omega_N^{mk/2}, & \text{si } N \text{ es par, tal que } \langle mk \rangle_N < \frac{N}{2}, \\ \omega_N^{(mk-N)/2}, & \text{si } N \text{ es par, tal que } \langle mk \rangle_N \geq \frac{N}{2}. \end{cases}$$

Al ser la DAF una función de dos dimensiones, puede representarse como una matriz.

DEFINICIÓN 10 (Matriz DAF). Sea $x, y \in \mathbb{C}^N$. La matriz DAF se define como $A_{x,y} \in \mathbb{C}^{N \times N}$ con $(A_{x,y})_{m,k} = \mathcal{A}_{x,y}[m, k]$, para $m, k \in \mathbb{Z}_N$.

La matriz DAF se puede implementar a través de un algoritmo en paralelo, mediante una estructura que utiliza el producto de Kronecker y de Hadamard. El Teorema 5 explica los detalles de la representación matricial del algoritmo.

TEOREMA 5. Sea $x, y \in \mathbb{C}^N$, $A_{x,y} \in \mathbb{C}^{N \times N}$ la matriz DAF. Entonces

$$A_{x,y} = P \odot \mathcal{R}_{N,N} \{ L_N^{N^2} [I_N \otimes \overline{F_N}] [X \odot (\mathbf{1}_N \otimes \bar{y})] \}, \quad (6)$$

donde

1. $X \in \mathbb{C}^{N^2}$ es $X = \bigsqcup_{m \in \mathbb{Z}_N} \mathcal{S}_{-m}\{x\}$;
2. $P \in \mathbb{C}^{N \times N}$ cumple $P_{m,k} = \lambda_{m,k}$;
3. $\overline{F_N} \in \mathbb{C}^{N \times N}$ es la matriz conjugada de F_N .

DEMOSTRACIÓN. Sea $a = L_N^{N^2} [I_N \otimes \overline{F_N}] [X \odot (\mathbf{1}_N \otimes \bar{y})]$, donde $a \in \mathbb{C}^{N^2}$. Desarrollando el vector a , se obtiene

$$a = L_N^{N^2} \bigsqcup_{m \in \mathbb{Z}_N} H_m, \quad (7)$$

donde $H_m \in \mathbb{C}^N$ es $H_m = \overline{F_N}(\mathcal{S}_{-m}\{x\} \odot \bar{y})$. Aplicando el operador $\mathcal{R}_{N,N}$ a ambos lados de la ecuación (7) y utilizando la Proposición 1, se obtiene

$$\mathcal{R}_{N,N}\{a\} = \mathcal{R}_{N,N} \left\{ L_N^{N^2} \bigsqcup_{m \in \mathbb{Z}_N} H_m \right\} = \left(\mathcal{R}_{N,N} \left\{ \bigsqcup_{m \in \mathbb{Z}_N} H_m \right\} \right)^T.$$

⁶La representación en continuo de la función de ambigüedad, la ecuación (5), posee la fracción $\frac{1}{2}$ en la constante $e^{i\pi t\theta} = e^{2i\pi \frac{t\theta}{2}}$, al realizar el cambio de variable. Como $\frac{1}{2}$ no es un elemento de $\mathbb{Z}_N$, se consideran tres casos de la constante $e^{i\frac{2\pi}{N}\langle \frac{mk}{2} \rangle_N}$. Ver [14] para más detalles.

Sea $H \in \mathbb{C}^{N \times N}$ dada por

$$H = \mathcal{R}_{N,N} \left\{ \bigsqcup_{m \in \mathbb{Z}_N} H_m \right\},$$

donde $H = (H_0 \ H_1 \ \dots \ H_{N-1})$ y cada H_m representa una columna de la matriz H . Entonces

$$\mathcal{R}_{N,N}\{a_{x,y}\} = H_{m,k}^T = H_{k,m} = H_m[k] = \sum_{n \in \mathbb{Z}_N} x[\langle n+m \rangle_N] \bar{y}[n] \omega_N^{nk}.$$

