

Implementação em FPGA de Algoritmos NCMA Aplicados a Beamforming

Adilson Chinatto^{1,2}, Cynthia Junqueira^{1,3} e João M. T. Romano¹

¹Universidade Estadual de Campinas, ²Spectrum Line Ltda., ³Instituto de Aeronáutica e Espaço

Resumo—Este artigo apresenta uma implementação prática em FPGA (Field Programmable Gate Array) do algoritmo de módulo constante normalizado (NCMA). O algoritmo é tratado na versão estritamente linear – ou clássica – e na versão largamente linear, sendo deduzida ainda a versão largamente linear otimizada para sinais reais. É apresentada comparação de custo computacional e resultados de cossimulação em *hardware* em um cenário relacionado a mitigação de interferentes no contexto de *beamforming*, mostrando ganho de desempenho das versões largamente lineares sobre a versão estritamente linear. Mostra-se, ainda, que a versão largamente linear otimizada para sinais reais apresenta considerável redução da complexidade computacional e da latência.

Palavras-Chave—NCMA, FPGA, Processamento Largamente Linear, Custo Computacional.

Abstract—This article presents a FPGA implementation of the well known normalized constant modulus algorithm (NCMA). The algorithm is treated in its strictly linear – or classic – and widely linear versions. An optimized version for real signals is yet derived. Computational costs comparisons and hardware cossimulation results in a scenario related to interference mitigation through beamforming are presented. The results show that widely linear versions perform better than the strictly linear one. Finally, it is shown that widely linear optimized for real signals versions lead to a significant reduction in computational complexity and latency.

Keywords—NCMA, FPGA, Widely Linear Processing, Computational Cost.

I. INTRODUÇÃO

O *beamforming*, ou conformação de feixe, é uma técnica aplicada a arranjos de antenas que possibilita o posicionamento do lóbulo principal e os nulos do diagrama de radiação do arranjo permitindo, por exemplo, mitigação de interferentes intencionais ou não e maximização da relação sinal ruído [1]. Aplicações de *beamforming* são comumente encontradas na literatura relacionadas a sistemas militares [2] e, nos últimos 15 anos, estão cada vez mais ligadas ao setor de telecomunicações [3], imageamento [4], astrofísica [5], geofísica [6] e biomédica [7]. Este avanço se dá devido à tecnologia de processamento de sinais e aos dispositivos de *hardware* reconfigurável desenvolvidos nos últimos anos.

A tecnologia de arranjos de antenas via processamento de sinal requer uso elevado de recursos de processamento e lida com velocidades computacionais extremamente altas, dificultando sua implementação prática através de *software* com execução sequencial. Por outro lado, o processamento

paralelo em altas taxas proporcionado por dispositivos FPGA de alta capacidade recentemente lançados no mercado permite o desenvolvimento de plataformas de alto desempenho e flexibilidade para implementação de diferentes arquiteturas de *beamforming* e algoritmos adaptativos.

Quanto ao uso de algoritmos adaptativos em *beamforming* podem-se empregar aqueles que necessitem ou não de sequência de treinamento. O uso dos algoritmos cegos aumenta o grau de liberdade da rede de antenas, pois o algoritmo independe de conhecimento prévio da direção de chegada de sinais e de sequências de treinamento. Dentro desta classe de algoritmos um dos mais promissores é o algoritmo do módulo constante (CMA) e suas variações, dentre elas o bem estabelecido algoritmo do módulo constante normalizado (NCMA) [8].

Dadas essas premissas, usando uma plataforma em FPGA implementa-se e compara-se o desempenho do algoritmo adaptativo NCMA no contexto do processamento estritamente linear (NCMA-SL), largamente linear (NCMA-WL) e largamente linear otimizado para sinais reais (NCMA-WL-R) [9]. Ressaltam-se as comparações de desempenho, custo computacional teórico fornecido pelas equações dos algoritmos e custo computacional obtido na implementação.

Este artigo está estruturado como se segue: a seção II apresenta a base matemática do processamento largamente linear e um resumo do algoritmo NCMA em suas versões estritamente linear, largamente linear e largamente linear otimizada para sinais reais para, em seguida, discutir-se sobre a complexidade computacional dos referidos algoritmos. A seção III inclui a descrição da implementação em FPGA, sua metodologia e complexidade computacional prática. Na seção IV são apresentados resultados no contexto de *beamforming* aplicado à mitigação de interferentes utilizando a implementação prática dos algoritmos em FPGA. Conclusões e perspectivas futuras são estabelecidas na seção V.

