

FFT otimizada com decomposição *Split-Radix-2/4*: Aplicação em GPS e SAR

Naiallen Carolyne Rodrigues Lima Carvalho¹, Cynthia Junqueira^{1,2} e Adilson Chinatto¹ e
¹ESPECTRO Ltda.,²UNICAMP-Universidade Estadual de Campinas

Resumo—A Transformada Rápida de Forrier (FFT) é uma importante ferramenta para processamento de sinais, por isso diversos algoritmos foram desenvolvidos com o objetivo de melhorar seu desempenho. Neste trabalho é apresentada a comparação entre o algoritmo clássico *Radix-2* e o *Split-Radix-2/4* aplicados à dados complexos e as versões desses algoritmos aplicadas à dados reais, cuja complexidade é reduzida pela remoção de redundâncias. Para testar esses algoritmos foram utilizados dados reais de sinal GPS, em que a FFT é utilizada na correlação do sinal, e imagens complexas SAR brutas, neste caso a FFT é utilizada para fazer compressão de dados e obter imagens processadas. Verificou-se melhorias em tempo de processamento entre as estruturas abordadas da ordem de 40% ou maior, quando aplicadas a processos relacionados a áreas tecnológicas importantes aos sistemas de defesa tais como os processamentos GPS e SAR.

Palavras-Chave—FFT, SAR, GPS

I. INTRODUÇÃO

A Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) é um importante processo matemático presente em um amplo espectro de aplicações. Como exemplo, pode-se citar o seu uso em telecomunicações, em que é parte integrante de sistemas OFDM (*Orthogonal Frequency Division Multiplex*) [1] e pode ser empregada em algoritmos de estimação de fase e frequência em sistemas ópticos coerentes [2]; em sistemas radar, notadamente em SAR (*Synthetic Aperture Radar*) é empregada para realizar a compressão de dados [3], aplicação similar àquelas encontradas em sistemas de estimação geofísica [4]; em sistemas de geoposicionamento como no GNSS (*Global Navigation Satellite System*) faz parte dos algoritmos destinados ao cálculo da correlação circular cruzada de códigos de espalhamento espectral [5]; finalmente, pode ser encontrada em analisadores de sinais, onde é empregada para estimação e cálculo do espectro de um determinado sinal de entrada, principalmente em situações em que as frequências são baixas [6].

O mais importante passo na otimização do cálculo da DFT foi dado por Cooley e Turkey [7] que desenvolveram o clássico algoritmo batizado de Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*), cuja grande vantagem é apresentar complexidade $O(N \log N)$ face à complexidade $O(N^2)$ da DFT, com a restrição de que o número de amostras N do conjunto de dados analisado tem que ser potência de dois, ou seja, $N = 2^n$.

Naiallen Carolyne Rodrigues Lima Carvalho, naiallen@yahoo.com.br, Cynthia Junqueira, cyjunqueira@uol.com.br, Adilson Chinatto, chinatto@espectro-eng.com.br. Os autores agradecem ao CNPq pelo financiamento do projeto GPS de Alta Dinâmica, No. 472419/2014-5.

Esse algoritmo é baseado em decomposições recursivas e a menor porção da decomposição é chamada de *radix-2* [8]. Porém, é possível a aplicação de varios tipos de decomposições, levando a *radix-m*, em que $N = m^n$, e mesmo a decomposições mistas chamadas de *split-radix*. Tem-se demonstrado que o uso desses outros tipos de decomposição comumente leva a ganhos em termos de performance computacional [9], embora possam limitar a flexibilidade do tamanho do conjunto de dados.

Neste trabalho são apresentadas otimizações ao cálculo da DFT, em que os conjuntos de dados a ser processados são provenientes de duas importantes aplicações em sistemas de defesa: Aquisição GPS (*Global Position System*) e processamento de imagem SAR (*Synthetic Aperture Radar*) bruta. Para essas duas aplicações a complexidade computacional da FFT é reduzida ao uso de técnicas do tipo *split-radix*, levando a ganho de desempenho e consequentemente redução de tempo de cálculo, obtendo algoritmos mais eficientes, tendo em vista sua aplicação em sistemas embarcados.

