

MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
CURSO DE MESTRADO EM ENGENHARIA ELÉTRICA

CARLOS PAÚL BERNAL OÑATE

ALGORITMOS LMS NO DOMÍNIO DA TRANSFORMADA  
*WAVELET*: NOVAS PROPOSTAS, ANÁLISE E APLICAÇÃO EM  
FILTRAGEM DE VOLTERRA

Rio de Janeiro  
2005

INSTITUTO MILITAR DE ENGENHARIA

CARLOS PAÚL BERNAL OÑATE

**ALGORITMOS LMS NO DOMÍNIO DA TRANSFORMADA *WAVELET*:  
NOVAS PROPOSTAS, ANÁLISE E APLICAÇÃO EM FILTRAGEM DE  
VOLTERRA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof. José Antonio Apolinário Jr. - D. Sc.

Co-orientador: Prof. Juraci Ferreira Galdino - D. C.

Rio de Janeiro  
2005

INSTITUTO MILITAR DE ENGENHARIA  
Praça General Tibúrcio, 80-Praia Vermelha  
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

B517a Bernal Oñate, Carlos Paúl  
Algoritmos LMS no Domínio da Transformada  
*Wavelet*: Novas Propostas, Análise e Aplicação em Fil-  
tragem de Volterra, Carlos Paúl Bernal Oñate.  
– Rio de Janeiro: Instituto Militar de Engenharia, 2005.  
75 p.:il, graf., tab.

Dissertação (mestrado) – Instituto Militar de Enge-  
nharia, Rio de Janeiro, 2005.

1. Algoritmos, modelo matemático. 2. Filtragem  
adaptativa LMS, Wavelet. I. Instituto Militar de Enge-  
nharia. II. Título.

CDD 511.8

**INSTITUTO MILITAR DE ENGENHARIA**  
**CARLOS PAÚL BERNAL OÑATE**  
**ALGORITMOS LMS NO DOMÍNIO DA TRANSFORMADA *WAVELET*:**  
**NOVAS PROPOSTAS, ANÁLISE E APLICAÇÃO EM FILTRAGEM DE**  
**VOLTERRA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Elétrica do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Orientador: Prof. José Antonio Apolinário Jr. - D. Sc.

Co-orientador: Prof. Juraci Ferreira Galdino - D. C.

Aprovada em 14 de Outubro de 2005 pela seguinte Banca Examinadora:

---

Prof. José Antonio Apolinário Jr. - D. Sc. do IME - Presidente

---

Prof. Juraci Ferreira Galdino - D. C. do IME

---

Prof. Marcílio Castro de Matos - D. C. do IME

---

Prof. Paulo Roberto Rosa Lopes Nunes - Ph. D. do IME

---

Prof. Paulo S. R. Diniz - Ph. D. da COPPE/UFRJ

Rio de Janeiro  
2005

A minha Mãe Grecia e em memoria de Papetor.

## AGRADECIMIENTOS

Al Profesor José Antonio Apolinário Jr. por la confianza que siempre tuvo en mí, por estos cuatro años como mentor dos en la graduación en la ESPE y por los dos restantes en el IME, mi eterna gratitud por su esfuerzo en intentar enseñarme sus conocimientos, en especial en que deduzca las ecuaciones.

Al Profesor Juraci Ferreira Galdino, por su caracter afable y por el sin número de bromas y basiles que me propinó, generalmente es al contrario, solo decirle muchas gracias por el tiempo que invirtio en este trabajo.

A todos los que conforman el SE-3 que de alguna manera ayudaron a este trabajo, en especial a los Profesores de la Pos-graduación.

A la Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por el apoyo financiero.

A los amigos que hice en el IME, los de mi promoción Diogo ("Muleque") Alessandra ("Concepción") e Roberto ("Robertoca"), gracias por las bromas. Los restantes Arturo ("Guaguazo") e Gleyson ("Político"), por la amistad y por los partidos de baloncesto. Al amigo Carlos, tipo mas encamador que conoci, por las buenas chumas que nos pegamos, para que se repitan en Ecuador, gracias.

A mis hermanos Rodrigo e Diana, por la confianza que siempre tuvieron en mí, ah sin olvidar al sobrino el mas nuevo integrante de la familia. A mis Tíos Oswaldo e Hugolino, por la ayuda que me brindaron, a mi Tía, a mis Abuelos maternos e mi Abuelita, por todo su cariño.

A todos los amigos y familiares que faltan de ser nombrados no los olvide mas quedaría muy largo, solo recordando a Ana por su forma de ser y sus decisiones.

A mi madre por haberle faltado estos dos años, mil disculpas por todos los problemas que afrontó sola. No me puedo olvidar el adiós que nunca le puede dar a Papetor, siempre lo llevaré por toda mi vida

Epigrafe “Dios hizo los abismos para que el ser humano comprendiera las montañas.”  
Platigórsky.

# SUMÁRIO

LISTA DE ILUSTRAÇÕES .....	9
LISTA DE TABELAS .....	10
LISTA DE ABREVIATURAS E SÍMBOLOS .....	11
<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 Introdução .....	14
1.2 Objetivo da Dissertação .....	15
1.3 Estado da Arte .....	15
1.4 Contribuições da Dissertação .....	16
1.5 Organização da Dissertação .....	17
<b>2 CONCEITOS BÁSICOS .....</b>	<b>18</b>
2.1 Definição do Problema .....	18
2.2 O Algoritmo LMS .....	20
2.3 O Algoritmo NLMS .....	21
2.4 Série de Volterra .....	22
2.5 Filtro de Volterra LMS .....	28
2.6 Filtro de Volterra LMS Interpolado .....	29
2.7 Filtro de Volterra LMS Parcialmente Interpolado .....	31
2.8 <i>Wavelets</i> .....	31
2.9 Matriz de Transformação <b>T</b> .....	35
2.10 Resumo do Capítulo .....	37
<b>3 ALGORITMOS LMS NO DOMÍNIO DA TRANSFORMADA <i>WAVELET</i></b>	<b>39</b>
3.1 Introdução .....	39
3.2 O Algoritmo WTD-LMS .....	39
3.3 Uma Nova Proposta para o Algoritmo WTD-LMS .....	45
3.4 Redução do Filtro de Volterra .....	47
3.5 Complexidade Computacional .....	49
3.6 Resumo .....	51

<b>4</b>	<b>ANÁLISES DOS ALGORITMO WTD-LMS</b> .....	52
4.1	Introdução .....	52
4.2	Crerios para a Análise .....	52
4.3	Análise na Média .....	53
4.4	Análise na Covariância do Algoritmo WTD-LMSM .....	60
4.5	Resumo .....	70
<b>5</b>	<b>CONCLUSÕES E COMENTÁRIOS FINAIS</b> .....	72
5.1	Conclusões .....	72
5.2	Trabalhos Futuros .....	73
5.3	Comentários Finais .....	73
<b>6</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	74

## LISTA DE ILUSTRAÇÕES

FIG.2.1	Configuração Básica de um Filtro Adaptativo. ....	19
FIG.2.2	Sistema de Volterra 3 Ordem .....	25
FIG.2.3	Decomposição do filtro de Volterra em blocos. ....	26
FIG.2.4	Filtro Volterra Adaptativo Interpolado. ....	31
FIG.2.5	Filtro Volterra Adaptativo Parcialmente Interpolado (somente no bloco de 3ª ordem) .....	33
FIG.2.6	Dois Estágios do Algoritmo Piramidal .....	36
FIG.3.1	Filtro adaptativo no domínio da transformada. ....	40
FIG.3.2	MSE dos Algoritmos WTD-LMSM e LMS. ....	42
FIG.3.3	MSE dos Algoritmos WTD-LMS3, LMS, NLMS e RLS. ....	44
FIG.3.4	Energia por Nível. ....	45
FIG.3.5	MSE dos Algoritmos WTD-LMS. ....	49
FIG.3.6	MSE dos Algoritmos RO WTD-LMS. ....	50
FIG.4.1	Modelo utilizado na análise. ....	53
FIG.4.2	Curva do $MSE \times \mu$ . ....	60
FIG.4.3	Curva do $MSE \times \mu$ . ....	61
FIG.4.4	Valores de $\mu$ .....	69
FIG.4.5	Valores de $\mu_{\max}$ das duas análises. ....	70
FIG.4.6	Valores de $\mu_{\max}$ das duas análises diferentes SNR. ....	71

## LISTA DE TABELAS

TAB.2.1	O algoritmo LMS. ....	21
TAB.2.2	O algoritmo NLMS ....	22
TAB.2.3	O algoritmo Volterra LMS ....	30
TAB.2.4	O algoritmo Volterra LMS Interpolado ....	32
TAB.2.5	O algoritmo Volterra LMS Parcialmente Interpolado ....	34
TAB.3.1	O Algoritmo <i>Wavelet Transform-Domain</i> LMS. ....	48
TAB.3.2	Complexidade computacional dos algoritmos investigados. ....	51

## LISTA DE ABREVIATURAS E SÍMBOLOS

### ABREVIATURAS

CG	-	Conjugate Gradient
<i>dB</i>	-	decibel
DCT	-	Discrete Cosine Transform
DFT	-	Discrete Fourier Transform
DWT	-	Discrete Wavelet Transform
FIR	-	Finite Impulse Response
IIR	-	Infinite Impulse Response
LMS	-	Least Mean Square
LS	-	Least Squares
MSE	-	Mean Square Error
NLMS	-	Normalized Least Mean Square
RLS	-	Recursive Least Squares
RPRF	-	Regular Perfect Reconstruction Filter
RSR	-	Razão Sinal Ruido
RO WTD-LMS	-	Reduced Order Wavelet Transform Domain-Least Mean Square
SNR	-	Signal-to-Noise Ratio
TDLMS	-	Transform Domain Least Mean Square
VPI-LMS	-	Volterra Parcialmente Interpolado Least Mean Square
WTD-LMS	-	Wavelet Transform Domain-Least Mean Square
WTD-LMSM	-	Wavelet Transform Domain-Least Mean Square M

## RESUMO

O algoritmo LMS tem provado sua versatilidade em um sem número de aplicações, falhando somente quando se trata de velocidade de convergência. Neste trabalho, numa tentativa de diminuir esta desvantagem, usamos uma transformação no vetor sinal de entrada de modo a descorrelatar seus elementos forçando o algoritmo a convergir mais rápido.

Nós empregamos uma transformada Wavelet no bojo de um algoritmo LMS no domínio da transformada, num cenário tipicamente não-linear. Uma nova abordagem foi então proposta, almejando uma melhor velocidade de convergência bem como um método eficiente para escolher o tamanho do passo ao mesmo tempo que mantendo uma limitada complexidade computacional.

Nesta dissertação, foi também abordada uma maneira de reduzir a complexidade computacional do algoritmo no domínio da transformada sob investigação por meio de uma diminuição na ordem do vetor de coeficientes do filtro.

## ABSTRACT

The LMS algorithm has proved its versatility in a number of applications, failing only when it comes to speed of convergence. In this work, in an attempt to overcome this drawback, we have used a transformation in the input signal vector in order to decorrelate its elements forcing the adaptive filter to converge faster.

We have employed a Wavelet transform in the framework of a Transform Domain LMS algorithm, in a typically non-linear scenario. A new approach was then proposed aiming an improved speed of convergence as well as an efficient method to choose the step size while, at the same time, keeping a limited increase in the computational complexity. An upper bound for the step size was obtained through an analysis of the proposed algorithm.

In this dissertation, it was also addressed a way to reduce the computational complexity of the transform domain algorithm under investigation by shortening the order of the filter coefficient vector.

# 1 INTRODUÇÃO

## 1.1 INTRODUÇÃO

A teoria de sistemas lineares permite usar um grande número de ferramentas matemáticas, modelos, interpretações, etc. Por outro lado, existem modelos determinísticos não-lineares que podem descrever, de forma simples, o comportamento de um grande número de fenômenos naturais extremamente complicados; como exemplos de tais fenômenos podemos citar turbulências em fluídos e condições atmosféricas (tempo). Além disso, esses modelos são extremamente importantes em diversas aplicações na área da engenharia elétrica tais como: controle ativo de processos não-lineares, cancelamento de eco acústico, identificação e redução de distorções em alto-falantes e pré-equalização de canais em comunicações via satélite.

Muitas análises e modelagens de sistemas não-lineares são realizadas, a bem da simplicidade computacional, por meio de aproximações que se baseiam na teoria de sistemas lineares. No entanto, em algumas aplicações, as não-linearidades não podem ser bem estudadas e analisadas usando esta abordagem. Nestes casos, o uso de ferramentas não-lineares pode proporcionar ganhos com respeito às ferramentas lineares, justificando a maior complexidade computacional das primeiras em relação às segundas.

Dentro do campo das não-linearidades, duas técnicas vem sendo usadas com sucesso na análise deste tipo de fenômenos: as Séries de Volterra e as Redes Neurais. Em particular, as séries de Volterra são usadas para modelar sistemas não-lineares por várias razões, uma delas sendo a possibilidade de empregá-las em estruturas clássicas de filtros adaptativos lineares (MATHEWS, 1991; DINIZ, 2002).

Vale mencionar que a filtragem adaptativa linear constitui-se, atualmente, num campo de grande importância na área de Processamento de Sinais. O uso de filtros adaptativos torna-se particularmente necessário em ambientes onde as características do sinal de interesse são desconhecidas ou quando as estatísticas deste ambiente variam com o tempo. Na literatura de processamento de sinais, existe uma grande quantidade de algoritmos para filtragem adaptativa cuja teoria já se encontra bastante desenvolvida (HAYKIN, 1996; DINIZ, 2002).

Um desses algoritmos é o LMS (Least Mean Square) o qual se caracteriza por ter uma baixa complexidade computacional, sendo uma qualidade desejada quando se trabalha com filtros de Volterra; por outro lado, tem a desvantagem de para sinais de entrada altamente correlacionados, apresentam convergência lenta. Para compensar esta deficiência, existem técnicas que podem ser empregadas, para ajudar a incrementar a velocidade de convergência; uma destas é a utilização de transformações para reduzir a correlação do sinal de entrada. Este abordagem será adotada neste trabalho .

## 1.2 OBJETIVO DA DISSERTAÇÃO

O objetivo desta dissertação é propor técnicas para melhorar o desempenho dos algoritmos adaptativos baseados no LMS, quando usado em aplicações que envolvem sinais que apresentam correlação e não-lineares, em particular quando tais sinais são modeladas por séries de Volterra.

Para levar a cabo esta tarefa, emprega-se a transformada *wavelet* para reduzir a correlação do sinal de entrada, contribuindo dessa maneira melhorar as características de convergência do algoritmo LMS, sem elevar o custo computacional. Além disso, pretende-se estabelecer os limites para a escolha do tamanho do passo adaptativo ( $\mu$ ).

## 1.3 ESTADO DA ARTE

A filtragem adaptativa não-linear, tem sido uma técnica pouco usada. Isto se dá, em grande parte, pela complexidade da sua implementação. Assim sendo, uma importante e atual linha de pesquisa nessa área é a busca ou o desenvolvimento de técnicas que visem a redução de complexidade de ferramentas não-lineares. A importância desse tema se justifica pela grande quantidade de aplicações onde essas ferramentas podem ser empregadas como em compensação de distorções não-lineares, cancelamento de eco não-linear, equalização de canais de comunicação via satélite (CHENG, 2001) e pré-distorção de canais não-lineares, além de muitas outras aplicações em variados tópicos do processamento digital de sinais (MATHEWS, 2000).

Em 1983, Narayan (NARAYAN, 1983) fez o primeiro trabalho sobre o algoritmo *Transform Domain Least Mean Square* (TDLMS), tendo empregado matrizes ortogonais para realizar a transformação: *Discrete Fourier Transform* (DFT) e *Discrete Cosine Transform* (DCT).

Com o decorrer dos anos e o surgimento de novas ferramentas matemáticas, tornou-se possível o emprego das *Wavelets* para gerar a matriz de transformação. Esta nova abordagem foi apresentada em (HOSUR, 1997), e obteve bons resultados com respeito às anteriores. Novas propostas para melhorar a complexidade computacional foram apresentadas em (ATTALLAH, 2000).