II. PROCESSAMENTO LARGAMENTE LINEAR E ALGORITMO NCMA

Seja $\mathbf{x}(k)$ um vetor amostra complexo recebido no instante k e seja $d(k)$ o valor do sinal de referência no mesmo instante. Definindo-se o resultado da filtragem de $\mathbf{x}(k)$ por um filtro complexo \mathbf{w} como

$$y(k) = \mathbf{w}^H \mathbf{x}(k), \quad (1)$$

onde $(\cdot)^H$ corresponde à transposição hermitiana, pode-se determinar o filtro ótimo $\mathbf{w} = \mathbf{w}_o$ que minimiza a função custo

Adilson Chinatto, Spectrum Line Ltda., Campinas, São Paulo, Brasil, Cynthia Junqueira, Instituto de Aeronáutica e Espaço, São José dos Campos, São Paulo, Brasil, João M. T. Romano, Departamento de Microondas e Óptica, Universidade Estadual de Campinas, Campinas, São Paulo Brasil. E-mails: chinatto@espectro-eng.com.br, cynthiaccmj@iae.cta.br, romano@dmo.fee.unicamp.br.

correspondente ao erro quadrático médio, definida como

$$J(\mathbf{w}_o) = E\left\{\left|d(k) - \mathbf{w}_o^H \mathbf{x}(k)\right|^2\right\} \quad (2)$$

onde $E\{\cdot\}$ corresponde ao operador esperança. O resultado desse problema é a bem conhecida solução de *Wiener*, dada por

$$\mathbf{w}_o = \mathbf{R}_{xx}^{-1} \mathbf{p} \quad (3)$$

onde $\mathbf{R}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^H(k)\}$ ou matriz de autocorrelação de $\mathbf{x}(k)$, $\mathbf{p} = E\{\mathbf{x}(k)d^*(k)\}$ ou correlação cruzada entre $\mathbf{x}(k)$ e $d(k)$, e $(\cdot)^*$ corresponde à operação de conjugação complexa.

Em [10] e [11] foi proposto e desenvolvido um método de filtragem baseada na minimização do erro quadrático médio e denominada processamento largamente linear, cuja essência é a utilização de $\mathbf{x}(k)$ e de $\mathbf{x}^*(k)$ conjuntamente para a determinação de dois filtros ótimos \mathbf{w}_1 e \mathbf{w}_2 tais que

$$y_w(k) = \mathbf{w}_1^H \mathbf{x}(k) + \mathbf{w}_2^H \mathbf{x}^*(k) \quad (4)$$

Com isso se exploram as características de circularidade [12] do sinal recebido, implicando, em certas condições, em ganho de desempenho. Usando-se as notações

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}^*(k) \end{bmatrix} \quad \text{e} \quad \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \quad (5)$$

e definindo a função custo a ser minimizada como

$$J_w(\tilde{\mathbf{w}}_o) = E\left\{\left|d(k) - \tilde{\mathbf{w}}_o^H \tilde{\mathbf{x}}(k)\right|^2\right\} \quad (6)$$

chega-se ao filtro ótimo largamente linear

$$\tilde{\mathbf{w}}_o = \mathbf{R}_{\tilde{x}\tilde{x}}^{-1} \tilde{\mathbf{p}} \quad (7)$$

em que $\tilde{\mathbf{p}} = E\{\tilde{\mathbf{x}}(k)d^*(k)\}$ e

$$\mathbf{R}_{\tilde{x}\tilde{x}} = \begin{bmatrix} \mathbf{R}_{xx} & \mathbf{C}_{xx} \\ \mathbf{C}_{xx}^* & \mathbf{R}_{xx}^* \end{bmatrix} \quad (8)$$

sendo $\mathbf{C}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^T(k)\}$ a matriz de pseudo-correlação de $\mathbf{x}(k)$.

