

O Algoritmo LMS no Domínio da Transformada *Wavelet* Aplicado a Filtros de Volterra

Carlos P. Bernal O., Juraci F. Galdino e José A. Apolinário Jr.

Resumo—Este artigo investiga o uso do algoritmo Least-Mean Square (LMS) no domínio da transformada *Wavelet* no cenário de filtros adaptativos não-lineares, implementados com uma série truncada de Volterra. Tal transformação visa aumentar a velocidade de convergência do algoritmo LMS, muito lenta neste tipo de aplicação, sem um proporcional aumento na complexidade computacional. A idéia básica que motiva tal abordagem é a característica especial das *Wavelets* de concentrar maior quantidade de energia em poucos coeficientes, possibilitando uma melhor ponderação do passo de adaptação bem como uma redução da ordem do vetor de coeficientes.

Palavras-Chave—Modelamento não-linear, Séries de Volterra, *wavelet*, Filtros Adaptativos.

Abstract—This paper investigates the use of the Least-Mean Square (LMS) Algorithm in the *Wavelet* transform domain in the scenario of non-linear adaptive filters implemented with truncated Volterra series. Such transformation aims an increase in the speed of convergence of the LMS algorithm, very slow for this particular application, without a proportional increase in the computational complexity. The basic idea behind this approach is the *Wavelet* special feature of concentrating a higher amount of energy in a few coefficients, making possible a better weighting of the step size as well as a reduction of the coefficient vector order.

Keywords—Nonlinear modeling, Volterra Series, *Wavelet*, Adaptive Filtering.

I. INTRODUÇÃO

A utilização de técnicas desenvolvidas para sistemas lineares em sistemas não-lineares tem sido uma prática muito comum devido, em grande parte, à complexidade das técnicas não-lineares. Uma maneira de trabalhar com não-linearidades usufruindo da difundida teoria já desenvolvida para os filtros adaptativos é o uso de Séries de Volterra [1]. Neste caso, é empregada uma versão truncada [2] da Série de Volterra para elaborar um núcleo (*Kernel*) que será inserido na estrutura de um filtro adaptativo convencional [3].

Diversos algoritmos de filtragem adaptativa tais como o LMS, o *Normalized LMS* (NLMS) e o *Recursive Least-Squares* (RLS) [3] podem ser empregados no contexto supramencionado. É conhecido que o algoritmo LMS tem uma menor complexidade computacional; por outro lado, sua velocidade de convergência é bem menor quando o sinal de entrada é fortemente correlacionado, fato indesejado em muitas aplicações.

Para tentar usufruir das características de baixa complexidade computacional e de robustez do LMS em problemas

que tipicamente envolvem uma lenta convergência, como é o caso de uma Série de Volterra, pode-se empregar uma transformação *Wavelet* [4], [5] no sinal de entrada do algoritmo para reduzir sua correlação, sem comprometer muito a complexidade computacional do algoritmo assim obtido em relação ao algoritmo LMS convencional.

Os filtros de Volterra apresentam um grande problema: o número total de coeficientes cresce exponencialmente com o aumento do tamanho dos retardos (memória) do filtro. Contribuindo neste sentido, o presente trabalho, além de apresentar uma nova abordagem não-linear de rápida convergência para o algoritmo LMS, emprega as propriedades e características da transformada *Wavelet* para reduzir o número de coeficientes, otimizando os núcleos dos filtros de Volterra.

Este artigo está dividido como segue: na Seção II, é feita uma descrição dos conceitos básicos das séries de Volterra e da transformada *Wavelet*; na Seção III, é feito um estudo do algoritmo LMS no domínio da transformada; na Seção IV, novos algoritmos LMS no domínio da transformada são propostos e seus desempenhos são apresentados. Por fim, na Seção V, são apresentadas as conclusões do trabalho.

II. CONCEITOS BÁSICOS

Esta seção tem como objetivo apresentar, de forma sucinta, duas técnicas importantes que serão utilizadas neste trabalho: os filtros adaptativos baseados em série truncada de Volterra e a transformada *Wavelet*.

A. Série de Volterra

Consideremos $x(n)$ e $y(n)$ como os sinais de entrada e saída, respectivamente, de um sistema não-linear causal e discreto no tempo. A expansão da série de Volterra usando $x(n)$ é dada por [2]

$$\begin{aligned}
 y(n) = & h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) \\
 & + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} h_2(m_1, m_2)x(n-m_1)x(n-m_2) + \dots \\
 & + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \dots \sum_{m_p=0}^{\infty} h_p(m_1, m_2, \dots, m_p)x(n-m_1)x(n-m_2) \\
 & \dots x(n-m_p) + \dots
 \end{aligned} \tag{1}$$

na qual, $h_p(m_1, m_2, \dots, m_p)$. É conhecida como a p -ésima ordem do núcleo do sistema de Volterra. Assumindo que os termos da expressão acima são simétricos, ou seja, que $x(n-m_i)x(n-m_j) = x(n-m_j)x(n-m_i)$, o núcleo de Volterra é formado considerando-se que $h_p(m_1, m_2, \dots, m_p)$ não sofre alterações para as $p!$ possíveis permutações dos índices de m_1, m_2, \dots, m_p .