O artigo está estruturado da seguinte forma: na seção II são apresentados os algoritmos FFT abordados neste trabalho, na seção III são abordados os tipos de dados, na IV são mostrados os resultados e por fim na V são feitas as considerações finais.

II. ALGORITMOS FFT

Como citado anteriormente, a FFT é um algoritmo usado para calcular a DFT de forma eficiente. Esse algoritmo decompõe o conjunto de dados de entrada em subconjuntos de dados menores [8], sendo um típico algoritmo do tipo “dividir para conquistar”.

Existem diversos algoritmos para computar a DFT e embora todos os que utilizem *radix* tenham a mesma complexidade, seus desempenhos dependem do comprimento da sequência de entrada e de como é realizada a decomposição desse conjunto de dados.

A estrutura mais simples e conhecida é o *radix-2*, mas existem outras de estruturas como o *radix-3*, *radix-4*, *radix-8*, bem como combinações de duas ou mais estruturas como o *split-radix-2/4*. A decimação dos dados pode ser feita em tempo ou em frequência. Neste trabalho abordamos o *radix-2* com decimação em tempo, e o *split-radix-2/4* com decimação em frequência.

A. *Radix-2*

O *radix-2*, proposto por Cooley-Tuckey [7], decompõe um conjunto de dados de tamanho N em dois subconjuntos de

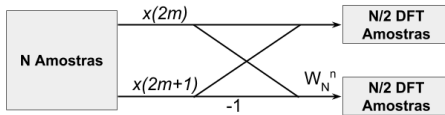


Fig. 1 - Butterfly - Radix-2

tamanho $N/2$. Neste trabalho utilizamos o algoritmo com decimação em tempo, que separa os índices n entre pares e ímpares.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (1)$$

$$X(k) = \sum_{m=0}^{N/2-1} x(2m)W_{N/2}^{kn} + W_N^{kn} \sum_{m=0}^{N/2-1} x(2m+1)W_{N/2}^{kn} \quad (2)$$

Em (1) é definida a DFT e em (2), a decimação em tempo. Em que $W_N = e^{-i2\pi/N}$ e $k = 0, 1, 2, \dots, N$.

A estrutura do *radix-2* é chamada de “butterfly” e é mostrada na figura Fig. (1).

B. split-radix-2/4

O algoritmo *split-radix-2/4*, desenvolvido por Duhamel[10], utiliza a decimação em frequência, neste caso o conjunto de dados de tamanho N é dividido diretamente em dois subconjuntos de tamanho $N/2$. O primeiro subconjunto computa os índices n de 0 a $N/2 - 1$ e o segundo subconjunto, de $N/2$ até $N - 1$. Fazendo a decimação em frequência dos k índices em pares e ímpares, tem-se:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (3)$$

$$X(2k) = \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{kn} \\ X(2k+1) = \sum_{n=0}^{N/2-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{kn} \quad (4)$$

em que (3) é a definição da DFT e (4) representa a decimação em frequência.

Observando (4), vemos que a parte par pode ser computada independentemente da parte ímpar. Aproveitando-se disso, o algoritmo *split-radix-2/4* sugere utilizar diferentes métodos para computar cada subconjunto.

Nessa metodologia, o subconjunto com índices pares ($X(2k)$) é computado pelo *radix-2* e o subconjunto com índices ímpares ($X(2k+1)$) é subdividido mais uma vez, ficando com tamanho $N/4$, e é computado pelo *radix-4*. Para os índices ímpares, teremos:

$$X(4k+1) = \sum_{n=0}^{N/4-1} \left[\left(x(n) - x\left(n + \frac{N}{2}\right) \right) \right] W_N^n W_{N/4}^{kn} \\ + \sum_{n=0}^{N/4-1} \left[i \left(x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right) \right] W_N^n W_{N/4}^{kn} \quad (5)$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \left[\left(x(n) - x\left(n + \frac{N}{2}\right) \right) \right] W_N^{3n} W_{N/4}^{kn} \\ - \sum_{n=0}^{N/4-1} i \left[\left(x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right) \right] W_N^{3n} W_{N/4}^{kn} \quad (6)$$