A. Algoritmo de Módulo Constante Normalizado

O algoritmo de módulo constante (CMA) é um dos muitos algoritmos adaptativos autodidatas ou cegos que exploram a característica constante do módulo de um sinal desejado, tendo sido desenvolvido de forma independente em [13] e [14], e cuja equação de adaptação é dada por

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \mathbf{x}(k) \left(|y(k)|^2 - 1 \right) y^*(k) \quad (9)$$

onde μ é uma constante arbitrária real chamada de passo de adaptação. Embora simples, o critério adotado no CMA faz

com que sua convergência seja penalizada em situações nas quais a potência do sinal de entrada sofra flutuações. Para contornar esse problema e impedir que os coeficientes do filtro de adaptativo divirjam, pode ser realizada uma ponderação do passo de adaptação em relação à potência do sinal de entrada, levando ao algoritmo de módulo constante normalizado (nomeado aqui como NCMA-SL por questões de clareza), proposto por vários autores, entre eles por Hilal e Duhamel [8], cuja equação de adaptação é

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{\|\mathbf{x}(k)\|^2} \mathbf{x}(k) \left(1 - \frac{1}{|y(k)|} \right) y^*(k) \quad (10)$$

Substituindo-se (4) e (5) em (10), é possível obter a versão largamente linear do algoritmo de módulo constante normalizado (NCMA-WL), cuja equação de adaptação é

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \frac{\mu}{\|\tilde{\mathbf{x}}(k)\|^2} \tilde{\mathbf{x}}(k) \left(1 - \frac{1}{|y_{WL}(k)|} \right) y_w^*(k) \quad (11)$$

com $y_w(k)$ dado por (4).

B. NCMA Otimizado para Sinais Reais

Embora o uso do processamento largamente linear possa implicar em ganho de desempenho sob certas condições de circularidade do sinal recebido, seu emprego resulta em aumento da complexidade computacional do algoritmo adaptativo, embora para determinadas situações específicas possa-se reduzir sua complexidade computacional aos níveis da apresentada pelo processamento clássico de filtragem [15]. Esse aumento da complexidade computacional é devido ao uso simultâneo de $\mathbf{x}(k)$ e de $\mathbf{x}^*(k)$ para a determinação dos filtros \mathbf{w}_1 e \mathbf{w}_2 .

Por outro lado, quando $d(k)$ é real tem-se $\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p}^T & \mathbf{p}^T \end{bmatrix}^T$, levando, através da avaliação de (7), a

$$\tilde{\mathbf{w}}_R = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_1^* \end{bmatrix} \quad (12)$$

Substituindo-se (9) em (4) chega-se a

$$y_R(k) = \Re\{\mathbf{w}_R^H \mathbf{x}(k)\} \quad (13)$$

onde $\Re\{\cdot\}$ é o operador real e \mathbf{w}_1 foi substituído por \mathbf{w}_R por questões de clareza. É fácil verificar que $y_R(k)$ apresenta menor complexidade computacional que $y_w(k)$ e que $y(k)$ e, usando a mesma premissa, é possível determinar o algoritmo NCMA largamente linear otimizado para sinais reais (NCMA-WL-R).

De acordo com [8] e considerando (13), para se determinar o valor ótimo do passo de adaptação em (9), é necessário definir dois tipos de erros, o erro *a priori* fornecido pela filtragem no tempo k

$$e_1(k) = |y_R(k)|^2 - 1 = \left| \mathbf{w}_R^H(k) \mathbf{x}(k) + \mathbf{x}^H(k) \mathbf{w}_R(k) \right|^2 - 1 \quad (14)$$

e o erro *a posteriori* provido pela nova filtragem no tempo $k+1$

$$e_2(k) = \left| \mathbf{w}_R^H(k+1)\mathbf{x}(k) + \mathbf{x}^H(k)\mathbf{w}_R(k+1) \right|^2 - 1 \quad (15)$$

O objetivo é encontrar o passo de adaptação μ_o para o qual

$$e_2(k) = 0, \quad \forall k \quad (16)$$

sabendo-se que a função custo é dada por

$$J_R(\mathbf{w}_R) = \left(\left| \mathbf{w}_R^H(k+1)\mathbf{x}(k) + \mathbf{x}^H(k)\mathbf{w}_R(k+1) \right|^2 - 1 \right)^2 \quad (17)$$