Carlos P. Bernal O., Juraci F. Galdino e José A. Apolinário Jr.: Departamento de Engenharia Elétrica, IME, Rio de Janeiro, Brasil. E-mails: pbernal@ime.eb.br, galdino@ime.eb.br e apolin@ime.eb.br. Os autores agradecem à CAPES, à FAPERJ e ao CNPQ por viabilizarem esta pesquisa.

Observando que a série infinita em (1) não é adequada para utilização em filtragem adaptativa, deve-se trabalhar com suas extensões truncadas. Realizando o truncamento e assumindo que $h_0 = 0$, a equação (1) fica dada por:

$$\begin{aligned}
 y(n) = & \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) \\
 & + \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2(m_1, m_2)x(n-m_1)x(n-m_2) \\
 & + \dots \\
 & + \sum_{m_1=0}^{N-1} \dots \sum_{m_p=0}^{N-1} h_p(m_1, m_2, \dots, m_p)x(n-m_1) \dots x(n-m_p)
 \end{aligned} \quad (2)$$

na qual $N - 1$ representa a quantidade de elementos de retardo considerada na série de Volterra truncada.

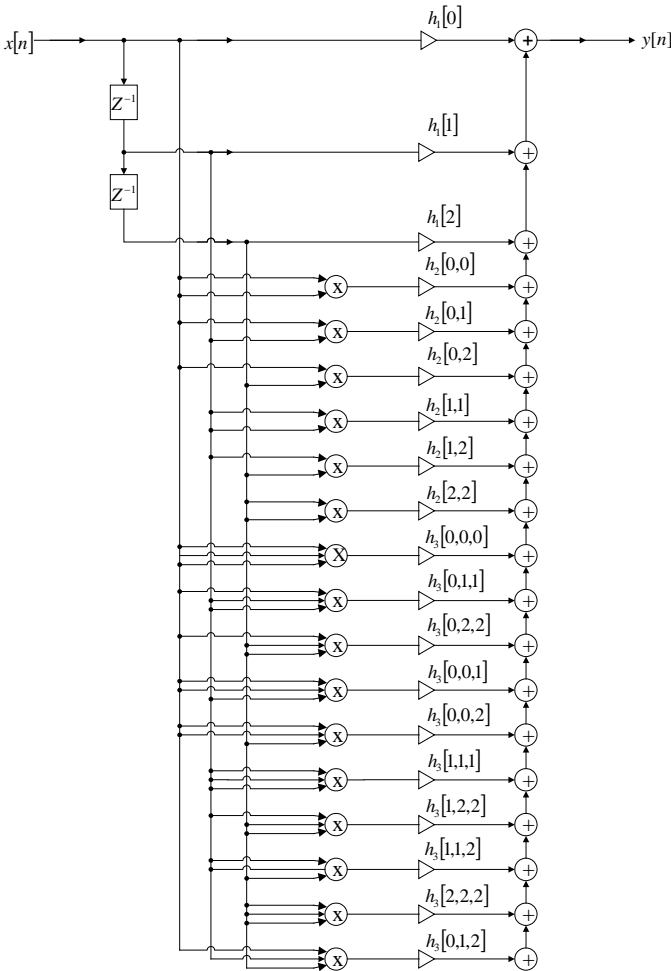


Fig. 1. Sistema de Volterra de ordem $p = 3$ e 2 (ou $N - 1$) elementos de retardo.

A Figura 1 apresenta o diagrama de uma série de Volterra truncada de ordem igual a 3 e com 2 elementos de retardo. Neste caso, é possível modelar um núcleo de Volterra que permita expressar o sinal de entrada de um filtro adaptativo pelo seguinte vetor [7]:

$$\begin{aligned}
 \mathbf{x}(n) = & [x(n), x(n-1), \dots, x(n-N+1), \\
 & x^2(n), x(n)x(n-1), \dots, x(n), x(n-N+1), \\
 & x^2(n-1), \dots, x^2(n-N+1), \\
 & x^3(n), x(n)x^2(n-1), \dots, x(n)x^2(n-N+1), \\
 & x^2(n)x(n-1), \dots, x^2(n)x(n-N+1), \dots, \\
 & x^3(n-N+1), x(n)x(n-1)x(n-2), \dots, \\
 & x(n-N-1)x(n-N)x(n-N+1)]^T
 \end{aligned} \quad (3)$$

A Figura 1 e a equação (3) possibilitam a compreensão de como se forma o Núcleo de Volterra. A estrutura deste vetor de entrada $\mathbf{x}(n)$ é a principal diferença para com os filtros lineares convencionais pois, ao invés de serem utilizadas somente amostras passadas do sinal de entrada, são utilizados também produtos entre amostras desse sinal tomadas no instante de tempo atual e passado.