Por outro lado, as séries truncadas de Volterra, empregadas em sistemas não-lineares permitem a utilização de estruturas típicas de filtragem adaptativa; uma boa leitura sobre o assunto é encontrado em (MATHEWS, 1991). Novos trabalhos forem apresentados, mas, até onde concerne o meu conhecimento, persiste o problema da elevada complexidade computacional, devido à geração do núcleo (*kernel*) de Volterra cuja quantidade de coeficientes aumenta de forma exponencial com a ordem. Esta, por vezes altíssima quantidade de coeficientes pode tornar inviável a implementação de filtros adaptativos de convergência rápida.

Na atualidade, tem-se verificado que alguns processos até então modelados com o uso de ferramentas lineares, apresentam de fato parcelas não-lineares. Essa constatação impulsiona ainda mais o desenvolvimento de ferramentas não-lineares que empreguem algoritmos com boa velocidade de convergência, mesmo quando aplicados a sinais com elevado grau de correlação (ZANUY, 1998).

#### 1.4 CONTRIBUIÇÕES DA DISSERTAÇÃO

Nesta dissertação é feita uma avaliação do desempenho do algoritmo LMS no domínio da transformada *wavelet* num ambiente puramente não-linear, modelado por séries de Volterra. Com base em simulações, são apresentadas as melhores opções de algoritmos. Novos algoritmos no domínio da transformada *wavelet*, baseados no algoritmo LMS, são apresentados. Esses novos algoritmos caracterizam-se por apresentar maior velocidade de convergência sem aumentar muito a complexidade computacional em relação ao algoritmo LMS convencional.

Outra contribuição importante deste trabalho é a abordagem da escolha de um passo de adaptação adequado para este tipo de algoritmo. Isto é bastante relevante especialmente em sistemas não-lineares onde a escolha do passo é uma atividade crítica. Neste contexto, vale mencionar que foram desenvolvidos limitantes superiores para o valor de passo no sentido de garantir a convergência dos algoritmos desenvolvidos. Dois limitantes foram obtidos com base na análise na média e na covariância do vetor de erros nos coeficientes.

## 1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

A dissertação é composta de 5 Capítulos. No Capítulo 2, são apresentadas as ferramentas teóricas necessárias ao desenvolvimento da dissertação: conceitos básicos de filtragem adaptativa, de séries de Volterra e de adaptação no domínio de uma transformada, além de uma breve revisão sobre transformada *wavelet*. No Capítulo 3, é apresentado um novo algoritmo baseado no LMS no domínio da transformada *wavelet*. No Capítulo 4, é feita uma análise de convergência visando estabelecer critérios de escolha para o valor do passo. Conclusões e comentários finais são apresentados no Capítulo 5.

## 2 CONCEITOS BÁSICOS

Neste capítulo são abordados, de forma resumida, alguns conceitos básicos de assuntos importantes para o desenvolvimento desta tese. Na Seção 2.1 são apresentados importantes resultados de filtragem adaptativa tais como, a solução de Wiener. Também são apresentados os algoritmos LMS (*Least Mean Square*) e NLMS (*Normalized Least Mean Square*) nas Seções 2.2 e 2.3, respectivamente.

Da Seção 2.4 até a Seção 2.7, faz-se uma introdução sobre a teoria das série de Volterra em filtragem adaptativa e alguns algoritmos desenvolvidos para estas aplicações. Nas Seções 2.8 e 2.9 são apresentados conceitos básicos da transformada *wavelet*. Por fim na Seção 2.10, apresenta-se o resumo do capítulo.

### 2.1 DEFINIÇÃO DO PROBLEMA

Uma configuração básica de um filtro adaptativo discreto no tempo é apresentada na FIG. 2.1, na qual  $\mathbf{x}(n)$  representa o vetor do sinal de entrada,  $d(n)$  o sinal desejado,  $y(n)$  a saída do filtro adaptativo e  $e(n)$  o erro ou a diferença entre o sinal desejado e a saída do filtro. Este sinal de erro é empregado para ajustar o vetor de coeficientes do filtro,  $\mathbf{w}(n)$ , por meio de um algoritmo de adaptação.

O processo de adaptação dos coeficientes de um filtro adaptativo é realizado visando a minimização de uma determinada função custo estabelecida a partir do sinal de erro. A função custo de uso mais comum é o erro médio quadrático, MSE, do inglês *Mean-Square Error*, que é definida como

$$\xi(k) = E [|e(n)|^2], \quad (2.1)$$

na qual  $E[\cdot]$  denota o operador valor esperado,

$$e(n) = d(n) - y(n) \quad (2.2)$$

é o erro *a priori* (antes da atualização dos coeficientes) no instante  $n$ , e  $y(n)$  é dada por

$$y(n) = \sum_{j=1}^N w_j(n)x(n-j+1) = \mathbf{w}^T(n)\mathbf{x}(n). \quad (2.3)$$

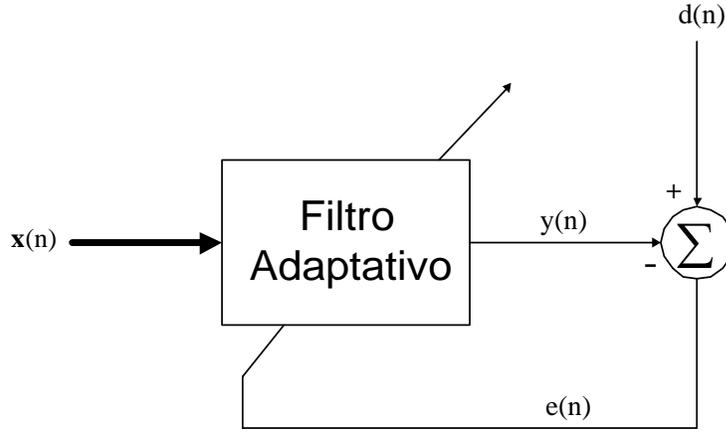


FIG. 2.1: Configuração Básica de um Filtro Adaptativo.

A expressão do erro *a priori* em função dos coeficientes do filtro adaptativo é dada por

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n). \quad (2.4)$$

Substituindo-se a EQ. 2.4 na EQ. 2.1 obtém-se

$$\begin{aligned} \xi(n) &= E \left[ \{d(n) - \mathbf{w}^T(n)\mathbf{x}(n)\}^2 \right] \\ &= E[d(n)^2] - 2E[d(n)\mathbf{x}^T(n)\mathbf{w}(n)] + E[\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)]. \end{aligned} \quad (2.5)$$

Denotando-se  $\mathbf{R}_{xx}$  como sendo a matriz de autocorrelação do sinal de entrada, e  $\mathbf{p}$  como sendo o vetor de correlação cruzada entre o sinal desejado e a entrada, ou seja

$$\mathbf{R}_{xx} = E[\mathbf{x}(n)\mathbf{x}^T(n)] \quad (2.6)$$

e

$$\mathbf{p} = E[d(n)\mathbf{x}(n)]; \quad (2.7)$$

e considerando o vetor de coeficientes,  $\mathbf{w}$  (determinístico) ao invés de  $\mathbf{w}(n)$ , a EQ. 2.5 torna-se

$$\xi = \sigma_d^2 - 2\mathbf{p}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}. \quad (2.8)$$

O gradiente da expressão acima em relação a  $\mathbf{w}$ , é dado por

$$\nabla_{\mathbf{w}}\xi = -2\mathbf{p} + 2\mathbf{R}_{xx}\mathbf{w}. \quad (2.9)$$

Igualando a última equação a zero e assumindo que a matriz  $\mathbf{R}_{xx}$  seja não singular, o vetor de coeficientes  $\mathbf{w}_0$  que minimiza a função custo é dado por

$$\mathbf{w}_0 = \mathbf{R}_{xx}^{-1}\mathbf{p}. \quad (2.10)$$

A EQ. 2.10 é conhecida como a solução de Wiener (HAYKIN, 1996). Na prática, porém, é mais comum a obtenção do vetor de coeficientes,  $\mathbf{w}$ , por meio de um algoritmo adaptativo. Isto ocorre tendo em vista que na pratica dificilmente se dispõe de  $\mathbf{R}_{xx}$  e  $\mathbf{p}$ .

## 2.2 O ALGORITMO LMS

O algoritmo LMS (*Least Mean Square*) continua sendo o mais usado em aplicações práticas por duas razões fundamentais: a sua simplicidade, implicando em um baixo custo computacional, e a garantia de sua convergência para uma escolha apropriada do seu passo de adaptação.

O algoritmo LMS é baseado no algoritmo *steepest-descent* cuja adaptação do vetor de coeficientes é dada por (HAYKIN, 1996; DINIZ, 2002):

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla_{\mathbf{w}}\xi(n), \quad (2.11)$$

onde  $\mu$  é denominado de *step-size* ou tamanho do passo de adaptação, cujo valor deve ser criteriosamente escolhido para garantir convergência do algoritmo (GALDINO, 2003).

Na prática, o gradiente  $\nabla_{\mathbf{w}}\xi(n)$  (ver EQ. 2.9), não pode ser obtido com exatidão, tendo em vista que  $\mathbf{p}$  e  $\mathbf{R}_{xx}$  não são conhecidas. Uma maneira de contornar este problema é usar estimativas.

Para a implementação do algoritmo LMS, emprega-se os seguintes estimadores despolarizados de  $\mathbf{p}$  e  $\mathbf{R}_{xx}$ :

$$\hat{\mathbf{p}} = \mathbf{x}(n)d(n) \quad (2.12)$$

e

$$\hat{\mathbf{R}}_{xx} = \mathbf{x}(n)\mathbf{x}^T(n). \quad (2.13)$$

TAB. 2.1: O algoritmo LMS.

LMS
Inicializar: $\mathbf{x}(0) = \mathbf{w}(0) = \mathbf{0}$ escolher $\mu$ no intervalo $0 < \mu < \frac{1}{\lambda_{\max}}$ for $n = 1, 2, \dots$ { $e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$ $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n)$ }

Para estes estimadores, pode-se obter uma estimativa despolarizada do gradiente como

$$\begin{aligned}
 \hat{\nabla}_{\mathbf{w}} \xi(n) &= -2\mathbf{x}(n)d(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) \\
 &= -2\mathbf{x}(n) [d(n) - \mathbf{x}^T(n)\mathbf{w}(n)] \\
 &= -2e(n)\mathbf{x}(n).
 \end{aligned} \tag{2.14}$$

Substituindo a estimativa da EQ. 2.14 na EQ. 2.11, obtemos a seguinte relação para a atualização do vetor de coeficientes correspondente ao algoritmo LMS

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n). \tag{2.15}$$

As equações referentes ao algoritmo LMS estão na TAB. 2.1, onde o passo de adaptação *step-size* incorpora o número 2, pois, na prática,  $\mu$  é inicializado apropriadamente para que não seja necessário multiplicá-lo por 2 a cada iteração.

### 2.3 O ALGORITMO NLMS

Um dos principais inconvenientes do algoritmo LMS é a baixa velocidade de convergência. Para tentar resolver este problema, muitos algoritmos de filtragem adaptativa baseados no LMS convencional vêm sendo propostos. Um desses algoritmos é o LMS normalizado, comumente denotado por NLMS, que emprega uma normalização do sinal de entrada.

A fim de melhorar a taxa de convergência do algoritmo LMS, adota-se no algoritmo NLMS um fator de convergência variável. O valor do  $\mu_n$  é escolhido tal que o erro a posteriori (após a atualização dos coeficientes) seja igual a zero, tal que  $\frac{\partial \Delta e^2(n)}{\partial \mu_n} = 0$ .

TAB. 2.2: O algoritmo NLMS

NLMS
Inicializar: $\mathbf{x}(0) = \mathbf{w}(0) = \mathbf{0}$ escolha $\mu_N$ no intervalo $0 < \mu_N \leq 2$ $\gamma =$ constante pequena for $n = 1, 2, \dots$ { $e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$ $\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_N}{\gamma + \mathbf{x}^T(n)\mathbf{x}(n)}e(n)\mathbf{x}(n)$ }

Resolvendo-se esta equação diferencial para  $\mu_n$  chega-se a:

$$\mu_n = \frac{1}{2\mathbf{x}^T(n)\mathbf{x}(n)}. \quad (2.16)$$

Considerando na EQ. 2.15, o uso de  $\mu_n$  em lugar de  $\mu$  e empregando-se na equação assim obtida a EQ. 2.16, chega-se a equação adotada pelo algoritmo LMS normalizado, que é dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}. \quad (2.17)$$

Usando um fator de convergência fixo  $\mu_N$ , introduzido na fórmula de atualização como controle do desajuste, e introduzindo um parâmetro  $\gamma$  a fim evitar possíveis divisões por zero, a equação de atualização dos coeficientes pode ser expressa por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_N}{\gamma + \mathbf{x}^T(n)\mathbf{x}(n)}e(n)\mathbf{x}(n). \quad (2.18)$$

O conjunto de equações do algoritmo NLMS é apreseto na TAB. 2.2 (DINIZ, 2002). Apesar de  $\mu_N$  poder assumir valores maiores que um e menores que dois, na prática, utiliza-se  $0 < \mu_N \leq 1$ .

## 2.4 SÉRIE DE VOLTERRA

Considerando  $x(n)$  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 (SCHETZEN, 1980; MATHEWS, 1991, 2000)

$$\begin{aligned}
y(n) = & h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n - m_1) \\
& + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} h_2(m_1, m_2)x(n - m_1)x(n - m_2) + \dots \\
& + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \dots \sum_{m_p=0}^{\infty} h_p(m_1, m_2, \dots, m_p)x(n - m_1)x(n - m_2) \\
& \dots x(n - m_p) + \dots
\end{aligned} \tag{2.19}$$

Na qual,  $h_p(m_1, m_2, \dots, m_p)$  é conhecida como a  $p$ -ésima ordem do núcleo do sistema de Volterra (SCHETZEN, 1980; MATHEWS, 2000). 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  $m_1, m_2, \dots, m_p$ .