Diferenciando (17) em relação a \mathbf{w}_R e igualando-se a zero, chega-se à expressão

$$\mathbf{w}_{k+1}^H \mathbf{x}_k \mathbf{x}_k^T \mathbf{w}_{k+1}^* + 2\mathbf{w}_{k+1}^H \mathbf{x}_k \mathbf{x}_k^H \mathbf{w}_{k+1} + \mathbf{w}_{k+1}^T \mathbf{x}_k^* \mathbf{x}_k^H \mathbf{w}_{k+1} = 0 \quad (18)$$

em que, por motivos de clareza, $\mathbf{w}_R(k+1) = \mathbf{w}_{k+1}$ e $\mathbf{x}(k) = \mathbf{x}_k$, e que pode ser escrita com a soma de fatores

$$\kappa_1 + 2\kappa_2 + \kappa_3 = 0 \quad (19)$$

Substituindo em (19) o valor de $\mathbf{w}(k+1)$ para do CMA, apresentado em (10), após extensas operações algébricas chega-se ao conjunto de equações

$$\begin{aligned} \kappa_1 &= \mathbf{w}_k^H \mathbf{x}_k \mathbf{x}_k^T \mathbf{w}_k^* + (\psi y_k)^2 - \psi (y_k^* \mathbf{w}_k^H \mathbf{x}_k + y_k \mathbf{w}_k^H \mathbf{x}_k) \\ \kappa_2 &= \mathbf{w}_k^H \mathbf{x}_k \mathbf{x}_k^H \mathbf{w}_k + (\psi |y_k|)^2 - \psi (y_k^* \mathbf{w}_k^H \mathbf{x}_k + y_k \mathbf{w}_k^H \mathbf{x}_k) \\ \kappa_3 &= \mathbf{w}_k^T \mathbf{x}_k \mathbf{x}_k^H \mathbf{w}_k + (\psi y_k)^2 - \psi (y_k \mathbf{w}_k^H \mathbf{x}_k + y_k \mathbf{w}_k^H \mathbf{x}_k) \end{aligned} \quad (20)$$

em que $\psi = \mu_o \mathbf{x}_k^H \mathbf{x}_k (|y_k| - 1)$ e $y_k = y(k+1)$. Substituindo (20) em (19) determina-se a equação de segundo grau cuja solução corresponde a μ_o , dada por

$$4\mu_o^2 \xi^2 \zeta |y_k|^2 - 4\mu_o \xi |y_k|^2 + 1 = 0 \quad (21)$$

onde $\xi = \mathbf{x}_k^H \mathbf{x}_k$ e $\zeta = |y_k| - 1$. A equação de segunda ordem (21) admite duas soluções, das quais a única solução praticável é

$$\mu_o = \frac{|y_k| - 1}{2\|\mathbf{x}_k\|^2 (|y_k|^2 - 1)|y_k|} \quad (22)$$

em que o termo ξ foi substituído pela norma euclidiana de \mathbf{x}_k . Finalmente, a substituição do valor de μ_o determinado em (22) na equação de adaptação do CMA (9) resulta na equação de adaptação para o algoritmo NCMA-WL-R, que é membro da família de algoritmos NCMA, largamente linear e otimizado para sinais desejados reais, e dada por

$$\mathbf{w}_R(k+1) = \mathbf{w}_R(k) - \frac{\mu}{2\|\mathbf{x}(k)\|^2} \mathbf{x}(k) \left(1 - \frac{1}{|y_R(k)|} \right) y_R(k) \quad (23)$$

sendo $y_R(k)$ dado por (13), sendo, portanto, real.

C. Complexidade Computacional Teórica

Conforme mencionado, o NCMA-WL deve apresentar uma maior complexidade computacional que o NCMA-SL devido ao fato de usar vetores de entrada e de ponderação com o dobro do tamanho. Por outro lado, o NCMA-WL-R, que é otimizado para sinais desejados reais, apresenta vetores de entrada e de ponderação do mesmo tamanho que o NCMA-SL, além de que a saída do filtro é real. A Tabela I apresenta o custo computacional dos algoritmos NCMA-SL, NCMA-WL e NCMA-WL-R em termos de somas reais, multiplicações reais, divisões reais e operações trigonométricas, considerando que o vetor de entrada tenha dimensão $M \times 1$.

A Tabela I mostra que o uso do NCMA-WL-R, nas situações em que ele se presta, leva a uma redução considerável de recursos computacionais, especialmente no que diz respeito ao uso de operações trigonométricas – que corresponde a operações de extração de módulo, por exemplo.