B. Wavelets

Nas últimas décadas, a transformada *wavelet* tem sido objeto de investigação de muitos pesquisadores que atuam em diversas áreas do conhecimento humano. Em consequência desta investigação, a transformada *wavelet* vem sendo empregada com sucesso em um grande número de aplicações.

Um sistema *wavelet* é composto de um conjunto de funções de base que realizam a decomposição e análise de sinais pertencentes ao espaço de Hilbert, e que gozam de diversas propriedades, sendo a análise de multiresolução uma das mais importantes.

As funções de base *wavelets* decompõem sinais no plano tempo-frequência com variados graus de resolução nesses domínios. Para tal, dentre as funções de base que formam um típico sistema *wavelet*, estão incluídas funções de base que permitem obter excelente resolução no domínio do tempo (com pobre resolução no domínio da frequência, de acordo com o princípio da incerteza de Heisenberg), assim como estão incluídas outras que permitem obter excelente resolução no domínio da frequência (com pobre resolução no domínio do tempo). Assim sendo, a análise de todos os coeficientes que compõem a transformada *wavelet* de um determinado sinal, permite identificar, com boa fidelidade, tanto efeitos do sinal que ocorrem instantaneamente, quanto aqueles que se manifestam em determinadas frequências. Característica que não pode ser obtida com outras transformadas ou sistemas de expansão, como a Transformada de Fourier e as Transformadas de Fourier Janeladas.

Considerando que $\{\psi_{j,k}(t), j, k = 1, 2, \dots\}$ são as funções de base de um sistema ou transformada *wavelet*, um sinal $f(t)$ pertencente ao espaço de Hilbert pode ser representado da seguinte maneira:

$$f(t) = \sum_j \sum_k a_{j,k} \psi_{j,k}(t), \quad (4)$$

sendo $a_{j,k}$ os coeficientes da expansão *wavelets* dados por

$$a_{j,k} = \int f(t) \psi_{j,k}(t) dt. \quad (5)$$

Para a primeira geração de sistemas *wavelets*, todas as funções de base $\psi_{j,k}(t)$ são obtidas a partir de uma única

função de duração finita (estritamente limitada no tempo), denominada *wavelet* mãe, usando as operações de deslocamento e escalonamento [5]:

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in \mathbf{Z}. \quad (6)$$

Vale mencionar que aumentando j as funções de base ficam cada vez mais compactas no tempo. Combinando este aumento com a variação do índice k , efeitos presentes no sinal que são localizados no tempo vão sendo identificados com boa resolução, na medida que tais efeitos se manifestam na obtenção de coeficientes com amplitudes grandes $a_{j,k}$ para j elevados. Reduzindo j , por outro lado, efeitos do sinal localizados na frequência podem ser identificados com maior fidelidade analisando os respectivos coeficientes da transformada. Vale lembrar também que, para um determinado índice j , a coleção de coeficientes obtidos variando o índice k , representa a informação do sinal contida no j -ésimo nível de resolução da transformada.

Para sinais discretos no tempo, tanto o processo de síntese, equação (4), quanto o processo de análise, equação (5), são realizados de forma eficiente através do algoritmo piramidal, o qual emprega uma sucessão de filtros passa-baixas e passa-altas acompanhados de operação de decimação no processo de análise, e acompanhadas de operações de interpolação para o processo de síntese. As respostas ao impulso desses filtros são obtidas a partir das funções de base da transformada *wavelet* [5],[7], que neste caso é denominada de transformada *wavelet* discreta (DWT, do termo em inglês *Discrete Wavelet Transform*).

É importante mencionar que a transformada *wavelet* em geral, e a DWT em particular, realiza uma operação linear. Assim sendo, para sinais discretos no tempo e de duração finita, a DWT pode ser representada por uma matriz, a matriz de análise—doravante denotada por \mathbf{T} , cuja inversa, a matriz de síntese, é dada pela transposta da matriz de análise (\mathbf{T}^T) quando as funções de base que formam o sistema *wavelet* são ortonormais, caso de interesse neste trabalho.

Com a representação matricial da DWT, os coeficientes $a_{j,k}$ são organizados em forma vetorial, \mathbf{a} . Nesta arrumação, são dispostos inicialmente aqueles coeficientes associados ao menor nível de resolução da transformada, que aqui será representado por $j = 0$; em seguida, são dispostos os coeficientes para $j = 1$. Este processo continua até $j = M - 1$, sendo M a quantidade de níveis de resolução da DWT. Vale mencionar que, de acordo com a escolha da função de base e da quantidade de níveis de resolução, a quantidade de coeficientes de cada nível fica determinada, uma vez que a faixa de valores permitidos para o índice k fica especificada.