Observando que a série infinita apresentada na EQ. 2.19 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 EQ. 2.19 fica dada por:

$$\begin{aligned}
y(n) = & \sum_{m_1=0}^{N-1} h_1(m_1)x(n - m_1) \\
& + \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2(m_1, m_2)x(n - m_1)x(n - m_2) \\
& + \dots \\
& + \sum_{m_1=0}^{N-1} \dots \sum_{m_p=0}^{N-1} h_p(m_1, m_2, \dots, m_p)x(n - m_1) \dots x(n - m_p),
\end{aligned} \tag{2.20}$$

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

A FIG. 2.2 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 (DINIZ, 2002):

$$\mathbf{x}(n) = \begin{bmatrix}
x(n) \\
x(n-1) \\
\vdots \\
x(n-N+1) \\
x^2(n) \\
x(n)x(n-1) \\
\vdots \\
x(n)x(n-N+1) \\
x^2(n-1) \\
\vdots \\
x^2(n-N+1) \\
x^3(n) \\
x(n)x^2(n-1) \\
\vdots \\
x(n)x^2(n-N+1) \\
x^2(n)x(n-1) \\
\vdots \\
x^2(n)x(n-N+1) \\
\vdots \\
x^3(n-N+1) \\
x(n)x(n-1)x(n-2) \\
\vdots \\
x(n-N-1)x(n-N)x(n-N+1)
\end{bmatrix}
\tag{2.21}$$

A FIG. 2.2 e a EQ. 2.21 possibilitam a compreensão de como se forma o Núcleo de Volterra. A estrutura deste vetor de entrada,  $\mathbf{x}(n)$ , é a principal diferença com relação aos filtros lineares convencionais pois, ao invés de serem utilizadas somente amostras passadas do sinal de entrada, são utilizados também produtos cruzados entre amostras atuais e passadas. Convém salientar que, à medida que se aumenta o número de retardos ou a

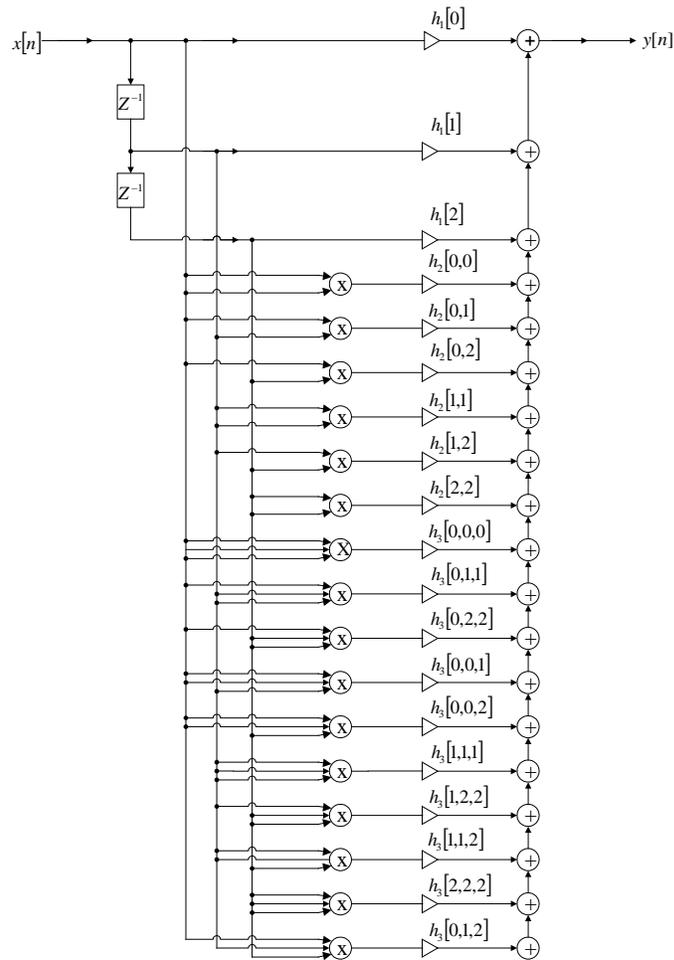


FIG. 2.2: Sistema de Volterra 3 Ordem

ordem do polinômio gerado pelos produtos cruzados, torna-se mais difícil a elaboração do núcleo, gerando uma estrutura de grande dimensão que pode inviabilizar a solução do problema.

Pode-se ressaltar que um grande número de algoritmos adaptativos (LMS, RLS, CG, QRD-RLS, etc.) pode ser aplicado para o ajuste dos parâmetros da estrutura (DINIZ, 2002; CHANG, 2000).

O filtro de Volterra pode ser decomposto em uma estrutura paralela de blocos, como ilustrado na FIG. 2.3, onde cada um dos blocos corresponde a um filtro de ordem  $p$ . Essa decomposição é bastante vantajosa do ponto de vista prático, pois em algumas aplicações, certos blocos podem ser tratados de maneira diferenciada ou mesmo desconsiderados.

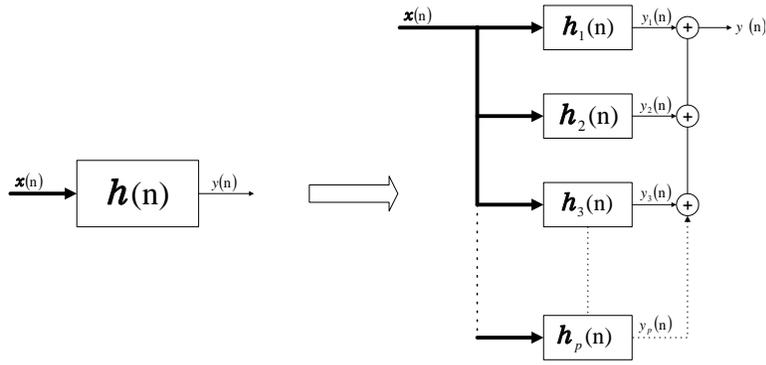


FIG. 2.3: Decomposição do filtro de Volterra em blocos.

A saída de cada um dos blocos do filtro de volterra pode ser escrita na forma vetorial. Essa abordagem é bem conhecida para o bloco de primeira ordem (OPPENHEIM, 1999) resultando em

$$y_1(n) = \mathbf{h}_1^T \mathbf{x}_1(n) \quad (2.22)$$

com  $\mathbf{x}_1(n)$  representando o vetor de entrada de primeira ordem definido como  $[x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$ , e  $\mathbf{h}_1$  o vetor dos coeficientes. Para os demais blocos, é possível obter representações análogas à apresentada na EQ. 2.22 considerando-se uma operação vetorial pseudo-linear (BATISTA, 2004). Por exemplo, para um bloco de segunda ordem pode-se escrever o vetor de entrada e o de coeficiente como segue:

$$\mathbf{x}_2(n) = \begin{bmatrix} x^2(n) \\ x(n)x(n-1) \\ \vdots \\ x(n)x(n-N+1) \\ x^2(n-1) \\ \vdots \\ x^2(n-N+1) \end{bmatrix}, \quad (2.23)$$

$$\mathbf{h}_2 = \begin{bmatrix} h_2(0, 0) \\ h_2(0, 1) \\ \vdots \\ h_2(0, N-1) \\ h_2(1, 1) \\ \vdots \\ h_2(N-1, N-1) \end{bmatrix} \quad (2.24)$$

resultando em uma relação de entrada e saída dada por

$$y_2(n) = \mathbf{h}_2^T \mathbf{x}_2(n). \quad (2.25)$$

Operação similar pode ser realizada para os outros blocos não-lineares do filtro Volterra, resultando na seguinte relação de entrada e saída para um bloco de uma ordem  $p$  qualquer:

$$y_p(n) = \mathbf{h}_p^T \mathbf{x}_p(n). \quad (2.26)$$

Com respeito à complexidade computacional requerida, o filtro Volterra apresenta um sério problema. O número de coeficientes cresce exponencialmente com o aumento do tamanho da memória. Se, denotamos  $D_{p_b}(N)$  como sendo o número de coeficientes de cada bloco de ordem  $p_b$  de um filtro Volterra com tamanho de memória  $N$ , tem-se (TAN, 2001):

$$D_{p_b}(N) = \frac{(N + p_b - 1)!}{(N - 1)! p_b!}. \quad (2.27)$$

Considerando  $D_V(N, p)$  como o número total de coeficientes de um filtro de Volterra de ordem  $p$  e tamanho de memória  $N$ , o número total de coeficientes é dado pelo somatório dos coeficientes de cada um dos blocos. Assim,

$$D_V(N, p) = \sum_{p_b=1}^p D_{p_b}(N) \quad (2.28)$$

o que resulta em

$$D_V(N, p) = \frac{(N + p)!}{N! p!} - 1. \quad (2.29)$$

## 2.5 FILTRO DE VOLTERRA LMS

Levando em consideração a EQ. 2.21 e a FIG. 2.2, pode-se definir o sinal  $y(n)$  como :

$$\begin{aligned}
 y(n) &= \sum_{m_1=0}^{N-1} w_{01}(m_1)x(n - m_1) \\
 &+ \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} w_{02}(m_1, m_2)x(n - m_1)x(n - m_2) \\
 &+ \dots \\
 &+ \sum_{m_1=0}^{N-1} \dots \sum_{m_p=0}^{N-1} w_{0p}(m_1, m_2, \dots, m_p)x(n - m_1) \dots x(n - m_p)
 \end{aligned} \tag{2.30}$$

considerando o vetor de coeficientes  $\mathbf{w}(n)$  definido como :

$$\mathbf{w}(n) = \begin{bmatrix}
w_0(n) \\
w_1(n) \\
\vdots \\
w_{n-N+1}(n) \\
w_{0,0}(n) \\
w_{0,1}(n) \\
\vdots \\
w_{0,n-N+1}(n) \\
w_{1,1}(n) \\
\vdots \\
w_{n-N+1,n-N+1}(n) \\
w_{0,0,0}(n) \\
w_{0,1,1}(n) \\
\vdots \\
w_{0,n-N+1,n-N+1}(n) \\
w_{0,0,1}(n) \\
\vdots \\
w_{0,0,n-N+1}(n) \\
\vdots \\
w_{n-N+1,n-N+1,n-N+1}(n) \\
w_{0,1,2}(n) \\
\vdots \\
w_{n-N-1,n-N,n-N+1}(n)
\end{bmatrix} \tag{2.31}$$

podemos reescrever as equações da Seções 2.2 na TAB. 2.3. Observe que usamos um passo para cada ordem como em (DINIZ, 2002).

## 2.6 FILTRO DE VOLTERRA LMS INTERPOLADO

Como se observa na Seção 2.4, reduzindo-se o tamanho da memória de um filtro de Volterra, obtém-se uma expressiva redução do número de coeficientes. Isso pode ser

TAB. 2.3: O algoritmo Volterra LMS

<b>Volterra LMS</b>	
Inicializar:	$\mathbf{x}(0) = \mathbf{w}(0) = \mathbf{0}$
	escolher $\mu_i$ no intervalo $0 < \mu_i < \frac{1}{\lambda_{\max}}$ , para $i = 1, 2, \dots, p$
for $n = 1, 2, \dots$	{
	$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$
	$\mathbf{w}(n+1) = \mathbf{w}(n) + \begin{bmatrix} \mu_1 \mathbf{I}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_2 \mathbf{I}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_p \mathbf{I}_p \end{bmatrix} e(n)\mathbf{x}(n)$
	}

obtido através do uso de um filtro Volterra adaptativo esparsos em conjunto com um filtro interpolador de entrada. A FIG. 2.4 mostra a estrutura proposta para o filtro de Volterra adaptativo interpolado, usando o algoritmo LMS no processo de adaptação (BATISTA, 2004).

Nesta figura,  $\mathbf{h}_{V_i}$  representa o filtro Volterra adaptativo esparsos, enquanto o interpolador é representado por um filtro FIR  $\mathcal{I} = [i_0, i_1, \dots, i_{M-1}]^T$  (FERMO, 2000), com  $M$  coeficientes. O sinal de entrada e o sinal de entrada interpolada são denotados por  $\mathbf{x}(n)$  e  $\mathbf{x}_i(n)$ , respectivamente, sendo

$$x_i = \sum_{j=0}^M i_j x(n-j), \quad (2.32)$$

Como em (BATISTA, 2004),  $L$  é usado para definir o fator de interpolação. Ele determina o grau de esparsidade do filtro Volterra. A entrada do filtro esparsos é obtida retirando  $(L-1)$  amostras de cada  $L$  amostras consecutivas do sinal de entrada original. Com isso, o tamanho de memória do filtro esparsos, representado por  $N_i$ , fica reduzido a

$$N_i = \left\lfloor \frac{N-1}{L} \right\rfloor + 1 \quad (2.33)$$

onde  $\lfloor \cdot \rfloor$  denota operação de truncamento, na TAB. 2.4, se encontram as expressões desta abordagem.

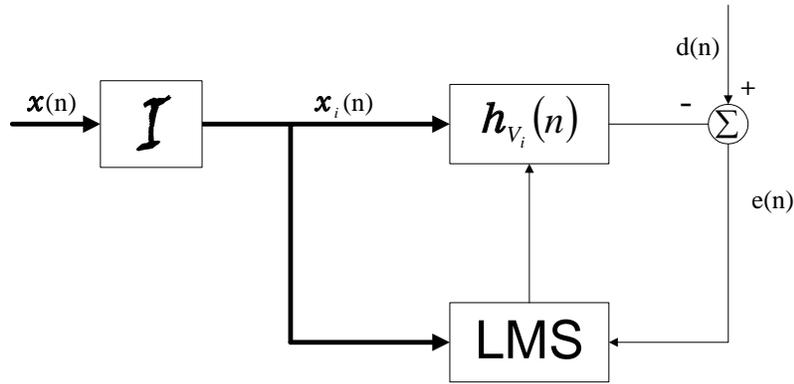


FIG. 2.4: Filtro Volterra Adaptativo Interpolado.

## 2.7 FILTRO DE VOLTERRA LMS PARCIALMENTE INTERPOLADO

Outra versão simplificada do filtro Volterra adaptativo é obtida através do uso da interpolação apenas nos blocos de maior ordem. Fazendo-se assim, é possível obter uma relação de complexidade bem próxima à do filtro interpolado, uma vez que esses blocos são os que possuem o maior número de coeficientes. A FIG. 2.5 mostra a estrutura de um filtro Volterra adaptativo parcialmente interpolado de terceira ordem.

Nesta figura,  $\mathbf{h}_1$ ,  $\mathbf{h}_2$  e  $\mathbf{h}_{3_i}$  representam os blocos de primeira ordem, segunda ordem e terceira ordem, respectivamente, com  $y_1(n)$ ,  $y_2(n)$  e  $y_3(n)$  representando seus sinais de saída. Os demais sinais da FIG. 2.5 são equivalentes aos da FIG. 2.4.

O número total de coeficientes para este filtro é determinado somando o número de coeficientes para cada um dos blocos, que é obtido através da EQ. 2.27. O tamanho da memória dos blocos interpolados é reduzido de acordo com EQ. 2.33. As expressões deste algoritmo são apresentadas na TAB. 2.5.

O uso de uma abordagem parcialmente interpolada sobre uma versão interpolada é que permite uma redução considerável de coeficientes chegando a uma solução sub-ótima similar as obtidas com um filtro interpolador.

## 2.8 WAVELETS

Nas últimas décadas, a transformada *wavelet* (DAUBECHIES, 1992) tem sido objeto de investigação de muitos pesquisadores que atuam em diversas áreas do conhecimento hu-

TAB. 2.4: O algoritmo Volterra LMS Interpolado

<b>Volterra LMS Interpolado</b>	
Inicializar:	
$\mathbf{x}(0) = \mathbf{h}(0) = \mathbf{0}$	
escolha $\mu_j$ no intervalo $0 < \mu_j < \frac{1}{\lambda_{\max}}$ para $j = 1, 2, \dots, i$	
for $n = 1, 2, \dots$	
{	
$\mathbf{x}_i(n) = \mathcal{I}\mathbf{x}(n)$	
$e(n) = d(n) - \mathbf{h}_i^T(n)\mathbf{x}_i(n)$	
$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) +$	$\begin{bmatrix} \mu_1 \mathbf{I}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_2 \mathbf{I}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_i \mathbf{I}_i \end{bmatrix} e(n)\mathbf{x}_i(n)$
}	

mano. 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-freqüência 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 freqüência, de acordo com o princípio da incerteza de Heiseberg), assim como estão incluídas outras que permitem obter excelente resolução no domínio da freqüência (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 freqüências. Tal característica não é observada em 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  $f(t)$  pertence ao espaço de Hilbert pode ser represen-

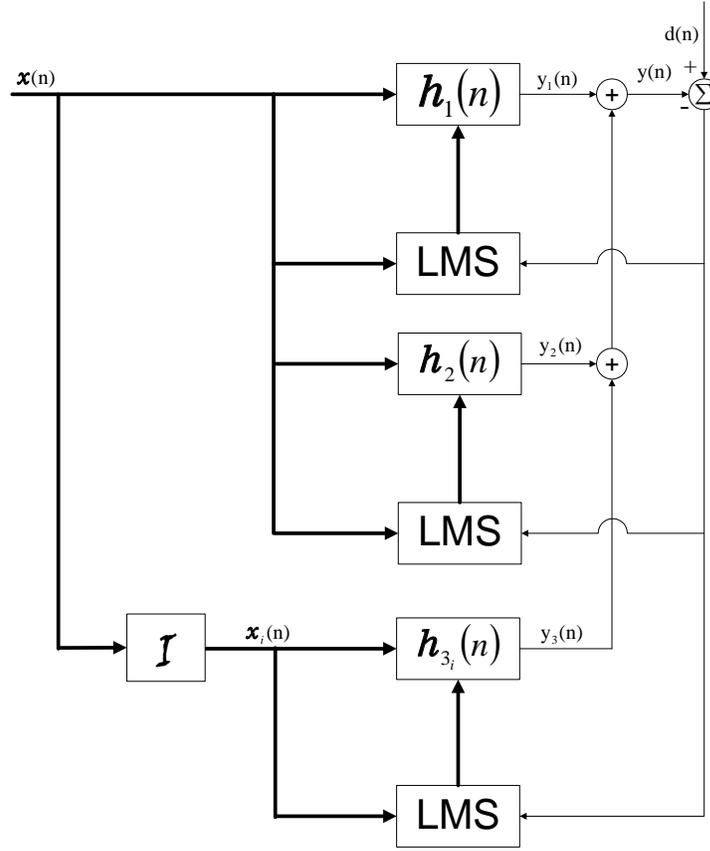


FIG. 2.5: Filtro Volterra Adaptativo Parcialmente Interpolado (somente no bloco de 3ª ordem)

tado da seguinte maneira:

$$f(t) = \sum_j \sum_k a_{j,k} \psi_{j,k}(t), \quad (2.34)$$

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

$$a_{j,k} = \int f(t) \psi_{j,k}(t) dt. \quad (2.35)$$

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 (extritamente limitada no tempo), denominada *wavelet* mãe, usando as operações de deslocamento e escalonamento (BURRUS, 1998):