Ahora, realizando el producto de Hadamard de P y H^T se obtiene

$$(P \odot H^T)_{m,k} = (P)_{m,k} \odot (H)_{k,m} = \lambda_{m,k} \sum_{n \in \mathbb{Z}_N} x[\langle n+m \rangle_N] \bar{y}[n] \omega_N^{nk} = A_{x,y}[m, k].$$

Por lo tanto, se concluye que la matriz DAF se puede obtener de la siguiente manera:

$$A_{x,y} = P \odot \mathcal{R}_{N,N}\{L_N^{N^2} [I_N \otimes \overline{F_N}] [X \odot (\mathbf{1}_N \otimes \bar{y})]\}. \quad \square$$

En el teorema anterior se puede apreciar que la matriz P es una matriz simétrica, ya que $\lambda_{m,k} = \lambda_{k,m}$. El Algoritmo 7 muestra la implementación de la ecuación (6). Los pasos 2-4 son independientes en cada iteración, lo que permite su cálculo en paralelo.

Algoritmo 7: Cálculo de la matriz DAF.

Entrada: $x, y \in \mathbb{C}^N$

Salida: $A_{x,y} \in \mathbb{C}^{N \times N}$

- 1: **para** $m = 0 : N - 1$ **hacer**
 - 2: $v \leftarrow x \odot \mathcal{S}_{-m}\{\bar{y}\}$
 - 3: $P_{aux} \leftarrow (\lambda_{m,0}, \lambda_{m,1}, \dots, \lambda_{m,N-1})^T$
 - 4: $A_{x,y}[:, m] \leftarrow P_{aux} \odot \mathcal{F}_v$
 - 5: **fin para**
 - 6: $A_{x,y} \leftarrow A_{x,y}^T$
-

4.3. DISTRIBUCIÓN DISCRETA DE COHEN

Una de las principales aplicaciones de la función de ambigüedad fue presentada por L. Cohen en el artículo «*Generalized Phase-Space Distribution Functions*» [6] y luego profundizada en su libro «*Time-frequency analysis: theory and applications*» [7]. Cohen desarrolló una representación general de cualquier función en tiempo-frecuencia continua, la cual denominó clase general. Cohen describe que toda representación bidimensional en tiempo-frecuencia, para $f, g \in L^2(\mathbb{R})$, puede ser obtenida a partir de la aplicación $\mathcal{C}_{f,g} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{C}$ definida como

$$\mathcal{C}_{f,g}(t, \theta) = \int_{\mathbb{R}} \int_{\mathbb{R}} \mathcal{A}_{f,g}(\tau, \eta) \phi(\tau, \eta) e^{-j2\pi(t\eta + \theta\tau)} d\eta d\tau, \quad (8)$$

donde $\mathcal{A}_{f,g}$ es la función continua de ambigüedad de f y g y $\phi \in L^2(\mathbb{R} \times \mathbb{R})$ es conocida como función núcleo. El Cuadro 1 muestran algunos ejemplos de las funciones núcleo más conocidas para el caso continuo explicado anteriormente, $\theta(\tau, \eta)$, y su representación discreta, $\theta[\tau, \eta]$.

Nombre	$\phi(\tau, \eta)$	$\phi[\tau, \eta]$
Wigner	1	1
Margenau-Hill	$\cos(\pi\tau\eta)$	$\cos(\pi\tau\eta)$
Kirwood-Rihanzek	$e^{-\pi\tau\eta}$	$e^{-\pi\tau\eta}$
Born-Jordan	$\frac{\text{sen}(\pi\tau\eta)}{\pi\tau\eta}$	$\frac{\text{sen}(\pi\tau\eta)}{\pi\tau\eta}$
Choi-Williams	$e^{-\alpha(\pi\tau\theta/2)^2}$	$e^{-\alpha(\pi\tau\theta/2)^2}$
Zhao-Atlas-Marks	$e^{-\alpha\tau^2} \tau \frac{\text{sen}(\alpha\pi\theta\tau/2)}{\alpha\pi\theta\tau/2}$	$e^{-\alpha\tau^2} \tau \frac{\text{sen}(\alpha\pi\theta\tau/2)}{\alpha\pi\theta\tau/2}$

Cuadro 1: Funciones núcleo: Caso continuo ($\theta(\tau, \eta)$) y discreto ($\theta[\tau, \eta]$).

