Transformada Rápida de Fourier Otimizada

G. Jerônimo da Silva Jr. e R. M. Campello de Souza

***Resumo*— Este artigo introduz o teorema da complexidade multiplicativa para transformadas e o aplica, em conjunto com a teoria de bases ciclotômicas, para construir a transformada rápida de Fourier (FFT) otimizada, isto é, aquela que apresenta complexidade multiplicativa mínima. O algoritmo também pode ser utilizado para computar um conjunto qualquer de componentes de forma otimizada.**

***Palavras-Chave*— Transformada Rápida de Fourier, bases ciclotômicas, matriz postunitária, complexidade multiplicativa.**

***Abstract*— This paper introduces the multiplicative complexity theorem for transforms and applies it, jointly with the cyclotomic basis theory, to construct an optimized fast Fourier transform implementation, i. e., an FFT with minimum multiplicative complexity. The algorithm can also be used to compute any set of components in an optimum way.**

***Keywords*— Fast Fourier transform, cyclotomic basis, rank-one matrix, multiplicative complexity.**

I. Introdução

OCORREU um grande avanço na área de processamento digital de sinais quando J. W. Cooley e J. W. Tukey apresentaram, em 1965, a transformada rápida de fourier (FFT)[1], um algoritmo capaz de computar a transformada discreta de Fourier (DFT) com uma complexidade aritmética menor em relação ao método direto. Em 1978, S. Winograd publica o algoritmo que hoje é conhecido como a FFT de Winograd [2], além de publicar diversos trabalhos sobre a complexidade multiplicativa [3]. Este trabalho se destaca por apresentar uma complexidade ainda menor que a FFT de Cooley-Tukey. Em 1988, M. T. Heideman publica um trabalho sobre a complexidade multiplicativa na computação da DFT [4], e mostra que as complexidades mínimas para os comprimentos $N = 3, 5, 7, 9, 10$ estavam abaixo da complexidade multiplicativa da mais eficiente FFT existente na época, a FFT de Winograd. Até recentemente, não tinha sido apresentado um único algoritmo com complexidade multiplicativa inferior a esta FFT. Somente em 2010, G. J. da Silva Jr. e R. M. Campello de Souza publicaram a primeira FFT de comprimento 3 que atinge a complexidade multiplicativa mínima estabelecida por Heideman. O artigo se baseia numa decomposição para o núcleo da transformada de Fourier denominada *decomposição em bases ciclotômicas* [5], a qual é utilizada neste artigo para obter um algoritmo FFT otimizado que atinge a complexidade multiplicativa mínima de Heideman para qualquer comprimento; além disso, o algoritmo apresentado neste artigo pode computar qualquer conjunto de componentes de forma otimizada.

G. J. da Silva Jr. e R. M. Campello de Souza, Grupo de Processamento de Sinais, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, PE, E-mails: gilsonjr@gmail.com, ricardo@ufpe.br. Este trabalho foi parcialmente financiado pelo CNPq (140304/2009-6).

A transformada discreta de Fourier de uma sequência (v_n), $n = 0, 1, \ldots, N-1$, é a sequência (V_k), $k = 0, 1, \ldots, N-1$, em que

$$V_k = \sum_{n=0}^{N-1} v_n W_N^{nk}, \tag{1}$$

em que $W_N \triangleq e^{-j\frac{2\pi}{N}}$ é um elemento de ordem N sobre $\mathbb{C}$ e $j \triangleq \sqrt{-1}$. Definindo-se os vetores

$$V \triangleq \begin{bmatrix} V_0 & V_1 & \ldots & V_{N-1} \end{bmatrix}^T, \tag{2}$$

$$v \triangleq \begin{bmatrix} v_0 & v_1 & \ldots & v_{N-1} \end{bmatrix}^T, \tag{3}$$

e a matriz de transformação

$$W \triangleq \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}, \tag{4}$$

pode-se então representar a transformada discreta de Fourier pela equação matricial