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in \mathbf{Z}. \quad (2.36)$$

TAB. 2.5: O algoritmo Volterra LMS Parcialmente Interpolado

<b>Volterra LMS Parcialmente Interpolado</b>	
Inicializar:	
$\mathbf{x}(0) = \mathbf{h}(0) = \mathbf{0}$	
escolha $\mu_j$ no intervalo $0 < \mu_j < \frac{1}{\lambda_{\max}}$ para $j = 1, 2, \dots, i_p$	
for $n = 1, 2, \dots$	
{	
$\mathbf{x}(n) = [\mathbf{x}(n) \ \mathbf{x}^2(n) \ \dots \ \mathbf{x}^p(n)]^T$	
$\mathbf{x}_{i_p}(n) = [\mathbf{x}(n) \ \mathbf{x}^2(n) \ \dots \ \mathcal{I}_p \mathbf{x}^p(n)]^T$	
$e(n) = d(n) - \mathbf{h}_{i_p}^T(n) \mathbf{x}_{i_p}(n)$	
$\mathbf{h}_{i_p}(n+1) = \mathbf{h}_{i_p}(n) +$	$\begin{bmatrix} \mu_1 \mathbf{I}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mu_2 \mathbf{I}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mu_{i_p} \mathbf{I}_{i_p} \end{bmatrix} e(n) \mathbf{x}_{i_p}(n)$
}	

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, EQ. 2.34, quanto o processo de análise, EQ. 2.35, são realizados de forma eficiente através do algoritmo piramidal, o qual emprega uma sucessão de filtros passa-baixa e passa-alta acompanhados de operação de decimação no processo de análise, e acompanhados 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* (BURRUS, 1998),(MALLAT, 1998), 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,  $\mathbf{a}$ . Nessa 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  $j = 1$ . Esse 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 permitida 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. Enquanto que, em geral, nos níveis de maior resolução ficam concentradas energias associadas aos sinais espúrios subjacentes ao sinal de interesse. Essa é uma das mais importantes propriedades dessa transformada que será explorada nesta dissertação.

## 2.9 MATRIZ DE TRANSFORMAÇÃO $\mathbf{T}$

Como comentado na seção prededente, as *wavelets* podem ser interpretadas como um banco de filtros; deste ponto de vista e usando a equação de escalonamento e a propriedade de ortogonalidade entre os subespaços, obtém-se:

$$c_j = \sum_n c_{j+1}(n)h_0(n - 2k) \quad \text{e} \quad d_j = \sum_n c_{j+1}(n)h_1(n - 2k), \quad (2.37)$$

expressões que são implementados pelo algoritmo piramidal.

Conforme apresenta a EQ. 2.37, o algoritmo piramidal (VIDAKOVIC, 1999) expressa os coeficientes *wavelets* de um determinado nível em termos dos coeficientes do nível adjacente. Uma forma elegante de representar essa operação é mediante uso do operador de decimação  $[2 \downarrow]$ . Este operador retém apenas amostras pares de uma seqüência, como

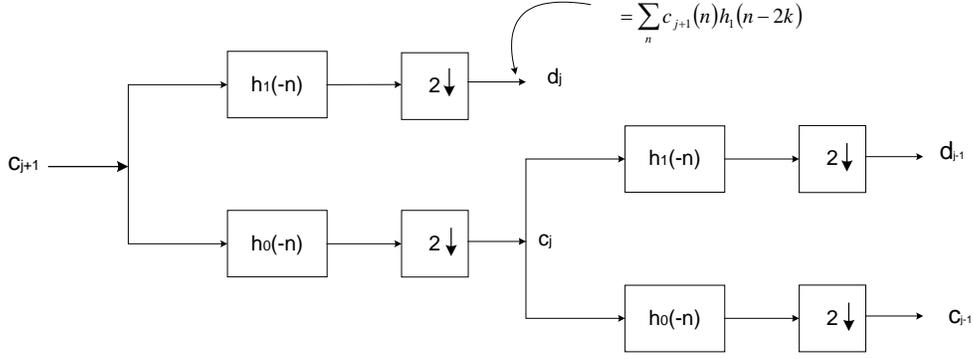


FIG. 2.6: Dois Estágios do Algoritmo Piramidal

indicado na FIG. 2.6. Vale ressaltar que, na operação de decimação, claramente existe a possibilidade de perda de informação pois, com o descarte dos dados, pode surgir o efeito de *aliasing*.

A seguir, apresentamos a representação matricial para uma dada transformação *wavelet*, caracterizada pelos filtros  $\mathbf{h}_0$  e  $\mathbf{h}_1$ , supondo que  $N_1 = 2^J$ .

Define-se um conjunto de matrizes  $\{\mathbf{H}_k^0\}$  indexadas por  $k$ , de dimensão  $(2^{(J-k)} \times 2^{(J-k+1)})$  cujos elementos são obtidos a partir do filtro de escalonamento, sendo o elemento genérico dessas matrizes  $\mathbf{H}_k^0(i, j)$  dado por

$$h_0(l), l = (N - 1) + (j - 1) - 2(i - 1) \text{ módulo } 2^{J-k+1}. \quad (2.38)$$

O conjunto de matrizes  $\{\mathbf{H}_k^1\}$  é definido de forma similar, exceto pelo fato de seus elementos serem obtidos a partir dos coeficientes do filtro *wavelet*.

Note-se que as matrizes definidas acima são circulares: suas  $i$ -ésimas linhas são iguais às respectivas primeiras linhas com deslocamento circular de  $2(i - 1)$  elementos.

A constante  $N$  na EQ. 2.38 afeta a posição das entradas nas sub-matrizes  $\mathbf{H}_k^0$  e  $\mathbf{H}_k^1$ . Para a família das *wavelets* de Daubechies, uma escolha apropriada desse parâmetro é dado pela quantidade de momentos nulos.

Em (VIDAKOVIC, 1999) é mostrado que as matrizes *wavelets* podem ser obtidas a partir das matrizes  $\mathbf{H}_k^0$  e  $\mathbf{H}_k^1$  da seguinte maneira

$$\mathbf{T}_1 = \begin{bmatrix} \mathbf{H}_1^1 \\ \mathbf{H}_1^0 \end{bmatrix}, \quad (2.39)$$

$$\mathbf{T}_2 = \begin{bmatrix} \begin{bmatrix} \mathbf{H}_2^1 \\ \mathbf{H}_2^0 \end{bmatrix} \cdot \mathbf{H}_1^1 \\ \mathbf{H}_1^0 \end{bmatrix}, \quad (2.40)$$

$$\mathbf{T}_3 = \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} \mathbf{H}_3^1 \\ \mathbf{H}_3^0 \end{bmatrix} \cdot \mathbf{H}_2^1 \\ \mathbf{H}_2^0 \end{bmatrix} \cdot \mathbf{H}_1^1 \\ \mathbf{H}_1^0 \end{bmatrix} \quad (2.41)$$

e assim sucessivamente, até se obter  $\mathbf{T}_J$ . O produto  $\boldsymbol{\alpha} = \mathbf{T}_J \mathbf{y}$  fornece a transformada *wavelet* desejada.

Na representação,  $\mathbf{T}_J$  é aplicada a um conjunto finito de dados o que corresponde a um truncamento da transformada *wavelet* em relação ao caso em que é aplicada a um sinal de duração infinita. Os coeficientes obtidos na primeira situação podem diferir daqueles obtidos na segunda.

Há várias abordagens para tentar contornar esse problema; porém, as mais usadas são a regra periódica e a da reflexão. Na regra periódica, um sinal com  $N_1$  amostras é transformado em um sinal de duração infinita e periódico, com período  $N_1$ . Este procedimento preserva a ortogonalidade, além de ser rápido e numericamente eficiente. Na regra de reflexão, os dados originais são refletidos nas fronteiras e posteriormente estendidos de forma periódica. Existem algoritmos disponíveis para implementar este procedimento (DAUBECHIES, 1992)

## 2.10 RESUMO DO CAPÍTULO

Neste capítulo foram apresentados de maneira resumida, os conceitos básicos relacionados a importantes assuntos que são abordados ou empregados neste trabalho. Neste contexto, foram apresentados breves revisões de filtragem adaptativa empregando os algoritmos da família LMS no domínio tempo.

Fez-se ainda uma pequena introdução às séries de Volterra e ao algoritmo LMS neste contexto. Foram discutidos alguns algoritmos da família LMS empregados em séries de Volterra que utilizam técnicas de redução de ordem do vetor de pesos, visando a redução do número de coeficientes do filtro e portanto a diminuição da complexidade computacional. Por fim, apresentou-se uma breve introdução à transformada *wavelet*.

#### 3.1 INTRODUÇÃO

A teoria dos filtros adaptativos vem sendo amplamente empregada; em particular, o algoritmo LMS tem sido muito usado pela sua simplicidade e robustez. Com o intuito de tentar melhorar a convergência deste algoritmo, lenta para sinais de entrada com elevada correlação entre suas amostras, neste capítulo estuda-se os algoritmos LMS no domínio da transformada (NARAYAN, 1983), particularmente a transformada *wavelets*. Cabe lembrar que nossa aplicação de interesse é um sistema não-linear e que sempre estaremos buscando os melhores resultados em termos de velocidade de convergência e redução do custo computacional.

A estrutura básica de como é feito o mapeamento do domínio do tempo para o domínio da transformada é apresentada na FIG. 3.1. No decorrer do capítulo, serão detalhadas todas as características deste tipo de filtro e, em particular, de nossa proposta.

#### 3.2 O ALGORITMO WTD-LMS

O algoritmo LMS apresenta a vantagem de baixa complexidade computacional em relação a outros algoritmos de filtragem adaptativa; porém, ele pode apresentar muito baixa velocidade de convergência quando o sinal em sua entrada for fortemente correlacionado (GALDINO, 2003).

Desta forma, muitos algoritmos baseados no LMS convencional vêm sendo propostos para suplantarem as limitações de convergência do LMS convencional sem, no entanto, aumentar sobremaneira a complexidade computacional.

Um procedimento proposto para melhorar o desempenho do algoritmo LMS convencional é a normalização do sinal de entrada do filtro adaptativo. O algoritmo assim obtido é o conhecido LMS normalizado, denominado NLMS, dado por (DINIZ, 2002)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)}e(n)\mathbf{x}(n). \quad (3.1)$$

Outra abordagem que pode ser adotada para melhorar as características de con-

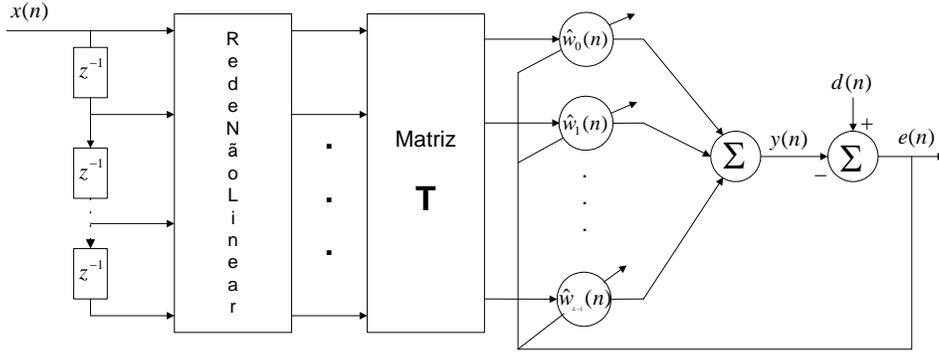


FIG. 3.1: Filtro adaptativo no domínio da transformada.

vergência do algoritmo diante de sinais de entrada que apresentam forte correlação é o LMS no domínio da transformada. De acordo com esse procedimento, o sinal de entrada do algoritmo LMS,  $\mathbf{s}(n)$ , é dado por

$$\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n), \quad (3.2)$$

onde  $\mathbf{T}$  é uma matriz que representa o operador linear que realiza o mapeamento entre o domínio do tempo e o domínio da transformada (TEWFIK, 1994).

O algoritmo LMS no domínio da transformada *wavelet*, denominado *Wavelet Transform Domain LMS* (WTD-LMS), também sofre um tipo de normalização dado por

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e(n) \boldsymbol{\sigma}(n) \mathbf{s}(n), \quad (3.3)$$

na qual  $\boldsymbol{\sigma}(n)$  é uma matriz diagonal cujo  $i$ -ésimo elemento é dado por  $[\gamma + \sigma_i^2(n)]^{-1}$ , sendo  $\sigma_i^2(n)$  uma estimativa da variância de  $s_i(n)$ , dado por

$$\sigma_i^2(n) = \alpha s_i^2(n) + (1 - \alpha) \sigma_i^2(n-1), \quad (3.4)$$

onde  $\alpha$  é um fator pequeno,  $0 < \alpha \leq 0.1$ , e  $\gamma$  uma constante muito pequena, empregada somente para evitar possíveis divisões por zero.

Vale lembrar que, no domínio do tempo (algoritmo NLMS), todos elementos do vetor sinal de entrada tem a mesma variância e por isso possui uma normalização igual;  $\boldsymbol{\sigma}^2$  seria, neste caso,  $(\mathbf{x}^T(n)\mathbf{x}(n))^{-1}\mathbf{I}$ . Contudo, ao empregar o algoritmo LMS em aplicações que envolvem filtros de Volterra, além de termos uma normalização diferente para cada elemento, também é importante adotar passos distintos para cada ordem do polinômio

de Volterra. Essa medida é fundamental para melhorar a velocidade de convergência do algoritmo LMS (DINIZ, 2002). 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:

$$\boldsymbol{\mu}_p = \begin{bmatrix} \mu_1 \mathbf{I}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_2 \mathbf{I}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_p \mathbf{I}_p \end{bmatrix}. \quad (3.5)$$

onde  $\mathbf{I}_j$  é uma matriz identidade de dimensão correspondente ao número de coeficientes de  $j$ -ésima ordem.