La aplicación $\mathcal{C}_{f,g}$ también recibe el nombre de distribución de Cohen y puede ser vista como la transformada continua de Fourier en dos dimensiones de $\mathcal{A}_{f,g}\phi$, considerando que los parámetros del dominio están transpuestos. M. Richman *et al.* [14] presentaron una formulación discreta de la expresión dada en (8), utilizando la función núcleo de Wigner, es decir, $\theta[\tau, \eta] = 1$. Basado en el trabajo de Richman, se define una representación discreta y finita de la clase general definida por Cohen, utilizando la DAF, denominada distribución discreta de Cohen.

DEFINICIÓN 11 (Distribución Discreta de Cohen). Sean $x, y \in l^2(\mathbb{Z}_N)$. La distribución discreta de Cohen de x, y , o simplemente DCD, se define como la aplicación $\mathcal{C}_{x,y} : \mathbb{Z}_N \times \mathbb{Z}_N \rightarrow \mathbb{C}$ dada por

$$\mathcal{C}_{x,y}[m, k] = \sum_{\tau \in \mathbb{Z}_N} \sum_{\eta \in \mathbb{Z}_N} \mathcal{A}_{x,y}[\tau, \eta] \phi[\tau, \eta] \omega_N^{-(m\eta + k\tau)},$$

donde $\phi \in \mathbb{C}^N$ es la función núcleo y $\mathcal{A}_{x,y}$ es la DAF de x, y .

Como en el caso de la DAF y debido a su estructura bidimensional, $\mathcal{C}_{x,y}$ puede ser también representada como una matriz.

DEFINICIÓN 12 (Matriz DCD). Sean $x, y \in \mathbb{C}^N$. La matriz DCD se define como $\mathcal{C}_{x,y} \in \mathbb{C}^{N \times N}$ con $(\mathcal{C}_{x,y})_{m,k} = \mathcal{C}_{x,y}[m, k]$, para $m, k \in \mathbb{Z}_N$.

Al igual que en el caso continuo, $\mathcal{C}_{x,y}$ puede ser vista como la transformada discreta de Fourier en dos dimensiones de $\mathcal{A}_{x,y} \odot \phi$, considerando que los parámetros del dominio están transpuestos, es decir:

$$\mathcal{C}_{x,y}[m, k] = \mathcal{F}_{\mathcal{A}_{x,y} \odot \phi}^{2D}[k, m].$$

Utilizando la representación matricial de la 2D-DFT, definida en la ecuación (4), la matriz DCD se puede expresar como

$$\mathcal{C}_{x,y} = \mathcal{R}_{N,N} \{ (F_N \otimes F_N) \mathcal{V} \{ (\mathcal{A}_{x,y} \odot \phi)^T \} \}, \quad (9)$$

donde $\phi \in \mathbb{C}^{N \times N}$ es la representación matricial de la función núcleo. Ahora, consideremos lo siguiente:

1. Por la Proposición 1,

$$\mathcal{V}\{(A_{x,y} \odot \phi)^T\} = L_N^{N^2} \mathcal{V}\{(A_{x,y} \odot \phi)\}.$$

Por tanto, aplicando esta sustitución en la ecuación (9) se obtiene

$$C_{x,y} = \mathcal{R}_{N,N}\{(F_N \otimes F_N)L_N^{N^2} \mathcal{V}\{(A_{x,y} \odot \phi)\}\}. \quad (10)$$

2. Por el Teorema 1,

$$F_N \otimes F_N = (I_N \otimes F_N)(F_N \otimes I_N).$$

Por tanto, aplicando esta sustitución en la ecuación (10) se obtiene

$$C_{x,y} = \mathcal{R}_{N,N}\{(I_N \otimes F_N)(F_N \otimes I_N)L_N^{N^2} \mathcal{V}\{(A_{x,y} \odot \phi)\}\}. \quad (11)$$

3. Por el Teorema 2,

$$F_N \otimes I_N = L_N^{N^2} (I_N \otimes F_N) L_N^{N^2}.$$

Por tanto, aplicando esta sustitución en la ecuación (11) se obtiene

$$C_{x,y} = \mathcal{R}_{N,N}\{(I_N \otimes F_N)L_N^{N^2} (I_N \otimes F_N)L_N^{N^2} \mathcal{V}\{(A_{x,y} \odot \phi)\}\}. \quad (12)$$