III. IMPLEMENTAÇÃO PRÁTICA EM FPGA

Para a validação do desempenho e da complexidade computacional dos algoritmos, foi realizada sua implementação em dispositivo de lógica programável e, através de recurso de cossimulação em *hardware*, os resultados foram verificados. Os algoritmos foram implementados em VHDL e compilados usando o *software* XST presente no pacote ISE[®] 12.4 da Xilinx[®] [16]. A cossimulação foi feita através da geração de vetores de teste e análise de resultados no *software* Matlab[®] R2008 da Mathworks[®] [17] e processamento dos algoritmos no cartão ML-505 da Xilinx[®] [18], que contém um dispositivo FPGA XC5VLX50T-11361C da família Virtex5[®]. A Figura 1 mostra uma fotografia do sistema usado para a validação da implementação prática.

TABELA I. COMPLEXIDADE COMPUTACIONAL DOS ALGORITMOS NCMA

Algoritmo	+	×	÷	trig.
NCMA-SL	10M - 1	10M + 5	1	1
NCMA-WL	18M - 1	14M + 5	1	1
NCMA-WL-R	6M - 1	6M + 1	1	0

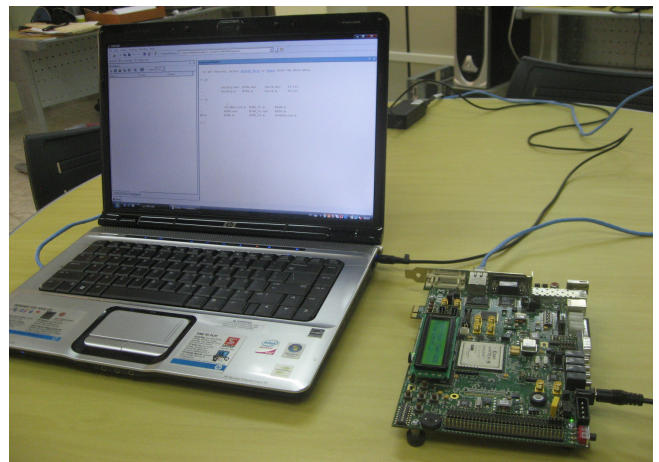


Fig. 1. Fotografia do sistema usado para validação da implementação prática em dispositivo FPGA.

A. Metodologia de Implementação

Conforme mencionado, os algoritmos NCMA empregam quatro tipos de operações aritméticas reais em sua implementação: somas, multiplicações, divisões e operações trigonométricas. As operações aritméticas do tipo soma e multiplicação podem ser realizadas em um único ciclo de relógio já que o dispositivo FPGA possui recursos dedicados para esses fins. Por outro lado, operações do tipo divisão demandam um número maior de ciclos de relógio para serem feitas, que é proporcional ao número de bits que compõem o dividendo e o divisor. O mesmo ocorre com as operações trigonométricas necessárias, do tipo extração do módulo ou obtenção de fase de número complexo. A necessidade de utilização de mais ciclos de relógio para realização das operações matemáticas implica em aumento da latência necessária à obtenção do vetor de ponderação e da saída do algoritmo. Assim, como é inevitável que a implementação dos algoritmos NCMA implique em latência, optou-se pela reutilização de determinadas estruturas lógicas em diferentes ciclos de relógio, levando a uma minimização da área de lógica usada, mesmo considerando os recursos usados para o controle.

Foram criados quatro tipos de estruturas para a implementação: somadores, dispositivos multiplicadores e acumuladores (MAC), divisor e CORDIC [19]. Os somadores foram gerados através de lógica combinacional apenas. Já os MAC empregam um multiplicador dedicado de duas entradas ligado a um somador e um acumulador e suas funções correspondem à realização de multiplicação de vetores reais. O divisor corresponde à função de inversão através de aproximações sucessivas de forma a minimizar a latência necessária à obtenção da resposta. Finalmente, o CORDIC implementado realiza a operação de determinação da fase do número complexo presente à sua entrada.

A Figura 2 mostra o diagrama de blocos da implementação em dispositivo de lógica programável para os algoritmos NCMA-SL e NCMA-WL. Já a Figura 3 apresenta o diagrama de blocos para o NCMA-WL-R.

B. Complexidade Computacional Prática

A complexidade computacional da implementação em FPGA pode ser computada através de recursos lógicos usados e através da latência necessária para a obtenção das respostas.