A estrutura do *split-radix-2/4* é mostrada na figura Fig.(2)

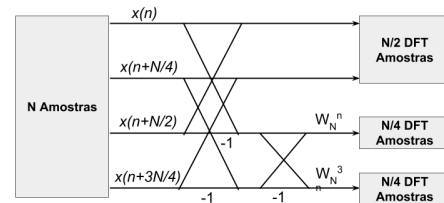


Fig. 2 - Butterfly - Split-Radix-2/4

C. Dado Real

Em muitas aplicações, a sequência de dados que será processada por FFT tem valores reais, como os dados de sinal GPS “g” utilizados no processo de aquisição e detecção do satélite que os está enviando.

Embora os algoritmos FFT sejam concebidos para computar números complexos, este algoritmo pode ser facilmente aplicado à números reais, levando à redução de complexidade quando as redundâncias do algoritmo são removidas.

Uma aproximação simples para utilizar o algoritmo FFT com dados do tipo g sem necessidade de ajuste, seria criar uma sequência complexa em que a componente real corresponderia aos elementos de g e a componente imaginária seria 0. Dessa forma, poderíamos aplicar o *radix-2* diretamente, no entanto esse método é ineficiente, já que se processaria uma sequência com o dobro do tamanho necessário.

A segunda abordagem seria utilizar g como uma sequência complexa com tamanho $N/2$ em que os índices pares seriam a componente real de g e os índices ímpares, a componente imaginária, em outra palavras, teríamos uma sequência $g_c(m) = g(2m) + ig(2m+1)$, com $m = 0, 1, \dots, N/2 - 1$. Neste caso podemos aplicar o *radix-2*, mas o resultado (G_c) é relativo a g_c . Para encontrar G , relativo a sequência g , é necessário realizar computações adicionais:

$$G_r(k) = G_{cr}(k)A_r(k) - G_{ci}(k)A_i(k) \\ + G_{cr}(N/2 - k)B_r(k) + G_{ci}(N/2 - k)B_i(k) \\ G_i(k) = G_{ci}(k)A_r(k) + G_{cr}(k)A_i(k) \\ + G_{cr}(N/2 - k)B_i(k) - G_{ci}(N/2 - k)B_r(k) \quad (7)$$

em que G_r é a parte real de G , G_i é a parte imaginária de G , G_{cr} é a parte real de G_c , G_{ci} é a parte imaginária de G_c , $A(k)$ e $B(k)$ são coeficientes em função de seno e cosseno e $k = 0, 1, \dots, N/2 - 1$ [7].

Para computar a DFT de g sem manipular sua sequência, usamos o *split-radix-2/4*, em que as redundâncias são removidas levando em consideração a propriedade de simetria. Para dados reais g , $G(k)$ e $G(N - k)$ são complexos conjugados, ou seja: $G(4k + 3) = G(N - (4K + 1)) = G(4K + 1)^*$ [10].

Portanto, para a parte par, o algoritmo *split-radix-2/4* é computado da mesma maneira que para o dado complexo. Já para parte ímpar, só é necessário computar $G(4K + 1)$.

D. Comparação entre o Radix-2 e o Split-Radix-2/4

Os algoritmos FFT têm a vantagem de serem executados “in-place”, ou seja, não necessitam de vetores auxiliares, e sua execução consiste no uso repetitivo da “butterfly”.

Na Tabela 1 [10] é mostrada uma comparação quantitativa entre o *radix-2* e o *split-radix-2/4*.

TABELA I
COMPARAÇÃO ENTRE RADIX-2 E SPLIT-RADIX-2/4

N	Multiplicações Reais		Adições Reais	
	Radix-2	Split-Radix-2/4	Radix-2	Split-Radix-2/4
16	24	20	152	148
32	88	68	408	388
64	264	196	1032	964
128	712	516	2054	2308
256	1800	1284	5896	5380
512	4360	3076	13566	12292
1024	10248	7172	30728	27652

A vantagem do *radix-2* é a simplicidade, disponibilidade de informações e flexibilidade. A complexidade desse algoritmo é $5O(N \log N)$ [11].