Os sinais de interesse prático apresentam fenômenos que ocorrem em maior intensidade em determinadas frequências e intervalos de tempo. Assim sendo, nesses casos, no domínio da transformada, a energia do sinal fica concentrada em determinados coeficientes. A concentração de energia no domínio da transformada *wavelet* ou da DWT ocorre para uma ampla classe de sinais de interesse prático, principalmente nos baixos níveis de resolução da transformada, ficando concentrada, em geral, nos níveis de maior resolução energias associadas aos

sinais espúrios subjacentes ao sinal de interesse. Esta é uma das mais importantes propriedades dessa transformada.

III. ALGORITMOS BASEADOS NO LMS

Quando o sinal de entrada de um filtro adaptativo, cujos coeficientes são ajustados pelo algoritmo LMS, apresenta correlação entre amostras consecutivas, o desempenho deste algoritmo tende a se degradar. Nesses casos, o algoritmo LMS pode apresentar baixa velocidade de convergência, principalmente à medida que as amostras do sinal de entrada tornam-se cada vez mais correlatadas.

Uma possibilidade para melhorar o desempenho do algoritmo LMS é realizar um processamento ou transformação linear nos dados de entrada no sentido de reduzir sua correlação e empregar o algoritmo LMS no domínio da transformada [6], [8]. Considerando que $\mathbf{x}(n)$ é o vetor que contém amostras do sinal no domínio do tempo como se mostra em (3) e que $\mathbf{s}(n)$ é o vetor que contém os coeficientes do domínio da transformada, tem-se que $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$.

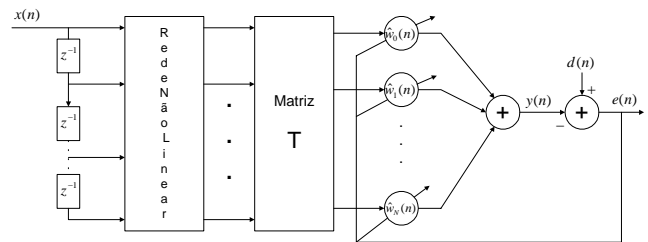


Fig. 2. Filtro adaptativo no domínio da transformada.

A Figura 2 apresenta um diagrama de blocos simplificado de um filtro não-linear adaptativo no domínio da transformada [3]. Nesta figura, $\hat{w}_j(n)$ representa o j -ésimo coeficiente do filtro adaptativo no domínio da transformada.

Muitas transformações lineares podem ser empregadas para se obter o algoritmo LMS no domínio da transformada. Neste trabalho, serão consideradas as DWT por três razões principais.

Em primeiro lugar, tais transformadas se aproximam da transformada de Karhunen-Loeve [7] para uma ampla classe de sinais. Como resultado disso, após a aplicação dessa transformada, as amostras na entrada do LMS serão bem menos correlatadas do que aquelas presentes no domínio do tempo $\mathbf{x}(n)$, contribuindo dessa forma para aumentar a velocidade de convergência do algoritmo.

Em segundo lugar, a transformada *wavelet* possui a propriedade de representação parcimoniosa, ou de concentração de energia em alguns coeficientes; assim sendo, é possível usar essa propriedade para reduzir a quantidade de coeficientes do filtro adaptativo, contribuindo para reduzir a complexidade computacional do algoritmo adaptativo.

Por fim, as matrizes que implementam as transformadas *wavelets* discretas são esparsas. Essa propriedade pode ser explorada para reduzir a complexidade na realização da transformada [6].

Outro procedimento que visa melhorar o desempenho do algoritmo LMS é a normalização do sinal de entrada no filtro

adaptativo. Essa normalização pode ser empregada tanto no domínio do tempo, quanto no domínio da transformada. Algoritmos LMS que empregam essa abordagem são denominados de LMS normalizados (NLMS).

No domínio do tempo, o algoritmo NLMS é dado por [3]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{2\mu e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}\mathbf{x}(n). \quad (7)$$

Ao passo que no domínio da transformada *wavelet*, o algoritmo LMS normalizado, doravante denominado de *Wavelet Transform Domain LMS* (WTD-LMS), é dado por

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + 2\mu e(n)\boldsymbol{\sigma}(n)\mathbf{s}(n), \quad (8)$$

na qual $\boldsymbol{\sigma}(n)$ é uma matriz diagonal cujo i -ésimo elemento da diagonal é dado por $[\gamma + \sigma_i^2(n)]^{-2}$, sendo σ_i^2 a potência de $s_i(n)$, que pode ser estimada por

$$\sigma_i^2(n) = \alpha s_i^2(n) + (1 - \alpha)\sigma_i^2(n-1), \quad (9)$$

sendo α um fator pequeno, $0 < \alpha \leq 0.1$, e γ uma constante pequena, empregada para evitar possíveis divisões por zero.