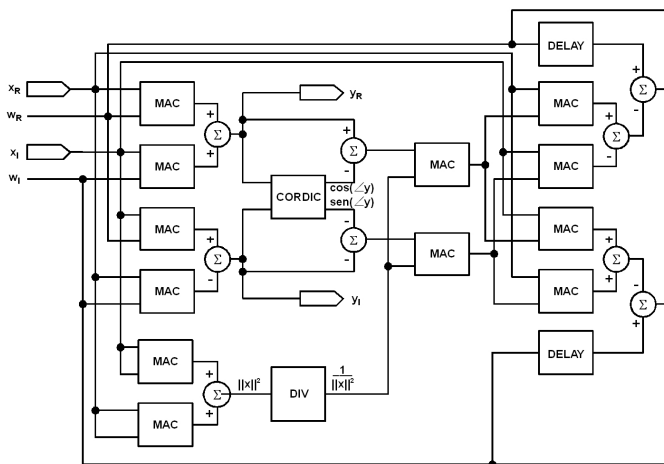


Fig. 2. Diagrama de blocos para os algoritmos NCMA-SL e NCMA-WL implementado em FPGA.

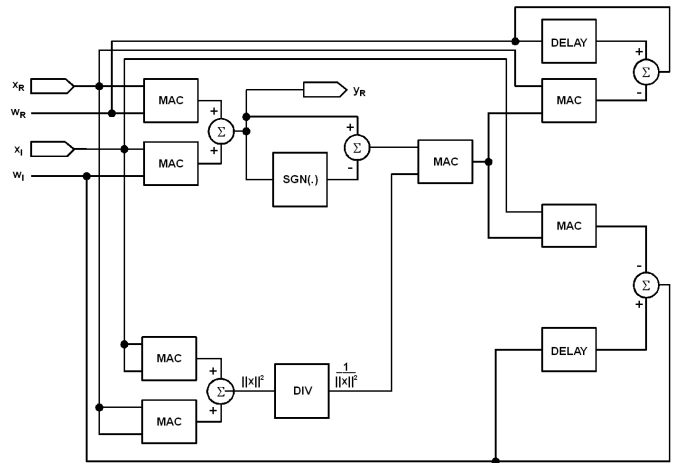


Fig. 3. Diagrama de blocos para o algoritmo NCMA-WL-R implementado em FPGA.

Cada dispositivo MAC implica em uma latência de $M + 1$ ciclos de relógio, sendo M o comprimento dos vetores a serem multiplicados. Os outros dispositivos apresentam latência fixa, sendo que o divisor disponibiliza o resultado da inversão 9 ciclos de relógio após receber a entrada e o CORDIC fornece a fase do sinal complexo presente em sua entrada em 11 ciclos de relógio. Embora o diagrama de blocos da Figura 2 apresente a implementação com 12 dispositivos MAC, graças à técnica de reuso esse número pode ser reduzido a 6. Dessa forma, essa implementação pode ser empregada tanto para o NCMA-SL quanto para o NCMA-WL, sendo a diferença de custo computacional entre um e outro dada apenas em termos de latência. Por outro lado, conforme mencionado, o NCMA-WL-R introduz uma significativa redução no número de elementos multiplicadores usados devido a considerar que o sinal desejado é real, além de não necessitar do uso de CORDIC por não possuir operações trigonométricas. Com isso, o NCMA-WL-R implementado em dispositivo de lógica programável tem seu diagrama de blocos reduzido ao mostrado na Figura 3.

A comparação dos diagramas de blocos mostrados na Figura 2 e na Figura 3 aponta para a redução de recursos proporcionada pelo NCMA-WL-R. O número de dispositivos MAC usados é reduzido a 7 e não há necessidade de utilização de CORDIC. Considerando as técnicas de reuso, o número de dispositivos MAC é reduzido a 4 para este caso.

Dadas essas considerações, a Tabela II apresenta uma comparação da complexidade computacional dos três algoritmos em termos de blocos funcionais, latência em ciclos de relógio e recursos lógicos do componente FPGA usado para sua implementação, considerando $M = 4$. Os recursos lógicos do componente FPGA foram divididos em *Lookup Tables* (LUT), *Flip-Flops* (FF) e multiplicadores dedicados (mult). Nota-se uma similaridade entre as complexidades computacionais teóricas e as obtidas praticamente em termos de latência, LUT e FF.