A vantagem do *split-radix-2/4* sobre o *radix-2*, além da menor quantidade de operações, é que este algoritmo tem a mesma estabilidade do *radix-4* e a mesma flexibilidade do *radix-2*. A complexidade desse algoritmo é $4O(N \log N)$ [11].

III. TIPOS DE DADOS

Como mencionado, para exemplificar a utilização da FFT com dados complexos e reais, consideramos o processamento de aquisição de sinais GPS (*Global Position System*) e o processamento de dados brutos SAR (*Synthetic Aperture Radar*).

A. Aquisição do Sinal GPS

Os satélites do GNSS, como por exemplo, o americano GPS, enviam continuamente sinais para Terra, os quais possuem basicamente três componentes [4]: 1) Dado de Navegação, que é uma mensagem binária contendo informações das efemérides, por exemplo; 2) PRN (*Pseudo-Random Noise*) que é um código binário único para cada satélite, responsável pelo espalhamento espectral do sinal, através dele é possível identificar a qual satélite o dado adquirido pertence; 3) Portadora, que é uma onda senoidal cuja frequência está na banda L.

Os códigos PRN, assim como os Dados de Navegação, correspondem a sequências de valores -1 e +1, chamados de “chips” e não *bits*, para enfatizar que eles não carregam informação.

A aquisição do sinal GPS tem como objetivo de determinar quais satélites estão visíveis, estimar a frequência Doppler normalizada F_d e o atraso do código PRN. O sinal GPS contém duas componentes em frequência [9] 1) L1 = 1575.42MHz, que contém os códigos C/A (civil) e P(Y) (militar, não disponível); 2) L2 = 1227.6 MHz que contém somente o código P(Y). A equação geral do sinal GPS na banda L1 pode ser descrita como [5]:

$$S_l^p(t) = A_p P^P(t) D^P(t) \cos(2\pi f_1 t) + A_c G^P(t) D^P(t) \sin(2\pi f_1 t) \quad (8)$$

em que A_p e A_c são respectivamente as amplitudes do código P(Y) e C/A, $P^P(t)$ é o PRN do código P(Y), $G^P(t)$ é o PRN do código C/A (também conhecido como *Gold Code*) e $D^P(t)$ é o dado de navegação.

O processo de aquisição consiste basicamente em realizar a correlação circular entre o dado que é recebido do satélite e um conjunto de códigos PRN conhecidos, determinando qual satélite enviou a mensagem, o desvio de frequência observado devido à movimentação relativa entre o transmissor e o receptor e o deslocamento em *chips* do código recebido. Essas informações são usadas para se obter a distância relativa entre um receptor e um conjunto de satélites visíveis e, por processo de trilateração, estimar a posição do receptor.

Neste trabalho utilizou-se sinais simulados para representar os dados dos satélites. A Fig. 3 apresenta o diagrama em blocos do processo de aquisição. Como pode-se observar, várias FFTs são empregadas para calcular a correlação circular entre os dados recebidos e os códigos PRN gerados.

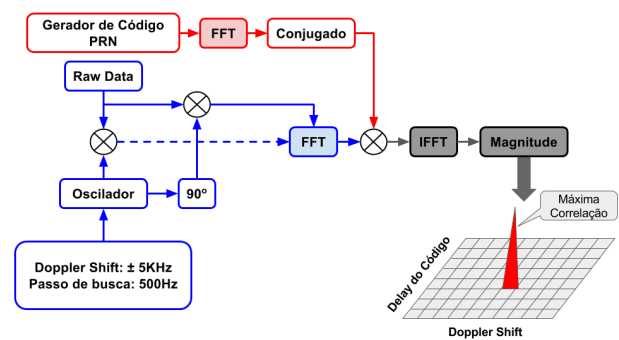


Fig. 3 - Processo de Aquisição do Sinal GPS

B. Processamento de dados brutos SAR

O SAR é um radar imageador que atua na região de micro-ondas do espectro eletromagnético e sintetiza uma antena de grande dimensão [12].