Vale lembrar que, para empregar o algoritmo LMS em aplicações que envolvem filtros de Volterra, é importante adotar passos distintos para cada ordem do polinômio de Volterra. Essa adoção é fundamental para garantir uma convergência mais rápida [3]. Para um polinômio de Volterra de ordem p , utiliza-se a matriz diagonal $\boldsymbol{\mu}_p$ que contém os valores de passo da seguinte forma:

$$\boldsymbol{\mu}_p = \text{diag}(\mu_1 \dots \mu_1 \mu_2 \dots \mu_2 \dots \mu_p \dots \mu_p) \quad (10)$$

Portanto, para o caso do algoritmo NLMS usado num núcleo de Volterra, sua equação de atualização pode ser dada pela seguinte expressão.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{2e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}\boldsymbol{\mu}_p\mathbf{x}(n), \quad (11)$$

No domínio da transformada *wavelet* não há razão para empregar passos distintos no sentido apresentado acima, uma vez que cada coeficiente, neste domínio, pode carregar informação de várias ordens do polinômio de Volterra devido à memória induzida pela matriz de análise (\mathbf{T}); mas, pode-se adotar um passo distinto para cada nível de resolução da DWT.

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + 2\boldsymbol{\mu}_M e(n)\boldsymbol{\sigma}(n)\mathbf{s}(n) \quad (12)$$

Sendo que, neste caso, $\boldsymbol{\mu}_M$ é uma matriz similar a $\boldsymbol{\mu}_p$ de (10) com cada elemento μ_j corresponde a cada um dos M níveis de decomposição *wavelet*.

A seguir, os algoritmos discutidos até aqui serão avaliados através de simulação computacional na identificação de um sistema com um núcleo de Volterra de terceira ordem com dois retardos [10] e um total de 19 coeficientes do filtro (o número de coeficientes é dado como em [11]).

Os passos adaptativos para os algoritmos dos filtros de Volterra LMS¹ são $\mu_1 = 0.001$, $\mu_2 = 0.003$ e $\mu_3 = 0.005$ e para o NLMS equação (8) são $\mu_1 = 0.003$, $\mu_2 = 0.006$ e

¹ $\mathbf{w}(n+1) = \mathbf{w}(n) + 2\boldsymbol{\mu}_p e(n)\mathbf{x}(n)$

$\mu_3 = 0.01$. Esses valores de passo foram obtidos experimentalmente ou sentido de aumentar a velocidade de convergência. Como mencionado anteriormente, cada um desses passos está associado a uma ordem do núcleo do filtro de Volterra.

Para o algoritmo WTD-LMS emprega-se a funções de base *wavelet* conhecidas como Daubechies, cujas funções *wavelets* possuem dois momentos nulos (Daubechies 2) [4], com três níveis de resolução.

Para realizar uma comparação justa com os algoritmos implementados no domínio do tempo, o algoritmo WTD-LMS também emprega três passos distintos, um para cada nível de resolução. Porém, devido à capacidade de concentração de energia nos níveis de menor resolução, os passos serão usados da seguinte maneira: no primeiro nível de resolução, que está associado às componentes de baixa frequência de $\mathbf{x}(n)$, será utilizado $\mu_0 = 0.005$; no segundo nível de resolução será empregado $\mu_1 = 0.003$ e $\mu_2 = 0.001$ no terceiro e último nível de resolução, o qual está associado às componentes de alta frequência de $\mathbf{x}(n)$.

Os resultados obtidos, em termos de Erro Médio Quadrático (ou MSE, do inglês *Mean Squared Error*) obtidos pelos algoritmos LMS, NLMS, WTD-LMS e RLS são mostrados na Figura 3, considerando uma razão sinal ruído de 30 dB.