Portanto, para o caso do algoritmo NLMS usado num núcleo de Volterra (KALLURI, 1999), sua equação de atualização pode ser dada pela seguinte expressão

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \boldsymbol{\mu}_p \mathbf{x}(n). \quad (3.6)$$

No domínio da transformada *wavelet*, entretanto, não há razão para empregar os passos da maneira como apresentada acima. Isto ocorre porque cada coeficiente, neste domínio, pode carregar informação de várias ordens do polinômio de Volterra devido a memória induzida pela matriz de análise ( $\mathbf{T}$ ). Pode-se, no entanto, adotar um passo distinto para cada nível de resolução da DWT, como indicado na expressão abaixo.

$$\widehat{\mathbf{w}}(n+1) = \widehat{\mathbf{w}}(n) + \boldsymbol{\mu}_M e(n) \boldsymbol{\sigma}(n) \mathbf{s}(n). \quad (3.7)$$

Neste caso,  $\boldsymbol{\mu}_M$  é uma matriz diagonal, como a apresentada na EQ. 3.5, com cada elemento  $\mu_j$  correspondendo a cada um dos  $M$  níveis de decomposição *wavelet*. Levando em consideração os critérios apresentados nesta seção, pode-se fazer uma simulação para determinar a validade das formulações. Para isto, empregamos o filtro adaptativo de Volterra na identificação de um sistema dado por:

$$\begin{aligned} d(n) = & -0.78x(n) - 1.48x(n-1) + 1.39x(n-2) + 0.54x(n-3) + 3.72x^2(n) \\ & + 1.86x(n)x(n-1) - 1.62x(n)x(n-2) + 0.76x(n)x(n-3) + 1.41x^2(n-1) \\ & + 0.04x(n-1)x(n-2) - 0.13x(n-1)x(n-3) - 0.23x^2(n-2) \\ & - 0.12x(n-2)x(n-3) - 1.52x^2(n-3) + r(n) \end{aligned} \quad (3.8)$$

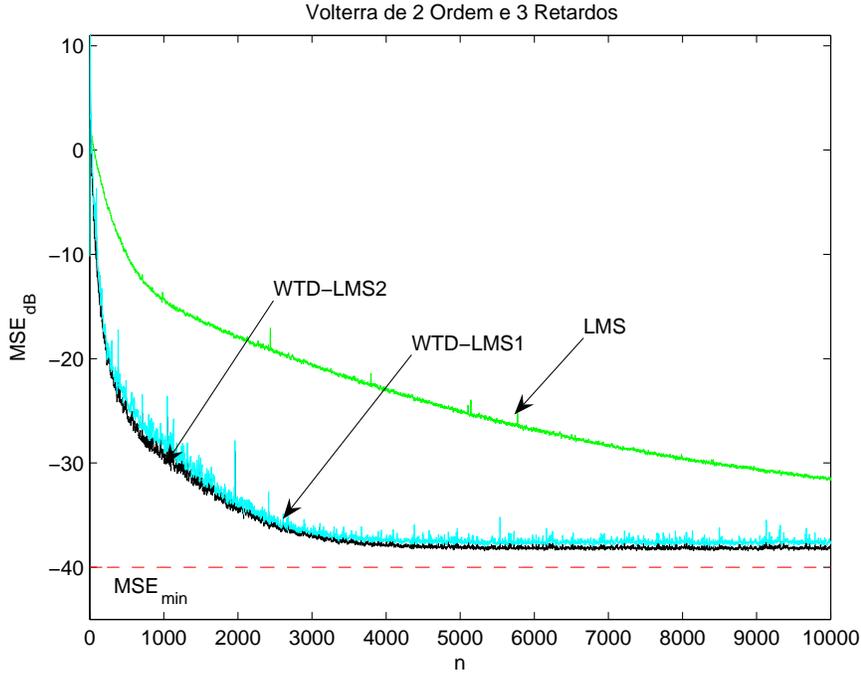


FIG. 3.2: MSE dos Algoritmos WTD-LMSM e LMS.

Observa-se, da equação anterior, que trata-se de um sistema de segunda ordem com três retardos, ou seja, com um total de quatorze coeficientes em seu núcleo de Volterra. Além disso somamos ruído de observação, aqui denotado por  $r(n)$ , com média zero e variância  $\sigma_r^2 = 1e^{-3}$ . Devido ao baixo número de coeficientes e levando em consideração a ordem do polinômio de Volterra, igual a dois, fazemos a decomposição em dois níveis e empregamos a *wavelet* de Haar (DAUBECHIES, 1992) para gerar a matriz  $\mathbf{T}$ .

O algoritmo WTD-LMS empregando  $M$  passos será denominado de WTD-LMSM. O desempenho deste algoritmo será avaliado para  $M = 1$  e  $M = 2$ . Para o primeiro caso, usou-se  $\mu = 0.04$ ; no segundo caso, dois passos foram empregados, um para cada nível de decomposição:  $\mu_0 = 0.025$  e  $\mu_1 = 0.035$ .

Os desempenho do algoritmo LMS Volterra convencional<sup>1</sup>, também é avaliado na simulação, neste caso foram considerados  $\mu_1 = 0.035$  e  $\mu_2 = 0.025$ . Os resultados dos algoritmos investigados, em termos do Erro Médio Quadrático (ou MSE do inglês *Mean Squared Error*), podem ser observados na FIG. 3.2:

---

<sup>1</sup> $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_p e(n)\mathbf{x}(n)$

Como se observa os, algoritmos WTD-LMS apresentam melhor desempenho do que o LMS convencional. Com relação aos algoritmos no domínio da transformada *wavelet*, verifica-se uma pequena melhoria para o que emprega dois passos. Porém este algoritmo, o WTD-LMS2 possui a dificuldade de ter que se escolher dois valores de passo, procedimento que a princípio deve ser realizado manualmente ou por simulação.

Empregamos os mesmos algoritmos na identificação de um sistema de terceira ordem com dois retardos (CHANG, 2003) dado por:

$$\begin{aligned}
d(n) = & -0.78x(n) - 1.48x(n-1) + 1.39x(n-2) + 0.54x^2(n) \\
& + 3.72x(n)x(n-1) + 1.86x(n-1)x(n-2) - 1.62x^2(n-1) \\
& + 0.76x(n-1)x(n-2) + 1.41x^2(n-2) + 0.04x^3(n) \\
& - 0.13x(n)x^2(n-1) - 0.12x(n)x^2(n-2) - 0.23x^2(n)x(n-1) \\
& - 1.52x^2(n)x(n-2) - 0.76x^3(n) - 0.75x(n-1)x^2(n-2) \\
& + 0.15x^2(n-1)x(n-2) + 0.5x^3(n-2) + 0.33x(n)x(n-1)x(n-2) + r(n)
\end{aligned} \tag{3.9}$$

Da EQ. 3.9, verifica-se que o número de coeficientes do filtro de Volterra é 19 (TAN, 2001). O sinal de entrada  $\mathbf{x}(n)$  passa pelo canal dado por:  $0.9045 + z^{-1} + 0.9045z^{-2}$  e a relação sinal-ruído foi feita igual a  $-30$  dB.

Os passos adaptativos para os algoritmos dos filtros de Volterra LMS são  $\mu_1 = 0.001$ ,  $\mu_2 = 0.003$  e  $\mu_3 = 0.005$  e para o NLMS (EQ. 3.1) são  $\mu_1 = 0.003$ ,  $\mu_2 = 0.006$  e  $\mu_3 = 0.01$ . Esses valores de passo foram obtidos experimentalmente visando 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-LMS3 foram empregadas as funções de base *wavelet* conhecidas como Daubechies, cujas funções *wavelets* possuem dois momentos nulos (Daubechies 2) (DAUBECHIES, 1992), e três níveis de resolução, sendo que se adota-se um valor distinto, para cada nível de resolução. 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  $\mathbf{x}(n)$ , e provavelmente um menor valor de SNR.

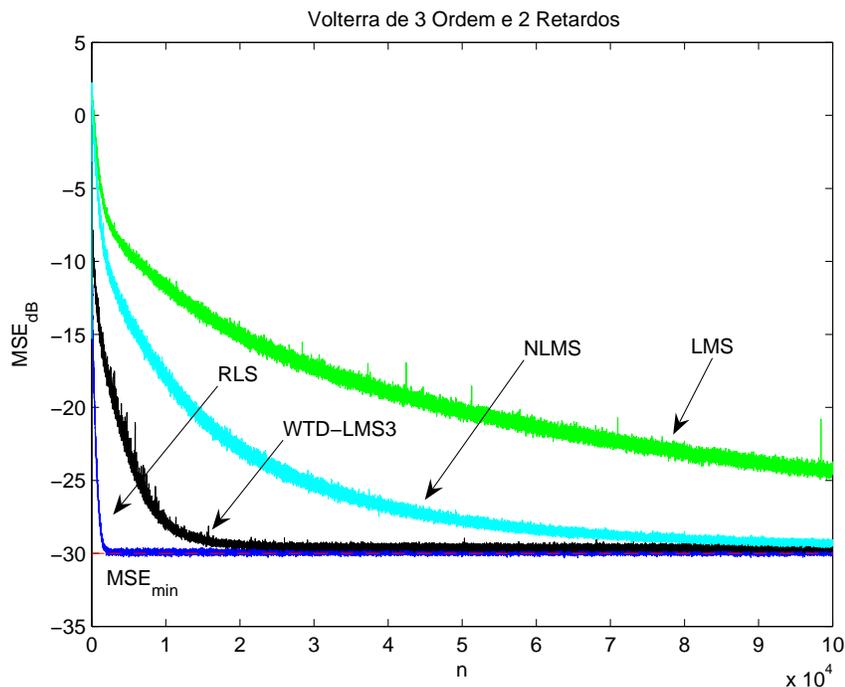


FIG. 3.3: MSE dos Algoritmos WTD-LMS3, LMS, NLMS e RLS.

Os resultados obtidos, em termos de erro médio quadrático obtidos pelos algoritmos LMS, NLMS, WTD-LMS3 e RLS (usado somente como comparação) são mostrados na FIG. 3.3 onde foi considerada uma relação sinal ruído de 30 dB.

Como pode-se observar na FIG. 3.3, 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. Dentre os algoritmos baseados no LMS, o algoritmo WTD-LMS3 é o que apresenta as melhores características de desempenho, com a maior velocidade de convergência para o valor de MSE mínimo (-30 dB). O desempenho deste algoritmo é superado apenas pelo algoritmo RLS (DINIZ, 2002). Porém, vale mencionar que o algoritmo RLS apresenta uma complexidade computacional bem superior aos demais algoritmos avaliados (baseados no LMS), além de não possuir uma garantida estabilidade numérica (DINIZ, 2002).

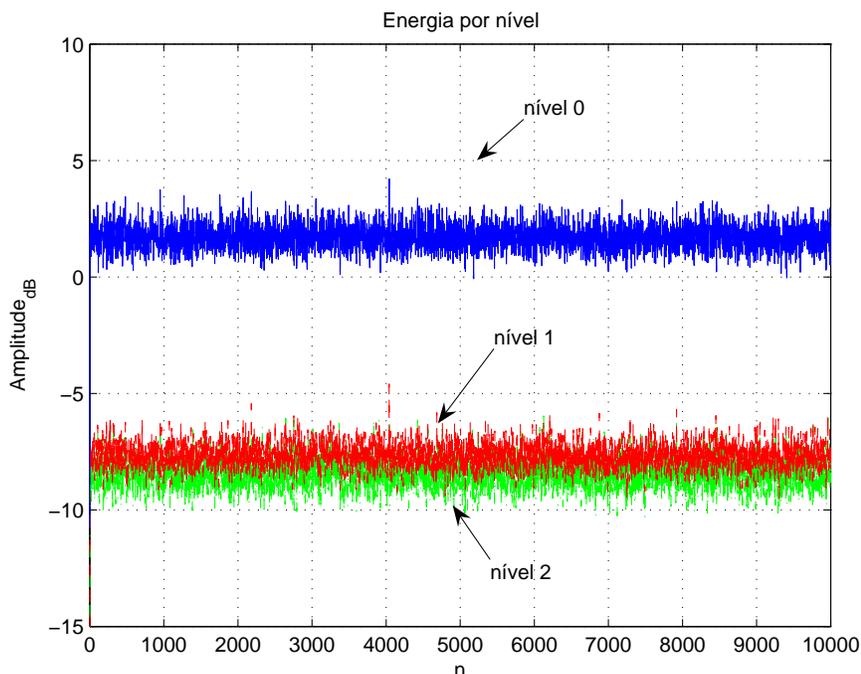


FIG. 3.4: Energia por Nível.

### 3.3 UMA NOVA PROPOSTA PARA O ALGORITMO WTD-LMS

A transformada *wavelet* tem a propriedade de concentrar maior quantidade de energia nos níveis de decomposição de menor resolução. Isto pode ser verificado mediante um simples exemplo: empregamos um sistema de Volterra de terceira ordem e dois retardos, e fazemos uma decomposição de três níveis. O resultado é apresentado na FIG. 3.4, na qual mostra-se uma clara concentração de energia no nível 0, em relação às energias do sinal decomposta nos níveis 1 e 2.

Até o momento, estamos somente usando a transformação proposta em (TEWFIK, 1994) num filtro de Volterra e com uma certa dificuldade em otimizar  $M$  valores de passos. 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

também maiores (GALDINO, 2004).

Assim sendo, a criteriosa escolha do passo do algoritmo LMS é importante para ter-se um bom compromisso entre velocidade de convergência e nível de MSE na condição de regime permanente.

Cabe mencionar que, para uma mesma aplicação, a escolha apropriada do passo do algoritmo LMS depende da Relação Sinal-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 (GALDINO, 2004).

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-se 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 bloco de dados do instante  $n$ , respectivamente.

Tendo em vista que o ajuste fino dos elementos  $\mu_j$  em EQ. 3.7 é crítico, nossa proposta é facilitar a utilização deste algoritmo ao deixar o passo de cada nível de decomposição proporcional à sua energia da maneira como segue.

$$\mu_j(n) = \frac{E_j(n)}{\sum_{i=0}^{M-1} E_i(n)} \mu \quad (3.10)$$

sendo  $M$  a quantidade de níveis de resolução e  $\mu$  o único parâmetro que deverá ser ajustado para controlar e garantir a convergência do algoritmo. Vale mencionar que tanto o denominador quanto o numerador de EQ. 3.10 podem ser obtidos sem considerável complexidade computacional, uma vez que esses parâmetros são também empregados na normalização.

Os passos definidos em EQ. 3.10 podem ser apresentados em forma vetorial da seguinte maneira

$$\boldsymbol{\mu}_M = \begin{bmatrix} \mu_0 \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_1 \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_{M-1} \mathbf{I}_{M-1} \end{bmatrix} \quad (3.11)$$

na qual assumiu-se  $M$  níveis de resolução.

É importante destacar três propriedades marcantes 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 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 na EQ. 3.10, o que reduz bastante a dificuldade de ajuste em relação a um algoritmo genérico WDT-LMSM que emprega M diferentes passos para os diferentes níveis de resolução. Substituindo-se a EQ. 3.10 na EQ. 3.11 tem-se que:

$$\boldsymbol{\mu}_j = \begin{bmatrix} \mu \frac{E_0(n)}{\sum_{i=0}^{M-1} E_i(n)} \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu \frac{E_1(n)}{\sum_{i=0}^{M-1} E_i(n)} \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu \frac{E_j(n)}{\sum_{i=0}^{M-1} E_i(n)} \mathbf{I}_{M-1} \end{bmatrix}. \quad (3.12)$$

Na TAB. 3.1, é apresentado de forma compacta o algoritmo WTD-LMS proposto (BERNAL O., 2005).

Empregando as equações da TAB. 3.1 e o sistema da EQ. 3.9, pode-se refazer o experimento anterior; os resultados obtidos são mostrados na FIG. 3.5

Os resultados apresentados na FIG. 3.5 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-LMSM possuem similares complexidades computacionais.

### 3.4 REDUÇÃO DO FILTRO DE VOLTERRA

Como mencionado anteriormente, a transformada *Wavelet* possui a propriedade de concentração de energia nos baixos níveis de resolução. Assim sendo, pode-se obter um

TAB. 3.1: O Algoritmo *Wavelet Transform-Domain* LMS.

WTD-LMS
<p>Inicialização</p> $\mathbf{x}(0) = \hat{\mathbf{w}}(0) = [0 \dots 0]^T$ <p><math>\gamma =</math> constante pequena</p> $0 < \alpha \leq 0.1$ <p>escolha <math>\mu</math></p> <p>Para cada <math>n</math></p> <p>{</p> $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$ $e(n) = d(n) - \mathbf{s}^T(n)\hat{\mathbf{w}}(n)$ $\sigma_i^2(n) = \alpha s_i^2(n) + (1 - \alpha)\sigma_i^2(n - 1)$ $\boldsymbol{\sigma}(n) = \text{diag} \left( [\gamma + \sigma_i^2(n)]^{-1} \right)$ $\mu_j(n) = \mu \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sum_{i=0}^{M-1} \mathbf{s}^T(n)\mathbf{s}(n)}$ $\boldsymbol{\mu}_j(n) = \text{diag} (\mu_j(n))$ $\hat{\mathbf{w}}(n + 1) = \hat{\mathbf{w}}(n) + \boldsymbol{\mu}_j(n)e(n)\boldsymbol{\sigma}(n)\mathbf{s}(n)$ <p>}</p>

algoritmo WTD-LMS com uma quantidade de coeficientes menor, na qual 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 parte 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 podem exercer forte influência no desempenho desse novo algoritmo.