O sinal transmitido por um SAR é um *chirp* modulado linearmente em frequência [3], cuja envoltória complexa é representada por (9).

$$s(t) = A \text{rect} \left[\frac{t}{T_p} \right] e^{i\pi\gamma_R t^2} \quad (9)$$

em que A é a amplitude do sinal, γ_R é o *chirp rate* ou a variação instatânea de frequência do sinal emitido, T_p é a duração do pulso transmitido e $\text{rect} \left[\frac{t}{T_p} \right]$ é a função retangular de amplitude unitária.

O sinal recebido pelo radar é chamado de *raw data*, ou seja, é um dado sem foco e sem compressão. Portanto, o objetivo do processamento desse sinal é gerar uma imagem focada que possa ser usada em processamentos posteriores de classificação, por exemplo.

Um dos algoritmos mais conhecidos e simples para essa finalidade é o Processador *Narrow Focus*, cujo processamento consiste em aplicar a FFT com o objetivo de comprimir os dados em *range* e azimuth. A Fig. 4 mostra o diagrama simplificado deste processador.

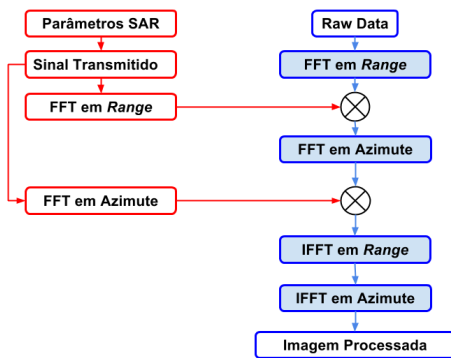


Fig. 4 - Diagrama do do processador *Narrow Focus*

Os dados utilizados como *raw data* neste trabalho é uma cena obtida pelo satélite ERS (*European Remote Sensing*), que pertencem a ESA (*European Space Agency*), e foram disponibilizadas pelo Centro Aeroespacial Alemão (DLR). O *raw data* e os parâmetros do sensor estão disponíveis em [13].

IV. RESULTADOS

Embora o objetivo principal do trabalho seja comparar a performance dos algoritmos FFT *Radix-2* com decimação em tempo e o *Split-Radix-2/4*, nas subseções a seguir também são mostrados os resultados dos processadores relativos a cada tipo de dado exemplificado neste trabalho.

Como os testes realizados visaram a comparação de desempenho entre os algoritmos FFT é importante fornecer informações sobre a máquina utilizada, a qual possui processador Intel(R) Core(TM) i7, com velocidade de 2.40GHz e sistema operacional Windows 10 (64 bits). E a linguagem de programação utilizada foi o C#.

A. Aquisição do Sinal GPS

Como citado na seção III, o objetivo da aquisição do sinal GPS é determinar quais satélites estão visíveis, para isso é realizada a correlação entre um sinal conhecido e o sinal recebido. O resultado da correlação é um pico que indica a

magnitude, o desvio da frequência Doppler e o atraso dos chips.

Na Fig. 5 é mostrado o resultado da aquisição de um sinal GPS em que houve autocorrelação e seu deslocamento de chips e frequência.

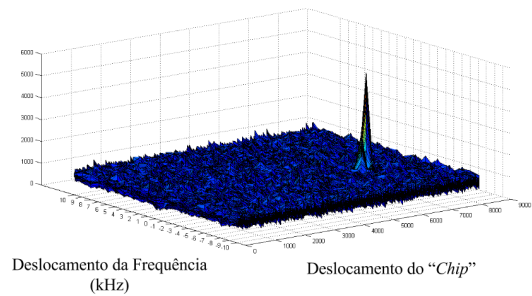


Fig. 5 - Autocorrelação do sinal GPS

Para esses dados, fez-se a comparação entre o algoritmo *radix-2* complexo (criou-se uma sequência complexa em que as componentes reais são os dados do sinal GPS e as imaginárias são preenchidas com zeros), *radix-2* real e o *split-radix-2/4* real.