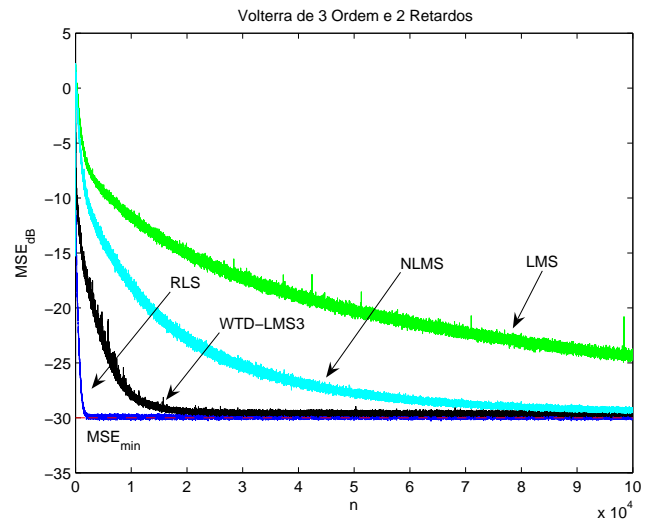


Fig. 3. Curvas de Erro Médio Quadrático

Como pode-se observar na Figura 3, dentre os algoritmos investigados, o que apresenta as piores características de desempenho é o LMS. O algoritmo NLMS apresenta velocidade de convergência um pouco melhor do que a apresentada pelo algoritmo LMS convencional. De forma diferente dos anteriores, o algoritmo WTD-LMS (com uma indicação "3" para referir-se aos $M = 3$ valores de μ em $\boldsymbol{\mu}_p$) apresenta uma alta velocidade de convergência para o valor de MSE mínimo (-30 dB); superada apenas por aquela obtida pelo algoritmo RLS. Porém, vale mencionar que o algoritmo RLS apresenta uma complexidade computacional bem superior aos demais algoritmos baseados no LMS, além de não possuir uma estabilidade numérica adequada.

IV. NOVAS PROPOSTAS

Até o momento, estamos somente usando a transformação proposta em [8] num filtro de Volterra e com uma certa dificuldade em otimizar M valores de passo. Como sabemos, as propriedades de convergência e o nível de erro médio quadrático obtido pelo algoritmo LMS, para uma dada entrada, dependem da escolha de seu passo. De modo geral, passos com valores pequenos acarretam lenta convergência porém proporcionam baixos valores de MSE na condição de regime permanente; por outro lado, valores elevados desses parâmetros produzem velocidades de convergência maiores para valores de MSE maiores.

Assim sendo, a criteriosa escolha do passo do algoritmo LMS é importante para provêr um bom comportamento entre velocidade de convergência e nível de MSE na condição de regime permanente. Mesmo nos algoritmos LMS normalizados essa questão é importante.

Cabe mencionar que, para uma mesma aplicação, a escolha apropriada do passo do algoritmo LMS depende da Relação Sina-Ruído (ou SNR, do inglês *Signal-to-Noise Ratio*). Em linhas gerais, para elevados valores de SNR pode-se adotar valores elevados para o passo. Porém, a medida que reduz-se essa razão, é importante usar valores de passo menores para controlar o efeito de ruído do gradiente [12].

A dependência da escolha do valor do passo com a SNR e a característica de concentração de energia promovida pela transformada *wavelet*, geralmente nos baixos níveis de resolução, podem ser explorados para estabelecer um critério apropriado para definir um valor de passo diferente para cada nível de resolução da transformada.

Seja $\mu_j(n)$ e $E_j(n) = \sum_{i=i_1}^{i_2} |s_i|^2$ o passo do LMS e a energia dos coeficientes *Wavelets* referentes ao j -ésimo nível de resolução do n -ésimo bloco de dados, respectivamente. O algoritmo proposto estabelece os passos da seguinte maneira:

$$\mu_j(n) = \frac{E_j(n)}{\sum_{i=0}^{M-1} E_i(n)} \mu = \frac{\mathbf{s}_j^T(n) \mathbf{s}_j(n)}{\mathbf{s}^T(n) \mathbf{s}(n)} \mu \quad (13)$$

sendo M a quantidade de níveis de resolução e μ o único parâmetro que deverá ser ajustado para garantir a convergência do algoritmo. Vale mencionar que tanto o denominador, quanto o numerador de (13) podem ser obtidos sem considerável complexidade computacional, uma vez que esses parâmetros são também empregados na normalização.

Os passos definidos em (13) podem ser apresentados em forma vetorial da seguinte maneira $\boldsymbol{\mu}_M = \text{diag}(\mu_0 \dots \mu_0 \mu_1 \dots \mu_1 \dots \mu_{M-1} \dots \mu_{M-1})$ na qual assumiu-se M níveis de resolução. É importante destacar três importantes propriedades do algoritmo proposto. A primeira é que quanto maior for a energia do nível, maior será o valor do passo e a SNR do nível, uma vez que a energia do ruído é diluída igualmente por todos os níveis da transformada. Vale lembrar que com o aumento da SNR é apropriado aumentar o valor do passo para melhorar as características de desempenho do algoritmo. A segunda é que os valores de passo mudam para cada bloco de dados: portanto, se a distribuição de energia ao longo dos níveis muda com o tempo, isso será refletido na especificação dos valores dos passos. A terceira é que, apesar

TABELA I

THE WAVELET TRANSFORM-DOMAIN LMS ALGORITHM.

WTD-LMS Algorithm	
Initialization	
$\mathbf{x}(0) = \hat{\mathbf{w}}(0) = [0 \dots 0]^T$	
$\gamma = \text{small constant}$	
$0 < \alpha \leq 0.1$	
for each n	
{	
$\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$	
$e(n) = d(n) - \mathbf{s}^T(n) \hat{\mathbf{w}}(n)$	
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + 2\boldsymbol{\mu}_j(n) e(n) \boldsymbol{\sigma}(n) \mathbf{s}(n)$	
}	

do emprego de diversos valores de passo (no caso em questão 3, por ter sido empregada uma transformada *wavelet* com apenas 3 níveis de resolução), apenas um parâmetro precisa ser obtido via simulação computacional (μ), isto reduz bastante a dificuldade de ajuste em relação a um algoritmo genérico WDT-LMS que emprega diferentes passos para diferentes níveis de resolução e diferentes blocos de dados. Na Tabela I, é apresentado de forma compacta o algoritmo WTD-LMS proposto.