$$V = Wv. \tag{5}$$

É possível decompor o núcleo da transformada em bases ciclotômicas, isto é

$$W_N^k = \sum_{i=1}^{L} a_{ki}\gamma_i, \tag{6}$$

em que $a_{ki} \in \mathbb{Q}$ e γ_i, $i = 1, 2, \ldots, L$, é a base ciclotômica [5]. Com isso, a equação matricial (5) torna-se

$$V = \sum_{i=1}^{L} \gamma_i A_i v, \tag{7}$$

em que a componente da linha k e coluna l da matriz A_i é o coeficiente racional da componente γ_i em W_N^{kl}. É possível demonstrar que o número de multiplicações reais, M_r, para se computar $\gamma_i A_i v$ é igual ao posto de A_i [5].

II. Teorema da Complexidade Multiplicativa para Transformadas

Toda matriz pode ser decomposta em matrizes de posto unitário (matrizes postunitárias) [6]. Considere agora uma decomposição ótima para as matrizes A_i em matrizes postunitárias, isto é, A_i sendo escrito na forma

$$A_i = \sum_{j=1}^{M_r} c_{ij} K_j, \tag{8}$$

para $i = 1, 2, \ldots, L$, em que K_j denota matrizes postunitárias, e não podendo ser escrita por nenhuma outra combinação linear de um conjunto de matrizes postunitárias de cardinalidade menor que M_r. Com isto, a Equação (7) torna-se

$$V = \sum_{i=1}^{L} \gamma_i \sum_{j=1}^{M_r} c_{ij} K_j v, \tag{9}$$

$$V = \sum_{j=1}^{M_r} K_j v \sum_{i=1}^{L} c_{ij} \gamma_i; \tag{10}$$

definindo β_j como

$$\beta_j \triangleq \sum_{i=1}^{L} c_{ij} \gamma_i, \tag{11}$$

a expressão (10) torna-se

$$V = \sum_{j=1}^{M_r} \beta_j K_j v. \tag{12}$$

Desde que todos os K_j são matrizes postunitárias, o número de multiplicações reais necessárias para computar V é M_r.

Teorema 1: Todo algoritmo de transformada $V = \Psi v$ escrito da forma $V = CBAv$, isto é, através da decomposição $\Psi = CBA$, em que C e A são matrizes de racionais, e B é uma matriz diagonal com M_r componentes que não pertencem aos racionais, pode ser escrito na forma

$$V = (\Psi_0 + \sum_{j=1}^{M_r} \beta_j K_j) v, \tag{13}$$

em que $\beta_j \notin \mathbb{Q}$, Ψ_0 é uma matriz de racionais e K_j são matrizes de racionais postunitárias.

Demonstração: Sendo C_i as colunas de C, b_i os elementos de B e A_i^T as linhas de A, então

$$\begin{bmatrix} C_1 & \cdots & C_N \end{bmatrix} \begin{bmatrix} b_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_N \end{bmatrix} \begin{bmatrix} A_1^T \\ \vdots \\ A_N^T \end{bmatrix} = CBA, \tag{14}$$

$$CBA = \sum_{j=1}^{N} b_j C_j A_j^T. \tag{15}$$

As matrizes $C_j A_j^T$ são postunitárias, chamando os M_r valores de $b_j \notin \mathbb{Q}$ de β_i, podemos escrever

$$CBA = (\sum_{b_j \in \mathbb{Q}} b_j C_j A_j^T) + \sum_{i=1}^{M_r} \beta_i K_i, \tag{16}$$

em que K_i é o respectivo $C_j A_j^T$ para $b_j \notin \mathbb{Q}$. ■

Desde que $\Psi_0 v$ não apresenta multiplicações, a complexidade multiplicativa do algoritmo está nas operações $\beta_j K_j v$, e desde que todas as matrizes K_j são postunitárias, existem exatamente M_r multiplicações. O próximo teorema estabelece uma relação se e somente se entre a complexidade do algoritmo e a decomposição em matrizes postunitárias das matrizes A_i.