Será, a seguir, avaliado o desempenho do RO WTD-LMS; neste experimento, empregou-se a transformada de Haar (DAUBECHIES, 1992) 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 de-

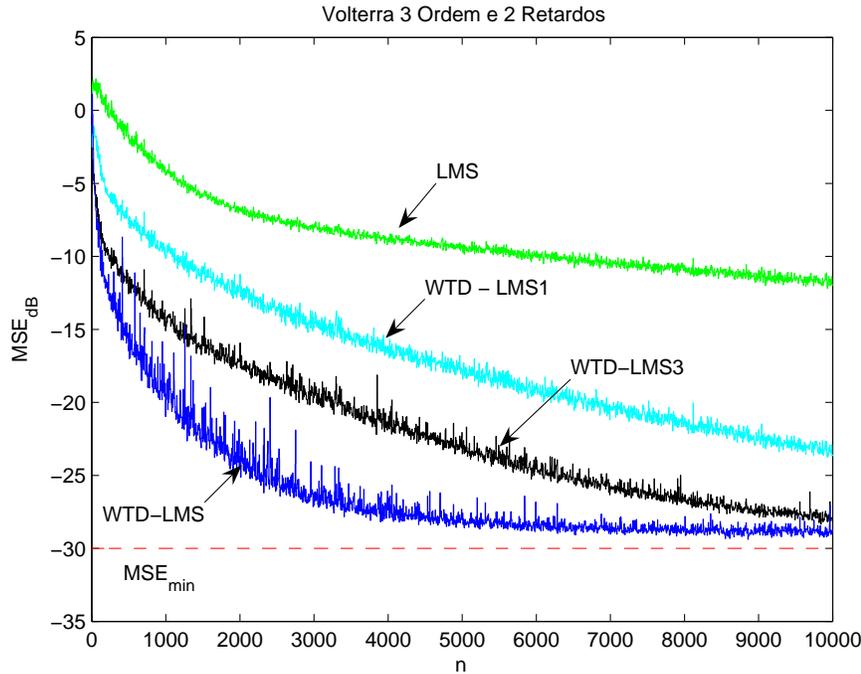


FIG. 3.5: MSE dos Algoritmos WTD-LMS.

nominado VPI-LMS proposto em (BATISTA, 2004). 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 FIG. 3.6.

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.

### 3.5 COMPLEXIDADE COMPUTACIONAL

Uma das causas de sucesso de um determinado algoritmo é, em grande parte, a sua eficiência computacional; dito de outra maneira, a quantidade de operações que devem ser realizadas pelo algoritmo em cada iteração. A grande vantagem dos algoritmos LMS, como já mencionado é a sua baixa complexidade computacional; por outro lado, em algumas aplicações ele é fortemente penalizado devido à sua lenta velocidade de convergência.

Para o caso do algoritmo WTD-LMS temos as seguintes considerações: seja a matriz

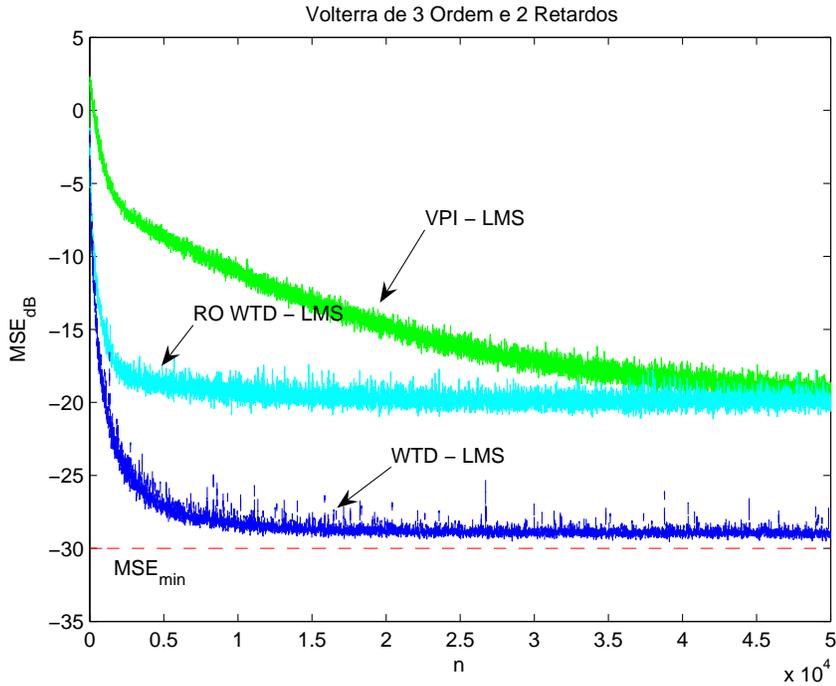


FIG. 3.6: MSE dos Algoritmos RO WTD-LMS.

de transformação  $\mathbf{T}$  de dimensão  $L \times L$  que vai ser multiplicado por um vetor  $\mathbf{f}$  de dimensão  $L \times 1$ , o custo computacional desta multiplicação é de  $O(L)$  operações (HOSUR, 1997; HERLEY, 1994). Levando isto em consideração, podemos definir o número de multiplicações, divisões e raízes quadradas. Lembrando que  $L$  representa o tamanho do filtro,  $M$  a quantidade de níveis de resolução da transformada e  $p$  o grau do polinômio, o número dessas operações por iteração, para cada algoritmo, se encontram na TAB. 3.2.

Desta tabela, observa-se que o algoritmo RLS tem a maior complexidade e o LMS é o que apresenta a menor complexidade computacional. O algoritmo proposto, WTD-LMS apresenta complexidade computacional bem menor que a do algoritmo RLS e apenas um pouco maior o que a do algoritmo LMS.

Da análise conjunta dos resultados de MSE apresentados na seção anterior e de complexidade computacional apresentados na TAB. 3.2, pode-se concluir que o algoritmo WTD-LMS apresenta um bom compromisso entre desempenho e complexidade, se comparado aos demais investigados.

TAB. 3.2: Complexidade computacional dos algoritmos investigados.

ALGORITMO	MULTIP.	DIVIS.
LMS	$3L + M$	0
WTD-LMSM	$4L + M$	$L$
WTD-LMS	$4L + M$	$L + M$
NLMS	$4L + M$	1
RLS	$3L^2 + L + 2$	$L^2$

### 3.6 RESUMO

Neste capítulo, foi apresentado uma nova versão do algoritmo WTD-LMS aplicado satisfatoriamente em filtragem de Volterra. Quanto à velocidade de convergência, verificou-se ser mais veloz que o NLMS e que as versões LMS convencionais. Também, pode-se observar que possui uma complexidade computacional relativamente baixa.

Foi também apresentada uma versão do algoritmo WTD-LMS que reduz o número de coeficientes do *kernel* de Volterra, visando uma redução no custo computacional. Esta versão com ordem reduzida mostrou-se eficiente quando comparada com a proposta de filtros parcialmente interpolados, reduz o mesmo número de coeficientes.

## 4 ANÁLISES DOS ALGORITMO WTD-LMS

### 4.1 INTRODUÇÃO

Neste capítulo foi desenvolvido a análise do erro na identificação dos coeficientes do filtro adaptativo WTD-LMS quando empregado para sinais de entrada modelado por série de Volterra. A análise aqui realizada contempla a média e a covariância dos erros nos coeficientes do filtro. Devido à complexidade do problema, a análise é realizada admitindo-se uma série de suposições simplificadoras brevemente discutidas neste capítulo; em particular, a análise na covariância segue o procedimento proposto em (GALDINO, 2004) que foi desenvolvido para a análise do algoritmo LMS convencional.

Como fruto desta análise são obtidos valores máximos de  $\mu$  ( $\mu_{\max}$ ) a partir da expressão do erro médio quadrático nos coeficientes. Para realizar as análises, será empregado o modelo mostrado na FIG. 4.1.

### 4.2 CRITÉRIOS PARA A ANÁLISE

A estrutura de um filtro de Volterra, com sua grande quantidade de coeficientes e produtos cruzados das amostras da entrada em diversos instantes de tempo, dificulta a realização de análises devido à elevada complexidade do problema.

Neste trabalho, para realizar uma análise matemática do algoritmo LMS no domínio da transformada *wavelet*, admitimos algumas hipóteses simplificadoras, que são detalhadas a seguir.

- **Suposição I** – Admite-se que a transformada *wavelet* aproxima a transformada de Karhunen-Loeve do vetor  $\mathbf{x}(n)$  (MALLAT, 1998);
- **Suposição II** – As amostras do vetor de entrada transformado,  $\mathbf{s}(n)$ , e do sinal desejado,  $d(n)$ , possuem média nula e são modelados por processos estocásticos Gaussianos estacionários em sentido amplo;
- **Suposição III** – O ruído em tempo discreto,  $r(n)$ , resulta da amostragem de um processo estacionário em sentido amplo que possui média nula e densidade espectral

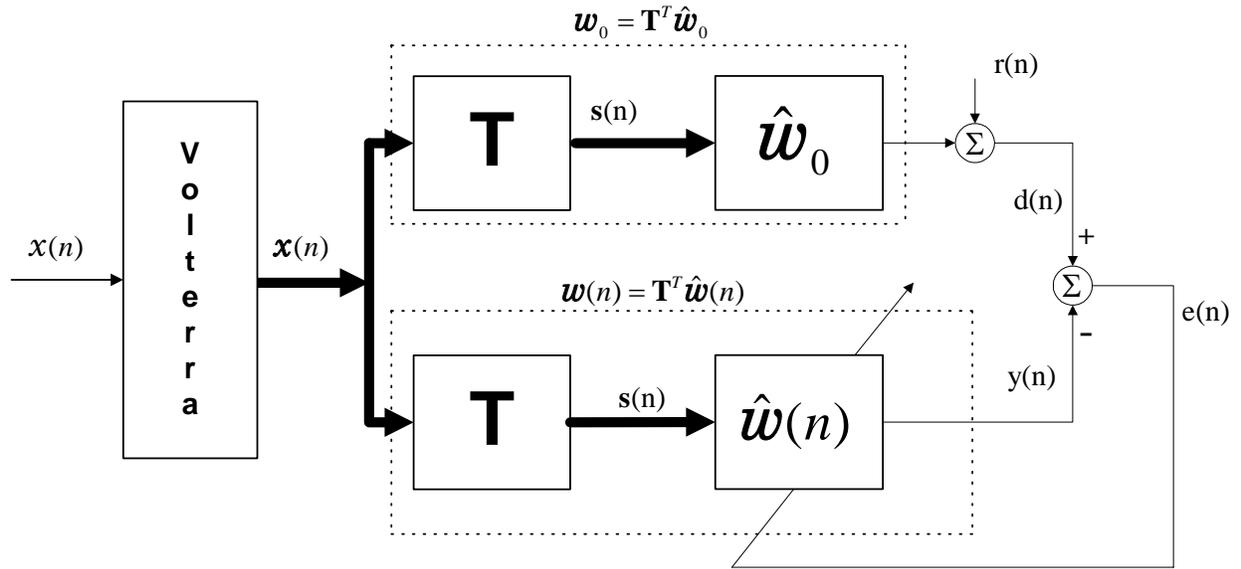


FIG. 4.1: Modelo utilizado na análise.

de potência constante dentro da faixa de frequência de interesse. Além disso, admite-se que este ruído é estatisticamente independente de  $\mathbf{s}(n)$  e de  $\mathbf{w}(n)$ , para qualquer  $n$ ;

- **Suposição IV** – Em regime permanente, o vetor  $\hat{\mathbf{w}}(n)$  é um processo estocástico estatisticamente independente de  $\mathbf{s}(n)$ .
- **Suposição V** – Na estatística do sinal temos que momentos ímpares são nulos, ou seja,  $E[x^l(i)] = 0$  para  $l$  ímpar.

### 4.3 ANÁLISE NA MÉDIA

Nesta seção, considerando as suposições apresentadas na sessão anterior, é analisado o valor esperado do vetor de erro nos coeficientes, definido por

$$\Delta \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}(n) \quad (4.1)$$

onde  $\hat{\mathbf{w}}_0 = \mathbf{T}\mathbf{w}_0$ , a DWT da solução de Wiener.

Como sabemos, o erro a *priori* é dado por:

$$e(n) = d(n) - \mathbf{s}^T(n)\hat{\mathbf{w}}(n) \quad (4.2)$$

e a atualização dos coeficientes para o algoritmo WTD-LMS, como visto no capítulo anterior, é dada por:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \boldsymbol{\mu}_M(n)\boldsymbol{\sigma}(n)\mathbf{s}(n)e(n). \quad (4.3)$$

Considerando-se o filtro no domínio da transformada, o vetor de entrada transformado é dado por:

$$\mathbf{s}(n) = \begin{bmatrix} \mathbf{s}_0(n) \\ \vdots \\ \mathbf{s}_j(n) \\ \vdots \\ \mathbf{s}_{M-1}(n) \end{bmatrix} = \mathbf{T}\mathbf{x}(n), \quad (4.4)$$

onde  $\mathbf{s}_j(n)$  é um vetor com os  $M_j$  elementos do  $j$ -ésimo nível de decomposição da transformada *wavelet*.

Considerando que  $\boldsymbol{\mu}_M$  é uma matriz diagonal que contém os passos e adotando-se um mesmo valor de passo em cada elemento de um mesmo nível da transformada, esta pode ser representada por:

$$\boldsymbol{\mu}_M(n) = \begin{bmatrix} \mu_0(n)\mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mu_1(n)\mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mu_{M-1}(n)\mathbf{I}_{M-1} \end{bmatrix}. \quad (4.5)$$

onde  $\mathbf{I}_j$  é a matriz identidade de dimensão  $M_j \times M_j$  e  $M$  o número de níveis de decomposição da transformada *wavelet*.

Relembramos que a equação proposta para a atualização do passo é dada por:

$$\mu_j(n) = \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\mathbf{s}^T(n)\mathbf{s}(n)}\mu \quad (4.6)$$

onde  $\mu$  controla as propriedades de convergência do algoritmo proposto.