TABELA II. COMPLEXIDADE COMPUTACIONAL DAS IMPLEMENTAÇÕES

	NCMA-SL	NCMA-WL	NCMA-WL-R
MAC	6	6	4
Divisor	1	1	1
CORDIC	1	1	0
Latência	24	32	22
LUT	1931	2358	1259
FF	873	1278	635
mult	7	7	5

IV. RESULTADOS

Para verificar o desempenho dos algoritmos implementados em FPGA, considerou-se a tarefa de mitigação de interferentes em arranjos de antenas através de *beamforming*. Os vetores de teste foram gerados de forma a emular a situação em que N sinais incidem sobre um arranjo composto por $M = 4$ antenas isotrópicas, dispostas de forma linear e igualmente espaçadas (ULA). Os resultados foram obtidos em forma de taxa de erro de símbolo (SER) de um sinal desejado s_D corrompido por $N - 1$ sinais interferentes em um ambiente com relação sinal-ruído (SNR) e relação interferente-ruído (INR) de 10dB. O sinal s_D apresentou modulação BPSK e os sinais interferentes modulação 8-PAM. A direção de chegada de s_D foi variada de -90° a $+90^\circ$, em passos de 1° , sendo a direção de chegada 0° correspondente à normal à linha do arranjo.

Foram consideradas duas situações, com $N = 4$ e com $N = 6$, cujos resultados são mostrados na Figura 4 e na Figura 5, respectivamente. Para $N = 4$ as direções de chegada dos interferentes foram -45° , -15° e $+45^\circ$ e para $N = 6$ foram -45° , -15° , $+15^\circ$, $+45^\circ$ e $+60^\circ$.

Para o caso não subparametrizado ($N = 4$), os três algoritmos apresentam bons resultados para mitigação de interferentes, resultando em SER alta apenas no entorno das direções de chegada dos sinais interferentes. Entretanto, o NCMA-WL e o NCMA-WL-R apresentam melhor desempenho já que a SER apresentada nesses casos é alta para menos valores de direção de chegada de s_D . Já no caso subparametrizado ($N = 6$), o uso do NCMA-SL não implica em mitigação de interferentes uma vez que os resultados de SER obtidos estão sempre próximos da situação de erro máximo. Porém, o uso do NCMA-WL e do NCMA-WL-R ainda leva a bons resultados de mitigação de interferentes. Salienta-se que os resultados obtidos com o NCMA-WL-R equiparam-se aos obtidos com o NCMA-WL, usando menos recursos lógicos e com menor latência.

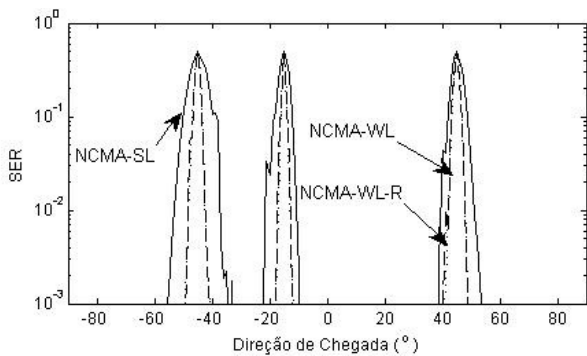


Fig. 4. SER para o sinal s_D obtida para situação em que 3 interferentes incidem sobre o arranjo ($N = 4$).

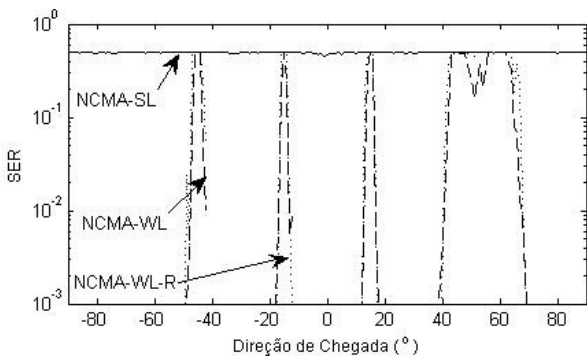


Fig. 5. SER para o sinal s_D obtida para situação em que 5 interferentes incidem sobre o arranjo ($N = 6$).