4. De la ecuación (2),

$$L_N^{N^2} L_N^{N^2} = I_{N^2},$$

y aplicando esta igualdad en la ecuación (12) se obtiene

$$C_{x,y} = \mathcal{R}_{N,N}\{(I_N \otimes F_N)L_N^{N^2} (I_N \otimes F_N)\mathcal{V}\{(A_{x,y} \odot \phi)\}\}.$$

Esto permite concluir el siguiente teorema:

TEOREMA 6. Sean $x, y \in \mathbb{C}^N$; $C_{x,y} \in \mathbb{C}^{N \times N}$ la matriz DCD de x, y ; $A_{x,y} \in \mathbb{C}^{N \times N}$ la matriz DAF de x, y ; $\phi \in \mathbb{C}^{N \times N}$ la matriz que representa a la función núcleo. Entonces

$$C_{x,y} = \mathcal{R}_{N,N} \left\{ (I_N \otimes F_N) L_N^{N^2} (I_N \otimes F_N) \mathcal{V}\{(A_{x,y} \odot \phi)\} \right\}. \quad (13)$$

El Algoritmo 8 muestra la implementación de la ecuación (13).

En el citado Algoritmo 8 se puede observar que los pasos 3–4 y 8–9, que representan el cálculo de la DFT, son independientes en cada iteración, por lo que se pueden implementar en paralelo.

Algoritmo 8: Cálculo de la matriz DCD.**Entrada:** $A_{x,y}, \phi \in \mathbb{C}^{N \times N}$ **Salida:** $C_{x,y} \in \mathbb{C}^{N \times N}$

1: $C_{x,y} \leftarrow A_{x,y} \odot \phi$	6: $C_{x,y} \leftarrow C_{x,y}^T$
2: para $n = 0 : N - 1$ hacer	7: para $n = 0 : N - 1$ hacer
3: $y_{aux} \leftarrow C_{x,y}[:, n]$	8: $y_{aux} \leftarrow C_{x,y}[:, n]$
4: $C_{x,y}[:, n] = \mathcal{F}_{y_{aux}}$	9: $C_{x,y}[:, n] = \mathcal{F}_{y_{aux}}$
5: fin para	10: fin para

5. CONCLUSIONES

El presente artículo desarrolla la representación matricial de tres transformadas discretas armónicas: la transformada discreta de Fourier, la función discreta de ambigüedad y la distribución discreta de Cohen. Tal desarrollo matricial se realiza utilizando un conjunto de operadores matriciales que permiten el diseño e implementación de algoritmos en paralelo (el producto de Hadamard, el producto de Kronecker, la suma directa, el operador de acumulación, el operador *vec* y su inverso) así como introduciendo tres operadores de permutación: el operador de permutación de orden N y paso R , el operador de traslación y el operador de orden exponencial.

También se presentan dos nuevos resultados, los Teoremas 5 y 6, relacionados con las matrices DAF y DCD, respectivamente. Además, se muestra el pseudo-código de los algoritmos desarrollados, implementados en MATLAB®, utilizando dos *toolbox* que permiten el cálculo en paralelo: *pMatlab Toolbox* ©, desarrollado por el Laboratorio Lincoln del MIT, y *Parallel Computing Toolbox* ©, diseñado por Mathworks.

Un ejemplo de la implementación aparece en la Figura 1, donde se muestra la gráfica del módulo de la DAF de la señal $u \in l^2(\mathbb{Z}_{100})$, la cual es una señal de Wiener,⁷ y la gráfica del módulo de la DCD de la señal $v \in l^2(\mathbb{Z}_{101})$, la cual es una señal de Björck,⁸ utilizando la función núcleo de Wigner. Para cada uno de los casos se utiliza la misma señal (la de Wiener y de Björck) como su respectiva señal eco. En la Figura 1 (a), el comportamiento de la DAF de una señal de Wiener muestra que, si $m + k \equiv 0 \pmod{100}$, entonces la DAF alcanza su máximo valor; en caso contrario el resultado es cero. En la Figura 1 (b), el comportamiento de la DCD de una señal

⁷Sea $M = N$, si N impar, o $M = 2N$, si N es par. Una *señal de Wiener* se define como la señal $u \in l^2(\mathbb{Z}_N)$ con $u[n] = e^{2i\pi kn^2/M}$, donde k y M son primos relativos.