Podemos, ainda, escrever de uma forma simplificada que:

$$\boldsymbol{\sigma}(n) = \begin{bmatrix} \ddots & & & \\ & \frac{1}{\alpha s_i^2(n) + (1-\alpha)\sigma_i^2(n-1)} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix}. \quad (4.7)$$

Para  $n$  grande e sinais estacionários, podemos assumir que  $\sigma_i^2(n) = \alpha s_i^2(n) + (1-\alpha)\sigma_i^2(n-1)$  é uma estimativa não polarizada de  $E[s_i^2(n)]$  e, empregando-se a Suposição II, aproximaremos cada elemento da diagonal pelo inverso da variância ( $1/\sigma_{s_i}^2$ ). Além disto, admitindo-se que as variâncias dos elementos  $s_i(n)$  de um mesmo nível  $j$  de decomposição são todas iguais a  $\sigma_j^2$ , podemos escrever que:

$$\boldsymbol{\sigma}(n) \approx \begin{bmatrix} \frac{1}{\sigma_{s_0}^2} \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sigma_{s_1}^2} \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{1}{\sigma_{s_{M-1}}^2} \mathbf{I}_{M-1} \end{bmatrix}. \quad (4.8)$$

Definimos, então,  $\boldsymbol{\beta}(n)$  como:

$$\boldsymbol{\beta}(n) = \boldsymbol{\mu}_M(n) \boldsymbol{\sigma}(n) \approx \begin{bmatrix} \frac{\mathbf{s}_0^T(n) \mathbf{s}_0(n)}{\sigma_{s_0}^2 \mathbf{s}^T(n) \mathbf{s}(n)} \mu \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{s}_1^T(n) \mathbf{s}_1(n)}{\sigma_{s_1}^2 \mathbf{s}^T(n) \mathbf{s}(n)} \mu \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{\mathbf{s}_{M-1}^T(n) \mathbf{s}_{M-1}(n)}{\sigma_{s_{M-1}}^2 \mathbf{s}^T(n) \mathbf{s}(n)} \mu \mathbf{I}_{M-1} \end{bmatrix} \quad (4.9)$$

e podemos reescrever a EQ. 4.3, da seguinte maneira:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \boldsymbol{\beta}(n) \mathbf{s}(n) [d(n) - \mathbf{s}^T(n) \hat{\mathbf{w}}(n)]. \quad (4.10)$$

Utilizando a EQ. 4.9, a equação anterior pode ser expressada por

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) +$$

$$\frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \frac{\mathbf{s}_0^T(n)\mathbf{s}_0(n)}{\sigma_{s_0}^2} \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{s}_1^T(n)\mathbf{s}_1(n)}{\sigma_{s_1}^2} \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{\mathbf{s}_{M-1}^T(n)\mathbf{s}_{M-1}(n)}{\sigma_{s_{M-1}}^2} \mathbf{I}_{M-1} \end{bmatrix} \mathbf{s}(n) [d(n) - \mathbf{s}^T(n)\hat{\mathbf{w}}(n)] \quad (4.11)$$

A seguir, consideramos o sinal de referência modelado por

$$d(n) = \mathbf{s}^T(n)\hat{\mathbf{w}}_0 + r(n), \quad (4.12)$$

de modo que podemos re-escrever a EQ. 4.1, usando a definição EQ. 4.11, como

$$\Delta\hat{\mathbf{w}}(n+1) = \Delta\hat{\mathbf{w}}(n) - \frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \frac{\mathbf{s}_0^T(n)\mathbf{s}_0(n)}{\sigma_{s_0}^2} \mathbf{I}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{s}_1^T(n)\mathbf{s}_1(n)}{\sigma_{s_1}^2} \mathbf{I}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{\mathbf{s}_{M-1}^T(n)\mathbf{s}_{M-1}(n)}{\sigma_{s_{M-1}}^2} \mathbf{I}_{M-1} \end{bmatrix} \times \mathbf{s}(n) [\mathbf{s}^T(n)\hat{\mathbf{w}}_0 + r(n) - \mathbf{s}^T(n)\hat{\mathbf{w}}(n)]. \quad (4.13)$$

Esta equação pode ser re-escrita como:

$$\Delta\hat{\mathbf{w}}(n+1) = \left[ \mathbf{I} - \frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \end{bmatrix} \mathbf{s}(n)\mathbf{s}^T(n) \right] \Delta\hat{\mathbf{w}}(n) -$$

$$\frac{\mu r(n)}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \mathbf{s}(n). \quad (4.14)$$

Aplicando o valor esperado na equação anterior, ficamos com:

$$E[\Delta\hat{\mathbf{w}}(n+1)] = E \left[ \left\{ \mathbf{I} - \frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \mathbf{s}(n)\mathbf{s}^T(n) \right\} \Delta\hat{\mathbf{w}}(n) \right] -$$

$$E \left[ \frac{\mu r(n)}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \mathbf{s}(n) \right] \quad (4.15)$$

Assumindo que o ruído de observação é decorrelacionado do vetor sinal transformado e que possui média nula (Suposição III), chega-se a:

$$E[\Delta\hat{\mathbf{w}}(n+1)] = E \left[ \left\{ \mathbf{I} - \frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \mathbf{s}(n)\mathbf{s}^T(n) \right\} \Delta\hat{\mathbf{w}}(n) \right] \quad (4.16)$$

Assumindo agora que  $\hat{\mathbf{w}}(n)$  é independente de  $\mathbf{s}(n)$  (Suposição IV), temos que

$$E[\Delta\hat{\mathbf{w}}(n+1)] = E \left[ \left\{ \mathbf{I} - \frac{\mu}{\mathbf{s}^T(n)\mathbf{s}(n)} \begin{bmatrix} \ddots & & & \\ & \frac{\mathbf{s}_j^T(n)\mathbf{s}_j(n)}{\sigma_{s_j}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \mathbf{s}(n)\mathbf{s}^T(n) \right\} E[\Delta\hat{\mathbf{w}}(n)] \right] \quad (4.17)$$

A EQ. 4.17, pode ser escrita da seguinte maneira:

$$E [\Delta \hat{\mathbf{w}}(n+1)] = \left\{ \mathbf{I} - E \left[ \frac{\mu}{\sum_{i=0}^{M-1} \mathbf{s}_i^T(n) \mathbf{s}_i(n)} \begin{bmatrix} \frac{\mathbf{s}_0^T(n) \mathbf{s}_0(n) \mathbf{s}_0(n) \mathbf{s}_0^T(n)}{\sigma_{s_0}^2} & \dots & \frac{\mathbf{s}_0^T(n) \mathbf{s}_0(n) \mathbf{s}_0(n) \mathbf{s}_{M-1}^T(n)}{\sigma_{s_0}^2} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{s}_{M-1}^T(n) \mathbf{s}_{M-1}(n) \mathbf{s}_{M-1}(n) \mathbf{s}_0^T(n)}{\sigma_{s_{M-1}}^2} & \dots & \frac{\mathbf{s}_{M-1}^T(n) \mathbf{s}_{M-1}(n) \mathbf{s}_{M-1}(n) \mathbf{s}_{M-1}^T(n)}{\sigma_{s_{M-1}}^2} \end{bmatrix} \right] \right\} E [\Delta \hat{\mathbf{w}}(n)] \quad (4.18)$$

Admitindo-se a aproximação conhecida como “*averaging principle*” (SAMSON, 1983),  $E [g(x)/f(x)] \approx \frac{E [g(x)]}{E [f(x)]}$ , na matriz da EQ. 4.18 e também a suposição V, todos os termos fora da diagonal principal são nulos e obtemos:

$$E [\Delta \hat{\mathbf{w}}(n+1)] \approx \left\{ \mathbf{I} - \begin{bmatrix} \ddots & & & \\ & \frac{\mu E [\mathbf{s}_j^T(n) \mathbf{s}_j(n) \mathbf{s}_j(n) \mathbf{s}_j^T(n)]}{\sigma_{s_j}^2 E [\sum_{i=0}^{M-1} \mathbf{s}_i^T(n) \mathbf{s}_i(n)]} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \right\} E [\Delta \hat{\mathbf{w}}(n)] \quad (4.19)$$

Finalmente, considerando uma vez mais a Suposição II, que estabelece que o sinal  $\mathbf{s}(n)$  é modelada por processos estacionários com distribuição gaussiana e usando o teorema de fatoração dos momentos de quarta ordem (PAPOULIS, 1991), obtemos:

$$E [\Delta \hat{\mathbf{w}}(n+1)] = \left\{ \mathbf{I} - \frac{\mu}{\sum_{j=0}^{M-1} M_j \sigma_{s_j}^2} \begin{bmatrix} (M_0 + 2) \sigma_{s_0}^2 \mathbf{I}_0 & & & \\ & (M_1 + 2) \sigma_{s_1}^2 \mathbf{I}_1 & & \\ & & \ddots & \\ & & & (M_{M-1} + 2) \sigma_{s_{M-1}}^2 \mathbf{I}_{M-1} \end{bmatrix} \right\} E [\Delta \hat{\mathbf{w}}(n)]$$

$$= \begin{bmatrix} \left[ 1 - \frac{\mu(M_0 + 2) \sigma_{s_0}^2}{\sum_{j=0}^{M-1} M_j \sigma_{s_j}^2} \right] \mathbf{I}_0 & & & \\ & \ddots & & \\ & & \left[ 1 - \frac{\mu(M_{M-1} + 2) \sigma_{s_{M-1}}^2}{\sum_{j=0}^{M-1} M_j \sigma_{s_j}^2} \right] \mathbf{I}_{M-1} & \end{bmatrix} E [\Delta \hat{\mathbf{w}}(n)] \quad (4.20)$$

Aplicando a expressão anterior para  $n = 1, 2, 3, \dots$  chega-se a

$$E[\Delta\hat{\mathbf{w}}(n+1)] = \begin{bmatrix} \ddots & & & & \\ & \left(1 - \frac{\mu(M_j+2)\sigma_{s_j}^2}{M_0\sigma_{s_0}^2 + \dots + M_{M-1}\sigma_{s_{M-1}}^2}\right)^{n+1} & & & \\ & & \mathbf{I}_j & & \\ & & & \ddots & \\ & & & & \end{bmatrix} E[\Delta\hat{\mathbf{w}}(0)] \quad (4.21)$$

Para assegurar convergência do algoritmo no sentido da média, devemos impor a seguinte condição.

$$\left|1 - \frac{\mu(M_j+2)\sigma_{s_j}^2}{M_0\sigma_{s_0}^2 + \dots + M_{M-1}\sigma_{s_{M-1}}^2}\right| < 1 \quad \text{para } \forall j, \quad (4.22)$$

que pode ser reescrita da seguinte maneira:

$$0 < \mu < \frac{2(M_0\sigma_{s_0}^2 + \dots + M_{M-1}\sigma_{s_{M-1}}^2)}{(M_j+2)\sigma_{s_j}^2}, \quad \forall j. \quad (4.23)$$

Como  $\mu$  deve ser menor que o maior valor possível de  $(M_j+2)\sigma_{s_j}^2$ , podemos definir um limite máximo para  $\mu$ ,  $0 < \mu < \mu_{\max}$ , com

$$\boxed{\mu_{\max} = \frac{2(M_0\sigma_{s_0}^2 + \dots + M_{M-1}\sigma_{s_{M-1}}^2)}{\max_j (M_j+2)\sigma_{s_j}^2}} \quad (4.24)$$

Para determinar se as suposições e equações deduzidas são válidas, empregamos o exemplo apresentado na Seção 3.2, ou seja, um sistema Volterra de segunda ordem com um número de quatorze coeficientes; usamos neste exemplo a *wavelet* de Haar (DAUBECHIES, 1992), com dois níveis de decomposição ou resolução.

Considerando varias relação sinal ruído de  $-10$ ,  $-20$ ,  $-30$ , e  $-40$  dB, apresentamos a seguir um gráfico do MSE em dB em função do  $\mu$ . Isto é feito para determinar se o  $\mu_{\max}$ , dado pela EQ. 4.24, se encontra em uma faixa razoável.

Usando a expressão na EQ. 4.24, obtém-se um valor de  $\mu_{\max}$  de 0.06705, que é indicado com um asterisco numa das curvas da FIG. 4.2

Como pode-se observar, o  $\mu_{\max}$  encontra-se no ponto onde a curva inicia seu processo de divergência. Além disso se determina que a expressão encontrada independe da SNR

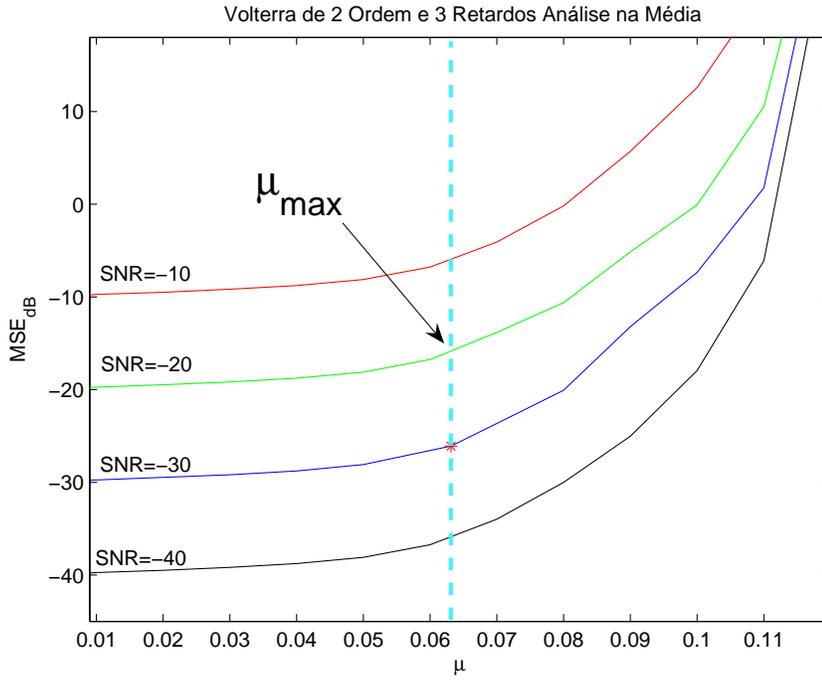


FIG. 4.2: Curva do  $MSE \times \mu$ .

do sinal. Outro experimento foi feito, desta vez com um Volterra de terceira ordem e dois retardos, com uma SNR de  $-30 \text{ dB}$ , para comprovar, num outro cenário, a validade desta análise. Neste caso, o resultado do  $\mu_{\max}$  foi de 0.08788 e os resultados deste segundo experimento encontram-se na FIG. 4.3.

Como pode ser observado na FIG. 4.3, o  $\mu_{\max}$  se encontra novamente próximo ao joelho da curva.

#### 4.4 ANÁLISE NA COVARIÂNCIA DO ALGORITMO WTD-LMSM

A análise na covariância para o algoritmo proposto é muito complexa e não foi possível obter uma expressão fechada. Assim sendo, para simplificar um pouco essa análise e obter expressões de algum valor prático, optou-se por assumir um  $\beta(n)$  definido EQ. 4.9, constante e substituído na equação de atualização dos coeficientes por  $\bar{\beta} = E[\beta(n)]$ . Este artifício resulta numa análise na covariância do algoritmo WTD-LMSM, que em realidade é uma versão simplificada do algoritmo proposto.