V. CONCLUSÕES

A partir de uma plataforma de hardware, utilizando-se a técnica de cossimulação, apresentou-se implementações práticas em FPGA de algoritmos adaptativos da família NCMA. Aliado a isto, realizou-se o cálculo do custo computacional das implementações dos algoritmos e da latência relacionada a cada um deles. Para verificação de seu desempenho emulou-se a tarefa de mitigação de interferentes em arranjos de antenas para cenário relacionado à mitigação de interferentes no contexto de *beamforming*, em casos não subparametrizado e subparametrizado. Em ambos os casos os algoritmos largamente lineares apresentaram desempenho superior ao algoritmo estritamente linear. Finalmente, no caso particular de sinais reais, o algoritmo largamente linear otimizado para esse tipo de sinal apresentou desempenho comparável à sua contrapartida não otimizada utilizando menos recursos de lógica e menor latência. Dados os resultados promissores mostrados neste artigo, tem-se como perspectivas futuras o teste dos algoritmos implementados em um arranjo de antenas real.

REFERÊNCIAS

- [1] T. S. Rappaport, "Wireless Communications: Principles and Practice," Prentice Hall, 1996
- [2] P. Barton, "Digital Beam Forming for Radar," IEE Proc. Communications, Radar and Signal Processing, v. 127, n. 4, pp. 266-277, Agosto 1980.
- [3] G. G. Rayleigh e J. M. Cioffi, "Spatio-temporal coding of wireless communication," IEEE Trans. on Communications, v. 46, pp. 357-366, Março 1998.
- [4] F. Ahmad, Y. Zhang e M. G. Amin, "Three Dimensional Wideband Beamforming for Imaging Through a Single Wall," IEEE Geoscience and Remote Sensing Letters, vol. 5, n. 2, pp. 176-179, Abril 2008.
- [5] B. Smolders e G. Hampson, "Deterministic RF Nulling in Phased Arrays for Next Generation of Radio Telescopes," IEEE Ant. Prop., vol. 44, n. 4, pp. 13-22, Agosto 2002.
- [6] J. A. TenCate et al., "Beamforming on Seismic Interface Waves With an Array of Geophones on the Swallow Sea Floor," IEEE Journal of Oceanic Eng., vol. 20, n. 4, pp. 300-310, Outubro 1995.
- [7] E. J. Bond et al., "Microwave Imaging via Space-Time Beamforming for Early Detection of Breast Cancer," IEEE Trans. on Ant. Prop., vol. 51, n. 8, pp. 1690-1705, Agosto 2003.
- [8] K. Hilal e P. Duhamel, "A convergence study of the constant modulus algorithm leading to a normalized-CMA and a block-normalized-CMA," Signal Processing VI: Theory and Applications, pp. 135-138, 1992.
- [9] A. Chinatto, C. Junqueira e J. M. T. Romano, "Optimized blind algorithms for widely linear beamforming," 7th International Telecommunications Symposium 2010, Setembro 2010.
- [10] P. Chevalier e B. Picinbono, "Widely Linear Estimation with Complex Data", IEEE Trans. on Signal Processing, v. 43, n. 8, pp 2030-2033, Agosto 1995.
- [11] P. Chevalier, "Optimal array processing for non stationary signals," Proc. ICCASP 1996, pp. 2868-2871, Maio 1996.
- [12] B. Picinbono, "On circularity", IEEE Trans. on Signal Processing, vol. 42, n. 12, pp. 3473-3482, Dezembro 1994.
- [13] D. N. Goddard, "Self-recovering equalization and carrier tracking in two-dimensional data communications system," IEEE Trans. on Communications, vol. COM-28, n. 11, Novembro 1980.
- [14] J. R. Treichler e B. G. Agee, "New approach to multipath correction of constant modulus signals," IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-31, n. 2, pp. 459-471, Abril 1983.
- [15] F. G. A. Neto, V. H. Nascimento e M. T. M. Silva, "Reduced-complexity widely linear adaptive estimation," 7th International Symposium on Wireless Communications Systems 2010, Setembro 2010.
- [16] The Xilinx Inc., www.xilinx.com.
- [17] The Mathworks Inc., www.mathworks.com.
- [18] http://www.xilinx.com/products/devkits/HW-V5-ML505-UNI-G.htm.
- [19] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. on Electronic Computing, vol. EC-8, pp. 330-334, Setembro 1959.