Teorema 2 (Teorema da complexidade multiplicativa): Seja

$$\Psi = \Psi_0 + \sum_{i=1}^{L} \gamma_i A_i \tag{17}$$

a matriz de uma transformada em que $\gamma_i \notin \mathbb{Q}$ são elementos da base ciclotômica que gera o núcleo da transformada, a matriz Ψ_0 é uma matriz de gaussianos e as matrizes A_i, $i = 1, \ldots, L$ são matrizes de racionais. Existe um algoritmo para computar $V = \Psi v$ com M_r multiplicações se, e somente se, existe uma decomposição das matrizes A_i como combinação linear de M_r matrizes postunitárias.

Demonstração: Supondo que existe um algoritmo com M_r multiplicações, então, pelo Teorema 1,

$$\Psi = \Psi_0' + \sum_{j=1}^{M_r} \beta_j K_j. \tag{18}$$

Mas todos os β_j podem ser escritos como combinação linear das base ciclotômicas γ_i por (11), então

$$\Psi = \Psi_0' + \sum_{j=1}^{M_r} \sum_{i=1}^{L} c_{ij} \gamma_i K_j = \Psi_0' + \sum_{i=1}^{L} \gamma_i \sum_{j=1}^{M_r} c_{ij} K_j, \tag{19}$$

usando (17) no primeiro membro da equação, chega-se a

$$\Psi_0 + \sum_{i=1}^{L} \gamma_i A_i = \Psi_0' + \sum_{i=1}^{L} \gamma_i \sum_{j=1}^{M_r} c_{ij} K_j, \tag{20}$$

então

$$(\Psi_0 - \Psi_0') + \sum_{i=1}^{L} \gamma_i (A_i - \sum_{j=1}^{M_r} c_{ij} K_j) = O, \tag{21}$$

em que O é uma matriz nula. Neste caso, como os γ_i são linearmente independentes sobre os racionais, a única solução para a equação matricial é que todas as matrizes devem ser nulas, isto é

$$(\Psi_0 - \Psi_0') = (A_i - \sum_{j=1}^{M_r} c_{ij} K_j) = O, \tag{22}$$

o que implica que existe uma decomposição em matrizes postunitárias para todos os A_i. Por outro lado, suponha que exista uma decomposição de A_i em M_r matrizes postunitárias como em (8). Substituindo a decomposição em (17), tem-se

$$\Psi = \Psi_0 + \sum_{j=1}^{M_r} K_j (\sum_{i=1}^{L} c_{ij} \gamma_i), \tag{23}$$

portanto

$$V = \Psi v = \Psi_0 v + \sum_{j=1}^{M_r} (\sum_{i=1}^{L} c_{ij} \gamma_i) K_j v, \tag{24}$$

que é computado com exatamente M_r multiplicações pelo fato de que as matrizes K_j são postunitárias. ■

O Teorema 2 transforma o problema da construção de um algoritmo ótimo em um problema de decomposição em matrizes postunitárias. Desde que Ψ pode ser qualquer matriz, é possível construir algoritmos para qualquer conjunto de componentes. A próxima seção mostra como resolver o problema da decomposição em matrizes postunitárias.

III. Decomposição Ortogonal Postunitária e Solução Ótima

Considere a decomposição para todas as matrizes A_i dada por

$$A_i = d_i U_l, \tag{25}$$

para $i = 1, \ldots, L$ em que U_l é uma matriz com todas as linhas ortogonais. A matriz U_l é obtida utilizando-se o processo de ortogonalização de Gram-Schmidt [6], sem ortonormalizar as matrizes. Assim

$$U_l = \begin{bmatrix} u_{l_1}^T \\ u_{l_2}^T \\ \vdots \\ u_{l_r}^T \end{bmatrix}. \tag{26}$$

Dessa forma, as linhas de U_l formam uma base ortogonal para as linhas de todas as matrizes A_i. Então a componente da linha m e coluna n de d_i pode ser computada por

$$d_{m,n,i} = \frac{\langle A_{m,i}, u_{l_n} \rangle}{\langle u_{l_n}, u_{l_n} \rangle}, \tag{27}$$

em que $A_{m,i}^T$ é a linha m da matriz A_i, isto é

$$A_i = \begin{bmatrix} A_{1,i}^T \\ A_{2,i}^T \\ \vdots \\ A_{M,i}^T \end{bmatrix}. \tag{28}$$