O desempenho do algoritmo proposto será comparado com outros dois algoritmos WTD-LMS. O primeiro deles, que será aqui denominado de WTD-LMS1, emprega apenas um passo μ , cujo valor foi obtido empiricamente; o segundo, que será aqui denominado de algoritmo WTD-LMS3, emprega três passos fixos ($\mu_0 = 0,005$, $\mu_1 = 0,003$ e $\mu_2 = 0,001$). Os resultados de desempenho são mostrados na Figura 4. Na qual, além do desempenho desses algoritmos, é apresentado também o desempenho do LMS convencional.

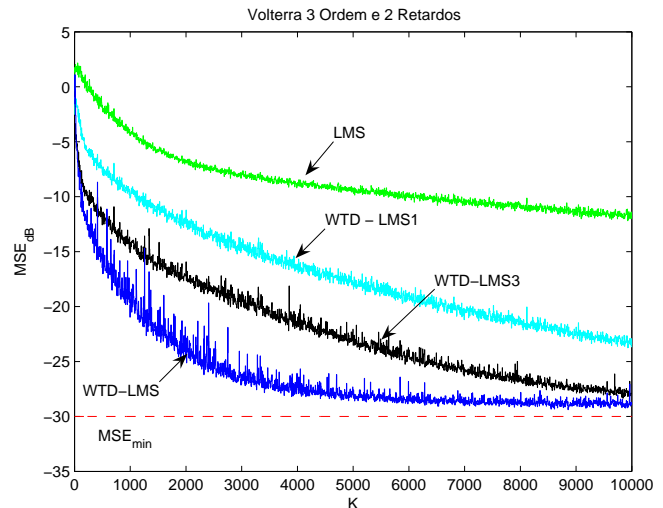


Fig. 4. Curva de Erro Médio Quadrático.

Os resultados apresentados na Figura 4 mostram claramente que, dentre os algoritmos baseados no LMS, o que apresenta maior velocidade de convergência é o WTD-LMS proposto. Além disto, podemos ressaltar que os algoritmos denominados WTD-LMS e WTD-LMS1 ou WTD-LMS3 possuem basicamente a mesma complexidade computacional.

Como mencionado anteriormente, a transformada *Wavelet* possui a propriedade de concentração de energia nos baixos níveis de resolução. Assim sendo, pode-se obter um algoritmo WTD-LMS com uma quantidade de coeficientes menor: apenas os coeficientes associados aos primeiros níveis de resolução são atualizados pelo algoritmo LMS, reduzindo, dessa forma, a complexidade computacional do algoritmo em relação ao WTD-LMS proposto. Essa versão do algoritmo proposto será doravante denominada de *Reduced Order WTD-LMS*, (RO WTD-LMS).

Vale mencionar que o algoritmo RO WTD-LMS deve conduzir a uma solução sub-ótima em função da energia do sinal de entrada que é desconsiderada na atualização dos coeficientes do algoritmo LMS. Quanto maior a concentração de energia nos baixos níveis de resolução, menor será essa degradação. Portanto, a apropriada escolha das funções de base *wavelets* e a quantidade de níveis de resolução exercem forte influência no desempenho desse novo algoritmo.

Será avaliado o desempenho do RO WTD-LMS, no qual empregou-se a transformada de Haar para realizar a DWT com 3 níveis de resolução, sendo que os coeficientes referentes ao último nível de resolução foram desconsiderados na equação do algoritmo. Além disso, o desempenho deste algoritmo será comparado com o de um outro também desenvolvido no sentido de reduzir a ordem do núcleo de Volterra; trata-se do filtro de Volterra adaptativo parcialmente interpolado denominado VPI-LMS proposto em [13]. Para este algoritmo empregou-se $L = 2$, resultando numa redução de 6 coeficientes dentre os de terceira ordem. Com essas especificações ambos os algoritmos empregam 13 coeficientes em suas equações recursivas. Os resultados obtidos são apresentados na Figura 5.