Na Fig. 6 é mostrada a comparação de tempo, em milissegundos, entre os algoritmos citados. A média de ganho de tempo do *radix-2* real sobre o *radix-2* complexo foi de aproximadamente 55% e do *split-radix-2/4* real sobre o *radix-2* complexo foi de aproximadamente 59%.

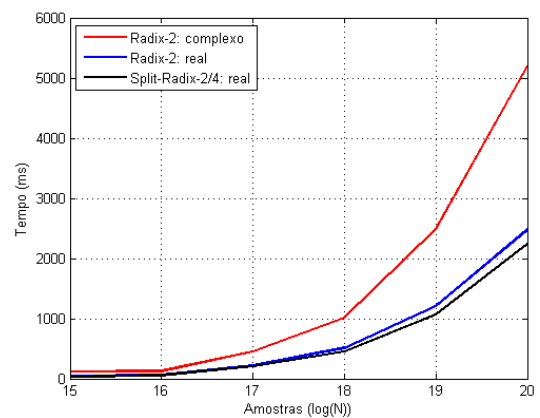


Fig. 6 - Comparação entre os algoritmos FFT

Para testar do algoritmo de aquisição completo é necessário realizar a correlação dos sinais de 32 satélites. Neste trabalho, utilizou-se 50 bits de dados e 8MHz de frequência de amostragem. Com essa configuração, de cada satélite é adquirido um volume de dados de aproximadamente 1.074.150 amostras. Usando o *radix-2* complexo para essa quantidade de amostras multiplicada pela quantidade de satélite, o tempo de execução ficou em torno de 15 minutos, e com o *split-radix-2/4* o tempo caiu para aproximadamente de 2 minutos.

B. Processamento de dados brutos SAR

Neste trabalho utilizou-se o algoritmo *Narrow Focus* para processar o *raw data*, que possui 2048x2048 amostras. Na Fig. 7 é mostrado o *Raw Data* e na Fig. 8 é mostrada a imagem processada.

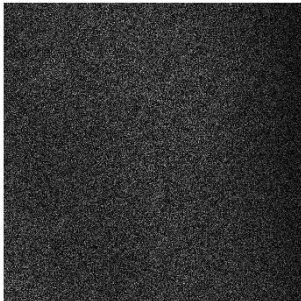


Fig. 7 - Raw Data - Imagem sem foco

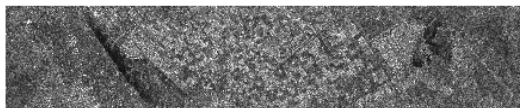


Fig. 8 - Imagem processada

É importante lembrar que como estamos trabalhando com imagens, a FFT é bidimensional. No entanto para o teste de desempenho de dados complexos utilizou-se os algoritmos FFT de maneira unidimensional.

Na Fig. 9 é mostrada as comparações de tempo entre os algoritmos *radix-2* e *split-radix-2/4*, ambos complexos. Assim como para dados reais, o uso do *split-radix-2/4* no processamento de imagens SAR leva a ganho significativo em tempo de processamento, especialmente em situações em que o número de amostras é grande. A média de ganho de tempo do *radix-2* real sobre o *split-radix-2/4* complexo foi de aproximadamente 43%.

Dessa forma, um algoritmo do tipo *Narrow Focus* utilizando o *split-radix-2/4* pode permitir o processamento de maior quantidade de imagens em menor tempo, ou ainda permitir que os radares SAR tenham o núcleo de processamento simplificado com redução de consumo energético, dimensão e massa.

V. CONCLUSÕES

Neste trabalho abordou-se o cálculo de DFT através de estruturas diferentes do tradicional *radix-2* complexo. Especificamente, empregou-se a decomposição mista chamada *split-radix-2/4*. Essa estrutura otimizada foi empregada no cálculo da DFT de dois importantes processos relacionados a várias áreas tecnológicas: processamento GPS e processamento SAR.