⁸Sea N un número primo tal que $N \equiv 1 \pmod{4}$. Una *señal de Björck* se define como la señal $v \in l^2(\mathbb{Z}_N)$ con $v[n] = e^{2i\pi\theta(\frac{n}{N})}$, donde $\theta = \arccos\left(\frac{1}{1+\sqrt{N}}\right)$ y $\left(\frac{n}{N}\right)$ es el símbolo de Legendre:

$$\left(\frac{n}{N}\right) = \begin{cases} 0, & \text{si } m \equiv 0 \pmod{N}, \\ 1, & \text{si } m \text{ es un cuadrado mód } N, \\ -1, & \text{si } m \text{ no es un cuadrado mód } N. \end{cases}$$

de Björck muestra que su máximo valor lo alcanza cuando $m = k = 0$. Véase [4] para más detalles sobre las señales de Wiener y de Björck.

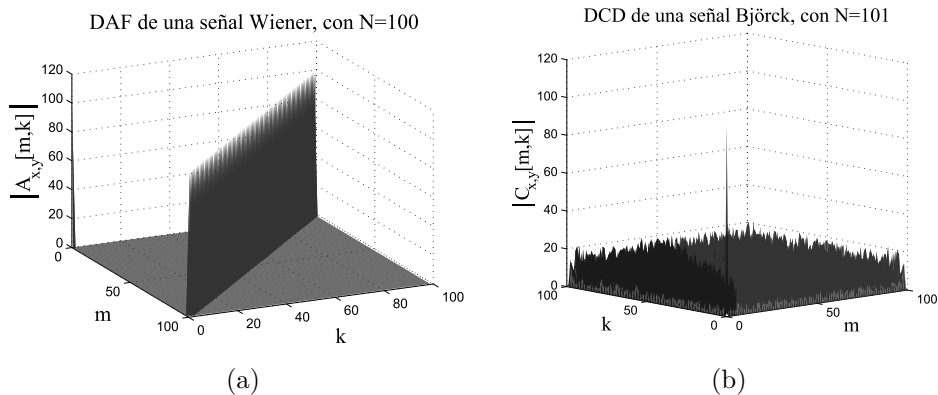


Figura 1: Ejemplo de gráficas de la DAF (a) y la DCD (b).

AGRADECIMIENTOS

Los autores quieren agradecer al profesor Filánder Sequeira (Universidad Nacional, Sede Central, Costa Rica), al estudiante de maestría Edwin Anaya y al estudiante doctoral Juan Valera (Universidad de Puerto Rico, Recinto Universitario de Mayagüez, Puerto Rico) su valiosa lectura del documento.

REFERENCIAS

- [1] L. AUSLANDER Y R. TOLIMIERI, Computing decimated finite cross-ambiguity functions, *IEEE Transactions on Acoustics, Speech and Signal Processing* (1988), Vol. 36, 359–364.
- [2] J. J. BENEDETTO Y J. J. DONATELLI, Ambiguity Function and Frame-Theoretic Properties of Periodic Zero-Autocorrelation Waveforms, *IEEE Journal of Selected Topics in Signal Processing* (June, 2007), Vol. I, 6–20.
- [3] J. J. BENEDETTO Y J. J. DONATELLI, Frames and a vector-valued ambiguity function, *42nd Asilomar Conference on Signals, Systems and Computers* (October, 2008), 8–12.
- [4] J. J. BENEDETTO, I. KONSTANTINIDIS Y M. RANGASWAMY, Phase-Coded Waveforms and Their Design, *IEEE Signal Processing Magazine* (January, 2009), 22–31.
- [5] A. BONAMI, G. GARRIGOS Y P. JAMING, Discrete radar ambiguity problems, *Applied and Computational Harmonic Analysis* (2007), Vol. 23, 388–414.
- [6] L. COHEN, Generalized Phase-Space Distribution Functions, *Journal of Mathematical Physics* (2000), Vol. 7, 781–786.