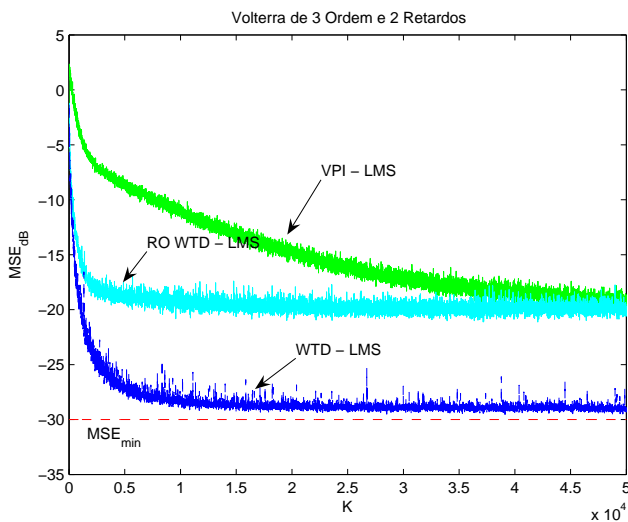


Fig. 5. Curva do Erro Médio Quadrático.

Como se pode observar, os algoritmos RO WTD-LMS e VPI-LMS apresentam o mesmo valor de MSE de regime permanente; porém, a velocidade de convergência do RO WTD-LMS é bem maior do que a do algoritmo VPI-LMS.

V. CONCLUSÕES

Neste trabalho, foram estudados os filtros Volterra adaptativos no domínio da transformada *Wavelet*. Dos resultados das simulações, pode-se concluir que o algoritmo aqui proposto, o WTD-LMS, para uma aplicação em filtro de Volterra, tem velocidade de convergência maior que os algoritmos Volterra LMS e Volterra NLMS, mantendo a mesma robustez numérica dos algoritmos da família LMS.

Além disto, ele apresenta uma menor dificuldade de ajuste em relação a um algoritmo WTD-LMS genérico por necessitar apenas um único μ ao invés de um ajuste de um passo ótimo para cada nível de resolução.

Em adição, a versão do algoritmo proposto com redução de ordem do núcleo de Volterra, RO WTD-LMS, apresentou um desempenho melhor que o algoritmo VPI-LMS, para a mesmo número de coeficientes.

Diante dos resultados obtidos, conclui-se que os algoritmos aqui propostos apresentam boas características de desempenho (velocidade de convergência, estabilidade numérica e complexidade computacional), tornando-os especialmente atrativos para utilização em filtros adaptativos de Volterra onde essas características de desempenho são extremamente importantes. Pretende-se continuar as pesquisas contemplando aplicações cujo filtro ótimo seja esparso, ou seja, apresenta vários coeficientes nulos ou quase nulos; neste trabalho, abordamos somente filtros com características dispersas.

REFERÊNCIAS

- [1] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley & Sons Ltd., 1980.
- [2] V. John Mathews. *Adaptive Polynomial Filters*, IEEE Signal Processing Magazine, Vol 6, pp. 10-26, 1991.
- [3] Diniz, Paulo S. R. *Adaptive Filtering Algorithms and Practical Implementation*, 2nd Edition, Kluwer Academic Publishers, Boston, 2002.
- [4] Daubechies, I. *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992. Notes From the 1990 CBMS-NSF Conference on *Wavelets* and Applications at Lowell, MA.
- [5] Burrus C. S.; Gopinath R. A. Gua H. *Introduction to Wavelets and Wavelet Transforms - A Primer*, Prentice Hall New Jersey, 1998.
- [6] Attallah S.; *The Wavelet Transform-Domain LMS Algorithm: A More Practical Approach*, IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol 47, No. 3, pp. 209-213, March 2000.
- [7] S. Mallat, *A Wavelet Tour of Signal Processing*, Second Edition, Academic Press, 1998.
- [8] Hosur S.; Tewfik A. H.; *Wavelet Transform Domain Adaptive FIR Filtering*, IEEE Trans. on Signal Processing, Vol 42, No. 3, pp. 572-585, March 1994.
- [9] Tewfik A. H.; Kim M.; *Fast Positive Linear System Solvers*, IEEE Trans. on Signal Processing, Vol 45, No. 3, pp. 617-630, March 1997.
- [10] Chang S.-L.; Ogunfunmi T.; *Stochastic Gradient Based Third-Order Volterra System Identification by Using Nonlinear Wiener Adaptive Algorithm*, IEE Proc-Vis Image Signal Processing, Vol 150, No. 2, pp. 90-98, April 2003.
- [11] Tan L.; Jiang J. *Adaptive Volterra Filters for Active Control of Nonlinear Noise Processes*, IEEE Trans. on Signal Processing, Vol 49, No. 8, pp. 1667-1676, August 2001.
- [12] Galdino, J.F.; Pinto, E.L.; de Alencar, M.S.; *Analytical Performance of the LMS Algorithm on the Estimation of WSS Channels*, IEEE Trans. on Commun., Vol 52, No. 6, pp. 982 - 991, June 2004.
- [13] Batista E. L. O., Tobias O. J., Seara R., *Filtros Volterra Adaptativos: Interpolado e Parcialmente Interpolado*, Simpósio Brasileiro de Telecomunicações-SBT'04, Vol. 48, No. 2, pp. 409-417, September 2004.