No caso do processamento GPS, a utilização do *split-radix-2/4* foi somada a simplificações decorrente da propriedade de simetria do sinal no domínio da frequência quando o sinal no domínio do tempo é real. Nas duas aplicações, foi constatado ganho de tempo superior a 40%. No caso da aquisição GPS o ganho em tempo de processamento foi maior que 55% devido

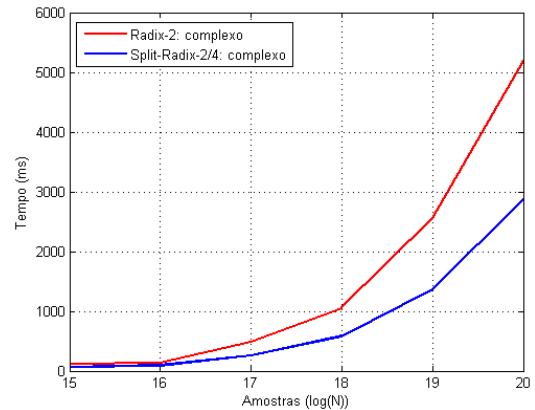


Fig. 9 - Comparação entre os algoritmos FFT

às simplificações decorrentes da utilização de apenas dados reais.

A utilização da decomposição *split-radix-2/4* permite simplificações de hardware e de software em diversas aplicações críticas, levando a economia energética e à diminuição de massa dos equipamentos. Esse tipo de economia pode ser determinante na aplicação de processos complexos em sistemas embarcados, por exemplo. Como continuidade natural deste trabalho destaca-se a implementação e análise de desempenho da FFT com decomposição *split-radix-2/4* em processadores DSP e em dispositivos FPGA, além da análise da influência da quantização dos coeficientes em processos que operam em ponto-fixado.

REFERÊNCIAS

- [1] CHANG, R. W. Synthesis of band-limited orthogonal signals for multi-channel data transmission. *Bell System Technical Journal*, Wiley Online Library, v. 45, n. 10, p. 1775–1796, 1966.
- [2] CAO, Y. et al. Frequency estimation for optical coherent mpsk system without removing modulated data phase. *IEEE Photonics Technology Letters*, IEEE, v. 22, n. 10, p. 691–693, 2010.
- [3] MURA, J. C. et al. Processador SAR compacto baseado em FPGA para monitoramento em tempo real. *Simposio Brasileiro de Sensoriamento Remoto*, v. 15, p. 7572–7579, 2011.
- [4] LECLERE CYRIL BOTTERON, P.-A. F. J. Acquisition of modern GNSS signals using a modified parallel code-phase search architecture. *Signal Processing*, v. 95, p. 177–191, 2014.
- [5] KAPLAN, E.; HEGARTY, C. *Understanding GPS: principles and applications*. [S.l.]: Artech house, 2005.
- [6] ADNANI, A. A.; DUPLICY, J.; PHILIPS, L. Spectrum analyzers today and tomorrow: part 1 towards filterbanks-enabled real-time spectrum analysis. *IEEE Instrumentation & Measurement Magazine*, IEEE, v. 16, n. 5, p. 6–11, 2013.
- [7] TEUKOLSKY, S.; PRESS, W. H. *Numerical recipes 3rd edition: The art of scientific computing*. [S.l.]: Cambridge university press, 2007.
- [8] COOLEY, J. W.; LEWIS, P.; WELCH, P. *The Fast Fourier Transform algorithm and its applications*. [S.l.]: IBM Watson Research Center, 1967.
- [9] LIN, W. et al. Acquisition of GPS Software Receiver Using Split-Radix FFT. *IEEE International Conference on Systems, Man, and Cybernetics*, p. 4608–4613, 2006.
- [10] DUHAMEL, P. Implementation of "split-radix"fft algorithms for complex, real, and real-symmetric data. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 34, n. 2, p. 285–295, 1986.
- [11] LIN, S.-J.; CHUNG, W.-H. The split-radix fast fourier transforms with radix-4 butterfly units. In: *IEEE. Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. [S.l.], 2013. p. 1–5.
- [12] LEE, J.-S.; POTTIER, E. *Polarimetric Radar Imaging: From Basics to Application*. [S.l.]: CRC Press, 2009.
- [13] SAREDU. <https://saredu.dlr.de>. Acessado: 28-6-2016.