Assim, define-se um processo de fatoração para as matrizes A_i chamado de fatoração dU, que será denotado na forma

$$(d_i, U_l) = \Gamma_{dU}(A_i). \tag{29}$$

Considere agora um segundo processo de fatoração sobre as colunas de d_i. O resultado desse processo é uma fatoração da forma

$$A_i = U_c D_i U_l, \tag{30}$$

em que U_c é uma matriz de colunas ortogonais e U_l é uma matriz de linhas ortogonais. Esse processo que gera, a partir das matrizes A_i, as matrizes D_i, U_c e U_l é denotado por

$$(U_c, D_i, U_l) = \Gamma_{CDL}(A_i). \tag{31}$$

O processo de fatoração $\Gamma_{CDL}(A_i)$ consiste em computar d_i e U_l a partir de (29), e então fazer

$$(D_i^T, U_c^T) = \Gamma_{dU}(d_i^T), \tag{32}$$

isto é, aplicar o processo Γ_{dU} sobre as colunas de d_i. O processo de fatoração Γ_{CDL} pode ser visto como uma transformada matricial ou simplesmente como uma decomposição em matrizes postunitárias ortogonais. Algumas propriedades podem ser verificadas a partir de (30), e. g., a linearidade

$$b_i A_i + b_j A_j = U_c (b_i D_i + b_j D_j) U_l. \tag{33}$$

Se

$$U_c = \begin{bmatrix} u_{c_1} & u_{c_2} & \ldots & u_{c_r} \end{bmatrix}, \tag{34}$$

e a componente da linha m e coluna n da matriz D_i, $D_{m,n,i} = 1$, é a única componente não nula, então a matriz A_i é dada por

$$A_i = u_{c_m} u_{l_n}^T, \tag{35}$$

e é uma matriz postunitária. Utilizando a linearidade, pode-se escrever que

$$A_i = \sum_{m=1}^{r} \sum_{n=1}^{r} D_{m,n,i} u_{c_m} u_{l_n}^T, \tag{36}$$

o que é uma decomposição de A_i em matrizes postunitárias ortogonais, visto que $\langle u_{c_m} u_{l_n}^T, u_{c_i} u_{l_j}^T \rangle$ é diferente de zero apenas se $m = i$ e $n = j$, isto é, se as matrizes são as mesmas.

Uma outra propriedade é que

$$\text{posto}(A_i) = \text{posto}(D_i), \tag{37}$$

o que significa que A_i é postunitária se, e somente se, D_i é postunitária.

A vantagem de se utilizar o processo de fatoração Γ_{CDL} é que o problema passa a ser a fatoração das matrizes D_i, no lugar das matrizes A_i, em matrizes postunitárias, que geralmente é um problema mais simples. Esse processo de fatoração é em si uma decomposição em matrizes postunitárias ortogonais, entretanto, a solução ótima para o processo nem sempre é composta de matrizes postunitárias ortogonais. Para ilustrar este fato considere o exemplo a seguir.

Exemplo 1: Considere o problema de decompor as matrizes

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

e

$$A_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

em matrizes postunitárias. Neste caso, $U_c = U_l = I_2$, as matrizes podem ser decompostas em

$$A_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix},$$

que é uma solução formada por 4 matrizes postunitárias ortogonais. Entretanto, considere as matrizes

$$K_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

e

$$A_2 + K_1 - K_2 = K_3 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}.$$

A matriz K_3 é postunitária, sendo assim pode-se escrever

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix},$$

que é uma decomposição melhor que a decomposição ortogonal, pois gera as matrizes com apenas 3 matrizes postunitárias.