- [7] L. COHEN, *Time-frequency analysis: theory and applications*, Prentice-Hall, Inc. (1995).
- [8] J. W. COOLEY Y J. W. TUKEY, An Algorithm for the Machine Calculation of Complex Fourier Series, *Mathematics of Computation* (June, 1965), Vol. 19, 297–301.
- [9] R. A. HORN Y C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press (1991).
- [10] J. R. JOHNSON, R. W. JOHNSON, D. RODRIGUEZ Y R. TOLIMIERI, A methodology for designing, modifying, and implementing Fourier transform algorithms on various architectures, *Circuits, Systems, and Signal Processing* (1990), Vol. 9, 449–500.
- [11] A. KEBE, I. KONSTANTINIDIS, J. J. BENEDETTO, M. R. DELLOMO Y J. M. SIERACKI, Ambiguity and sidelobe behavior of CAZAC coded waveforms, *Radar Conference, 2007 IEEE* (April, 2007), 99–103.
- [12] P. KUMHOM, J. R. JOHNSON Y P. NAGVAJARA, Design, optimization, and implementation of a universal FFT processor, *ASIC/SOC Conference, 2000, Proceedings 13th Annual IEEE International* (2000), 182–186.
- [13] P. A. REGALIA Y S. K. MITRA, Kronecker products, unitary matrices, and signal processing applications, *SIAM Rev.* (December, 1989), 586–613.
- [14] M. S. RICHMAN, T. W. PARKS Y R. G. SHENOY, Discrete-time, discrete-frequency, time-frequency analysis, *IEEE Transactions on Signal Processing* (June, 1998), Vol. 46, 1517–1527.
- [15] D. RODRÍGUEZ, J. SEGUEL Y E. CRUZ, Algebraic methods for the analysis and design of time-frequency signal processing algorithms, *IEEE International Symposium on Circuits and Systems* (May, 1993), Vol. 1, 3–6.
- [16] G. SAN ANTONIO, D. R. FUHRMANN Y F. C. ROBEY, MIMO Radar Ambiguity Functions, *IEEE Journal of Selected Topics in Signal Processing* (June, 2007), Vol. I, 167–177.
- [17] J. P. SOTO QUIRÓS Y D. RODRÍGUEZ, Un nuevo enfoque a la criptografía matemática usando la función discreta de ambigüedad, *Revista de Matemática: Teoría y Aplicaciones, Universidad de Costa Rica*, enviado para su publicación (2012).
- [18] R. TOLIMIERI, M. AN Y C. LU, *Algorithms for Discrete Fourier Transform and Convolution*, Springer (1997).
- [19] R. TOLIMIERI, M. AN, C. LU Y C. S. BURRUS, *Mathematics of multidimensional Fourier transform algorithms*, Springer (1997).
- [20] R. TOLIMIERI Y S. WINOGRAD, Computing the ambiguity surface, *IEEE Transactions on Acoustics, Speech and Signal Processing* (October, 1985), Vol. 33, No. 5, 1239–1245.
- [21] C. F. VAN LOAN, *Computational Frameworks for the Fast Fourier Transform*, SIAM (1992).
- [22] D. F. WALNUT, *An Introduction to Wavelet Analysis*, Applied and Numerical Harmonic Analysis, Springer (2002).

- [23] S. WINOGRAD, On Computing the Discrete Fourier Transform, *Mathematics of Computation* (1978), Vol. 32, 175–199.
- [24] P. M. WOODWARD, *Probability and Information Theory, with Applications to Radar*, McGraw-Hill (1953).

JUAN PABLO SOTO QUIRÓS, ESCUELA DE MATEMÁTICAS, UNIVERSIDAD DE COSTA RICA, SAN JOSÉ, COSTA RICA

Correo electrónico: juan.sotoquiros@ucr.ac.cr

DOMINGO RODRÍGUEZ, DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y COMPUTADORAS, UNIVERSIDAD DE PUERTO RICO, RECINTO UNIVERSITARIO DE MAYAGÜEZ, MAYAGÜEZ, PR, 00681, USA

Correo electrónico: domingo@ece.uprm.edu