IV. TRANSFORMADA RÁPIDA DE FOURIER OTIMIZADA

Para se obter um algoritmo ótimo para computar uma transformada discreta de Fourier de comprimento N, utilizam-se as bases ciclotômicas cos-CB, para N múltiplo de 4, ou sen/cos-CB, para outros valores de N [5]. Seja W a matriz de transformação da DFT de N pontos. Então, para N múltiplo de 4 tem-se

$$W = (A_1 - jA_2) + \sum_{i=3}^{\phi(N)} \gamma_i A_i, \tag{38}$$

ou, para valores de N não múltiplos de 4,

$$W = A_1 + \sum_{i=2}^{\phi(N)} \gamma_i A_i. \tag{39}$$

O Teorema da complexidade multiplicativa afirma que o algoritmo ótimo consiste em fatorar as matrizes A_i referentes aos $\gamma_i \notin \mathbb{Q}$ no processo Γ_{CDL} da melhor forma possível, isto é, obter as matrizes U_c, D_is e U_l, e encontrar a decomposição ótima de D_i em matrizes postunitárias, da seguinte forma

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_L \end{bmatrix} = c \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_{M_r} \end{bmatrix}, \tag{40}$$

em que a matriz c, $L \times M_r$, é formada pelos elementos c_{ij} de (8). Pela linearidade

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_L \end{bmatrix} = c \begin{bmatrix} U_c K_1 U_l \\ U_c K_2 U_l \\ \vdots \\ U_c K_{M_r} U_l \end{bmatrix}. \tag{41}$$

A partir de (11), pode se escrever que

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{M_r} \end{bmatrix} = c^T \begin{bmatrix} \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{\phi(N)} \end{bmatrix}, \tag{42}$$

e então, finalmente,

$$W = W_0 + \sum_{i=1}^{M_r} \beta_i U_c K_i U_l. \tag{43}$$

Utilizando o Teorema 1, pode-se colocar W na fatoração $W_0 + CBA$, em que C e A são matrizes de elementos racionais e B uma matriz diagonal, aplicando-se o processo Γ_{dU} sobre as matrizes $U_c K_i U_l$. Dessa forma se

$$C_i A_i^T = U_c K_i U_l, \tag{44}$$

então, utilizando (14) e (15),

$$C = \begin{bmatrix} C_1 & \cdots & C_{M_r} \end{bmatrix}, \tag{45}$$

$$B = \begin{bmatrix} \beta_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \beta_{M_r} \end{bmatrix} \tag{46}$$

e

$$A = \begin{bmatrix} A_1^T \\ \vdots \\ A_{M_r}^T \end{bmatrix}, \tag{47}$$

o que resulta em

$$W = W_0 + CBA, \tag{48}$$

e o algoritmo é implementado por

$$V = W_0 v + CBAv. \tag{49}$$

Desde que $W_0 v$ não contém multiplicações, a complexidade multiplicativa do algoritmo está em $CBAv$ que contém M_r multiplicações. A matriz W_0 pode ser ainda anexada nas matrizes CBA como no algoritmo de Winograd [7].

Exemplo 2: Para obter uma transformada ótima para $N = 5$, utiliza-se a base ciclotômica sen/cos-CB para escrever a matriz de transformação como

$$W = A_1 + \gamma_2 A_2 + \gamma_3 A_3 + \gamma_4 A_4,$$

em que, com $t = 2\pi/5$,

$$\begin{bmatrix} \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} = \begin{bmatrix} -j\text{sen}(t) \\ \cos(t) \\ -j\text{sen}(2t) \end{bmatrix},$$

e

$$A_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & -0.5 & -0.5 & 0 \\ 1 & -0.5 & 0 & 0 & -0.5 \\ 1 & -0.5 & 0 & 0 & -0.5 \\ 1 & 0 & -0.5 & -0.5 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & -1 & 1 & 1 & -1 \\ 0 & -1 & 1 & 1 & -1 \\ 0 & 1 & -1 & -1 & 1 \end{bmatrix}$$

e

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}.$$

Em seguida utiliza-se o processo Γ_{CDL} sobre as matrizes A_2, A_3 e A_4, pois é necessário a decomposição das mesmas em matrizes postunitárias. Com isso, tem-se

$$U_c = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix},$$

$$U_l = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & -1 & 1 \end{bmatrix},$$

$$D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

e

$$D_4 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Observa-se que as matrizes D_i são mais simples para a fatoração. Note ainda que D_2 e D_4 podem ser decompostas como no Exemplo 1, logo

$$\begin{bmatrix} D_2 \\ D_3 \\ D_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix},$$

em que

$$K_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$K_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

e

$$K_4 = \begin{bmatrix} 1 & 1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Sendo

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -j\text{sen}(t) \\ \cos(t) \\ -j\text{sen}(2t) \end{bmatrix},$$

pode-se escrever W como

$$W = A_1 + \sum_{i=1}^{4} \beta_i U_c K_i U_l,$$

ou na forma

$$W = A_1 + CBA,$$

em que

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & -1 & -1 & -1 \\ -1 & 0 & 1 & -1 \end{bmatrix}$$

e

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix}.$$

O algoritmo é obtido por

$$V = A_1 v + CBAv.$$

Até onde os autores conhecem, este é o primeiro algoritmo proposto na literatura que computa uma DFT de 5 pontos com apenas 4 multiplicações reais.

De forma semelhante, obtêm-se os algoritmos ótimos para os comprimentos $N = 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 24$. Os algoritmos para $N = 5, 7, 9, 10$ superam os melhores algoritmos atuais em termos de complexidade multiplicativa. A Tabela I compara a complexidade de alguns algoritmos bem conhecidos na literatura.

TABELA I

COMPLEXIDADE MULTIPLICATIVA REAL DA: OFFT - A TRANSFORMADA DE FOURIER OTIMIZADA; HMMC - COMPLEXIDADE MULTIPLICATIVA MÍNIMA PARA FFT DE HEIDEMAN; CT/GT - COMBINAÇÃO DA FFT DE COOLEY-TUKEY E GOOD-THOMAS OTIMIZADA; SW-FFT - A FFT DE WINOGRAD

N	(OFFT)	(HMMC)	(CT/GT)	(SW-FFT)
3	1	1	-	2
4	0	0	0	0
5	4	4	-	5
6	2	2	16	-
7	7	7	-	8
8	2	2	4	2
9	8	8	60	10
10	8	8	64	-
12	4	4	32	-
16	10	10	20	10
24	12	12	108	-

V. CONCLUSÕES

O teorema da complexidade multiplicativa para transformadas foi introduzido e utilizado para derivar algoritmos FFT ótimos, i.e., com complexidade multiplicativa mínima, para diversos comprimentos. Um método para decomposição de um conjunto de matrizes em matrizes postunitárias ortogonais foi introduzido e utilizado para encontrar decomposições ótimas.

AGRADECIMENTOS

Os autores agradecem ao Prof. Dr. Hélio M. de Oliveira por suas valorosas contribuições ao presente trabalho. O primeiro autor agradece ao CNPq pelo apoio recebido durante a realização deste trabalho.

REFERÊNCIAS

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. [Online]. Available: http://www.jstor.org/stable/2003354

[2] S. Winograd, "On computing the discrete Fourier transform," *Mathematics of Computation*, vol. 32, pp. 175–199, Jan. 1978.

[3] ——, *Arithmetic Complexity of Computations*. SIAM Publications, 1980.

[4] M. T. Heideman, *Multiplicative Complexity, Convolution, and the DFT*, 2nd ed. Springer-Verlag, 1988.

[5] G. J. da Silva Jr. and R. M. Campello de Souza, "Cyclotomic basis for computing the discrete Fourier transform," *International Telecommunications Symposium, ITS*, vol. 7, pp. 1–5, September 2010.

[6] G. Strang, *Linear Algebra and Its Aplications*, 4th ed. Thomson Brooks/Cole, 2006.

[7] R. E. Blahut, *Fast Algorithms for Signal Processing*, 2nd ed. Cambridge University Press, 2010.