Doravante, iremos substituir  $\beta(n)$  por  $\bar{\beta}$  sendo que esta matriz pode ser obtida, de

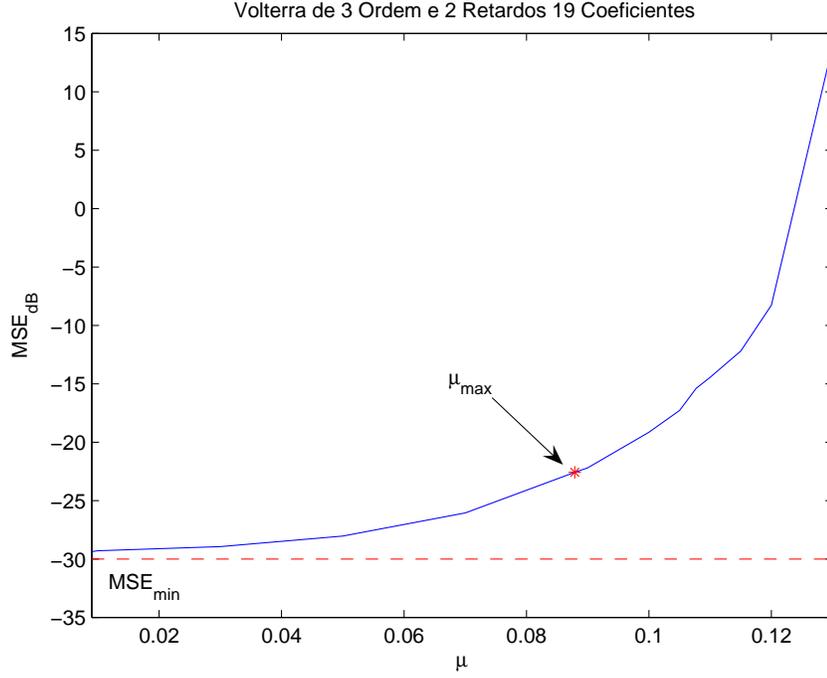


FIG. 4.3: Curva do  $\text{MSE} \times \mu$ .

forma aproximada, empregando-se o “*averaging principle*” (SAMSON, 1983):

$$\bar{\boldsymbol{\beta}} = \begin{bmatrix} \ddots & & & & \\ & \frac{\mu E[\mathbf{s}_j^T(n)\mathbf{s}_j(n)]}{\sigma_{s_j}^2 E[\mathbf{s}^T(n)\mathbf{s}(n)]} \mathbf{I}_j & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix} = \begin{bmatrix} \ddots & & & & \\ & \frac{\mu M_j}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \mathbf{I}_j & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix}. \quad (4.25)$$

Observe que, ao usarmos o vetor  $\bar{\boldsymbol{\beta}}$  da equação anterior, estaremos basicamente analisando o algoritmo WTD-LMSM com passos dados por

$$\mu_j = \frac{\mu M_j \sigma_{s_j}^2}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \quad (4.26)$$

Usando  $\bar{\boldsymbol{\beta}}$  acima ao invés de  $\boldsymbol{\beta}(n)$ , o modelo do sinal de referência da EQ. 4.12 e a definição de  $\Delta \hat{\mathbf{w}}(n)$  da EQ. 4.1, na EQ. 4.10 obtemos:

$$\Delta \hat{\mathbf{w}}(n+1) = \Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) r(n). \quad (4.27)$$

Seguindo o procedimento usado em (GALDINO, 2004) para analisar o desempenho do LMS, adotamos duas figuras de mérito para avaliar o desempenho dos algoritmos de filtragem adaptativa: a matriz de covariância do vetor de erros nos coeficientes  $\mathbf{K}(n)$  e o traço do erro médio quadrático nos coeficientes, aqui denotado por  $D(n)$ :

$$\mathbf{K}(n) = E [\Delta \mathbf{w}(n) \Delta \mathbf{w}^T(n)] = \mathbf{T}^T E [\Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n)] \mathbf{T} \quad (4.28)$$

$$D(n) = \text{tr} [\mathbf{K}(n)] = \text{tr} [\mathbf{T}^T \hat{\mathbf{K}}(n) \mathbf{T}] = \text{tr} [\mathbf{T} \mathbf{T}^T \hat{\mathbf{K}}(n)] = \text{tr} [\hat{\mathbf{K}}(n)] \quad (4.29)$$

Usando a EQ. 4.27,  $\hat{\mathbf{K}}(n)$  pode ser escrita como:

$$\hat{\mathbf{K}}(n+1) = E \left\{ \begin{aligned} & [\Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) r(n)] \\ & [\Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) - \bar{\boldsymbol{\beta}} \mathbf{s}(n) r(n)]^T \end{aligned} \right\} \quad (4.30)$$

Para simplificar a notação fazemos

$$\hat{\mathbf{K}}(n+1) = E \left[ [\Delta \hat{\mathbf{w}}(n) - \mathbf{a}(n) - \mathbf{b}(n)] [\Delta \hat{\mathbf{w}}(n) - \mathbf{a}(n) - \mathbf{b}(n)]^T \right] \quad (4.31)$$

onde

$$\mathbf{a}(n) = \bar{\boldsymbol{\beta}} \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \quad \Rightarrow \quad \mathbf{a}(n) = \Delta \hat{\mathbf{w}}(n) \mathbf{s}^T(n) \mathbf{s}^T(n) \bar{\boldsymbol{\beta}}$$

$$\mathbf{b}(n) = \bar{\boldsymbol{\beta}} \mathbf{s}(n) r(n) \quad \Rightarrow \quad \mathbf{b}(n) = \mathbf{s}^T(n) \bar{\boldsymbol{\beta}} r(n)$$

Desenvolvendo o produto acima, temos:

$$\begin{aligned} \hat{\mathbf{K}}(n+1) = & E \left[ \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) - \Delta \hat{\mathbf{w}}(n) \mathbf{a}^T(n) - \Delta \hat{\mathbf{w}}(n) \mathbf{b}^T(n) - \mathbf{a}(n) \Delta \hat{\mathbf{w}}^T(n) + \mathbf{a}(n) \mathbf{a}^T(n) + \right. \\ & \left. \mathbf{a}(n) \mathbf{b}^T(n) - \mathbf{b}(n) \Delta \hat{\mathbf{w}}^T(n) + \mathbf{b}(n) \mathbf{a}^T(n) + \mathbf{b}(n) \mathbf{b}^T(n) \right] \end{aligned} \quad (4.32)$$

Logo,

$$\begin{aligned} \hat{\mathbf{K}}(n+1) = & E \left[ \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \right] - E \left[ \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \bar{\boldsymbol{\beta}} \\ & - E \left[ \Delta \hat{\mathbf{w}}(n) \mathbf{s}^T(n) r(n) \right] \bar{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \right] \\ & + \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \bar{\boldsymbol{\beta}} \\ & + \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \mathbf{s}^T(n) r(n) \right] \bar{\boldsymbol{\beta}} \\ & - \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \Delta \hat{\mathbf{w}}^T(n) r(n) \right] + \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \Delta \hat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) r(n) \right] \bar{\boldsymbol{\beta}} \\ & + \bar{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) r^2(n) \right] \bar{\boldsymbol{\beta}} \end{aligned}$$

Levando em consideração que o ruído  $r(n)$  é decorrelacionado dos demais sinais e que tem média zero (Suposição III), ficamos com:

$$\begin{aligned}\widehat{\mathbf{K}}(n+1) = & E \left[ \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \right] - E \left[ \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \\ & - \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \right] \\ & + \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \\ & + \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} E \left[ r^2(n) \right]\end{aligned}\quad (4.33)$$

Os processos  $\widehat{\mathbf{w}}(n)$  e  $\mathbf{s}(n)$  na condição de regime estacionário são considerados estatisticamente independentes, conforme a Suposição V. Diante disto, é razoável aceitar que  $\Delta \widehat{\mathbf{w}}(n)$  e  $\mathbf{s}(n)$  também são estatisticamente independentes em regime estacionário. Desta forma, temos

$$\begin{aligned}\widehat{\mathbf{K}}(n+1) = & \widehat{\mathbf{K}}(n) - \widehat{\mathbf{K}}(n) E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \\ & - \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \widehat{\mathbf{K}}(n) + \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \\ & + \sigma_r^2 \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}}\end{aligned}\quad (4.34)$$

Considerando-se a Suposição I,  $E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right]$  é uma matriz diagonal. Assim sendo, tem-se que

$$E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} = \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right]. \quad (4.35)$$

Desta forma, a EQ. 4.34 pode ser re-escrita como:

$$\begin{aligned}\widehat{\mathbf{K}}(n+1) = & \widehat{\mathbf{K}}(n) - \widehat{\mathbf{K}}(n) \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] - E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \widehat{\mathbf{K}}(n) \\ & + \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \\ & + \sigma_r^2 \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}}\end{aligned}\quad (4.36)$$

Aplicando a definição do  $\text{tr}[\cdot]$  em EQ. 4.36, temos:

$$\begin{aligned}\text{tr} \left[ \widehat{\mathbf{K}}(n+1) \right] = & \text{tr} \left[ \widehat{\mathbf{K}}(n) \right] - 2 \text{tr} \left[ \widehat{\mathbf{K}}(n) \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \right] \\ & + \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right], \\ & + \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right]\end{aligned}\quad (4.37)$$

Empregando a definição de  $D(n)$ , a EQ. 4.37 fica

$$\begin{aligned}D(n+1) = & D(n) - 2 \text{tr} \left[ \widehat{\mathbf{K}}(n) E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right] \\ & + \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \widehat{\mathbf{w}}(n) \Delta \widehat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right] \\ & + \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right]\end{aligned}\quad (4.38)$$

Fazendo:

$$\begin{aligned}\mathcal{A}(n) &= -2\text{tr} \left[ \widehat{\mathbf{K}}(n) E [\mathbf{s}(n)\mathbf{s}^T(n)] \overline{\boldsymbol{\beta}} \right], \\ \mathcal{B}(n) &= \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n)\mathbf{s}^T(n) \Delta \widehat{\boldsymbol{w}}(n) \Delta \widehat{\boldsymbol{w}}^T(n) \mathbf{s}(n)\mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right] \text{ e} \\ \mathcal{C}(n) &= \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}} E [\mathbf{s}(n)\mathbf{s}^T(n)] \overline{\boldsymbol{\beta}} \right],\end{aligned}$$

tem-se que

$$D(n+1) = D(n) + \mathcal{A}(n) + \mathcal{B}(n) + \mathcal{C}(n) \quad (4.39)$$

O segundo termo do lado direito da EQ. 4.38, denotado por  $\mathcal{A}(n)$ , é dado por

$$\mathcal{A}(n) = -2\text{tr} \left[ \widehat{\mathbf{K}}(n) E [\mathbf{s}(n)\mathbf{s}^T(n)] \overline{\boldsymbol{\beta}} \right] \quad (4.40)$$

$$\begin{aligned}&= -2\text{tr} \left[ \widehat{\mathbf{K}}(n) \mathbf{R}_s \overline{\boldsymbol{\beta}} \right] \\ &= -2\text{tr} \left[ \widehat{\mathbf{K}}(n) \cdot \begin{bmatrix} \ddots & & & \\ & \sigma_{s_j}^2 \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \ddots & & & \\ & \frac{\mu M_j}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \right] \\ &= -2\text{tr} \left[ \widehat{\mathbf{K}}(n) \begin{bmatrix} \ddots & & & \\ & \frac{\mu M_j \sigma_{s_j}^2}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \right] = -2\text{tr} \left[ \widehat{\mathbf{K}}(n) \begin{bmatrix} \ddots & & & \\ & \sigma_{s_j}^2 \overline{\beta}_j \mathbf{I}_j & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \right] \quad (4.41)\end{aligned}$$

com

$$\overline{\beta}_j = \frac{\mu M_j}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \quad (4.42)$$

Admitindo-se que os coeficientes a serem estimados são independentes, pode-se aproximar a matriz  $\widehat{\mathbf{K}}(n)$ , por uma matriz diagonal. Além disso, para simplificar a análise, os elementos da diagonal principal serão aproximados por:

$$E [\|\widehat{\boldsymbol{w}}_{0j} - \widehat{\boldsymbol{w}}_j(n)\|^2] = \frac{D(n)M_j}{L}, \quad \text{com } L = \sum_{j=0}^{M-1} M_j, \quad (4.43)$$

ou seja, admitiu-se que o erro nos coeficientes é uniformemente distribuído entre os elementos da diagonal.

Com tais aproximações, o termo  $\mathcal{A}(n)$  fica dado por:

$$\mathcal{A}(n) \approx -\frac{2D(n)}{L} \sum_{j=0}^{M-1} M_j^2 \sigma_{s_j}^2 \bar{\beta}_j \quad (4.44)$$

onde, como visto anteriormente,  $\bar{\beta}_j = \frac{\mu M_j}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2}$ .

Continuando o desenvolvimento da EQ. 4.38, trabalhamos a seguir sobre o terceiro termo da direita, denotado por  $\mathcal{B}(n)$ .

$$\begin{aligned} \mathcal{B}(n) &= \text{tr} \left[ \bar{\beta} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \right] \bar{\beta} \right] \\ &= E \left[ \text{tr} \left[ \bar{\beta} \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \mathbf{s}(n) \mathbf{s}^T(n) \bar{\beta} \right] \right]. \end{aligned} \quad (4.45)$$

Como  $\text{tr}[\mathbf{AB}] = \text{tr}[\mathbf{BA}]$ , desde que os produtos  $\mathbf{AB}$  e  $\mathbf{BA}$  sejam definidos, pode-se escrever que

$$\begin{aligned} \mathcal{B}(n) &= E \left[ \text{tr} \left[ \mathbf{s}(n) \mathbf{s}^T(n) \bar{\beta}^2 \mathbf{s}(n) \mathbf{s}^T(n) \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \right] \right] \\ &= \text{tr} \left[ E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \bar{\beta}^2 \mathbf{s}(n) \mathbf{s}^T(n) \right] E \left[ \Delta \hat{\mathbf{w}}(n) \Delta \hat{\mathbf{w}}^T(n) \right] \right] \\ &= \text{tr} \left[ \hat{\mathbf{\Lambda}} \hat{\mathbf{K}}(n) \right] \end{aligned} \quad (4.46)$$

onde

$$\hat{\mathbf{\Lambda}} = E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \bar{\beta}^2 \mathbf{s}(n) \mathbf{s}^T(n) \right] \quad (4.47)$$

Em EQ. 4.47, necessitamos computar o valor esperado da seguinte expressão:

$$\begin{aligned} & \begin{bmatrix} \mathbf{s}_0(n) \mathbf{s}_0^T(n) & \mathbf{s}_0(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_0(n) \mathbf{s}_{M-1}^T(n) \\ \mathbf{s}_1(n) \mathbf{s}_0^T(n) & \mathbf{s}_1(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_1(n) \mathbf{s}_{M-1}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{M-1}(n) \mathbf{s}_0^T(n) & \mathbf{s}_{M-1}(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_{M-1}(n) \mathbf{s}_{M-1}^T(n) \end{bmatrix} \begin{bmatrix} \bar{\beta}_0^2 \mathbf{I}_0 & & & \\ & \bar{\beta}_1^2 \mathbf{I}_1 & & \\ & & \ddots & \\ & & & \bar{\beta}_{M-1}^2 \mathbf{I}_{M-1} \end{bmatrix} \times \\ & \begin{bmatrix} \mathbf{s}_0(n) \mathbf{s}_0^T(n) & \mathbf{s}_0(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_0(n) \mathbf{s}_{M-1}^T(n) \\ \mathbf{s}_1(n) \mathbf{s}_0^T(n) & \mathbf{s}_1(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_1(n) \mathbf{s}_{M-1}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{M-1}(n) \mathbf{s}_0^T(n) & \mathbf{s}_{M-1}(n) \mathbf{s}_1^T(n) & \cdots & \mathbf{s}_{M-1}(n) \mathbf{s}_{M-1}^T(n) \end{bmatrix} \end{aligned}$$

Como  $\widehat{\mathbf{K}}(n)$  é uma matriz diagonal dominante e se deseja obter o traço de  $\widehat{\mathbf{\Lambda}}(n)\mathbf{T}(n)$ , os valores  $\widehat{\mathbf{\Lambda}}(k)$  que não estão na diagonal principal não têm influência no resultado final. Assim sendo, serão calculados os valores dos termos da diagonal principal dessa matriz.

Tomando o  $j$ -ésimo bloco da diagonal de  $\widehat{\mathbf{\Lambda}}$  (esperança dos produtos da diagonal da equação anterior), temos

$$\sum_{i=0}^{M-1} \overline{\beta}_i^2 E \left[ \mathbf{s}_j(n) \mathbf{s}_i^T(n) \mathbf{s}_i(n) \mathbf{s}_j^T(n) \right],$$

cujo  $p$ -ésimo elemento da diagonal é dado por

$$\sum_{i=0}^{M-1} \overline{\beta}_i^2 E \left[ \sum_{m=0}^{M_j-1} s_{jp}^2(n) s_{im}^2(n) \right] \quad (4.48)$$

Da definição de  $\mathcal{B}(n)$  em EQ. 4.46, do  $p$ -ésimo elemento da diagonal do  $j$ -ésimo bloco diagonal da matriz  $\widehat{\mathbf{\Lambda}}$  expresso em EQ. 4.48 e da aproximação dos elementos da diagonal de  $\widehat{\mathbf{K}}(n)$  feito em EQ. 4.43, temos

$$\begin{aligned} \mathcal{B}(n) &= \text{tr} \left[ \widehat{\mathbf{\Lambda}} \widehat{\mathbf{K}}(n) \right] \\ &= \sum_{j=0}^{M-1} \left\{ \sum_{p=0}^{M_j-1} \left( \sum_{i=0}^{M-1} \overline{\beta}_i^2 E \left[ \sum_{m=0}^{M_j-1} s_{jp}^2(n) s_{im}^2(n) \right] \right) \right\} \frac{D(n) M_j}{L} \\ &= \frac{D(n)}{L} \sum_{j=0}^{M-1} M_j \sum_{i=0}^{M-1} \overline{\beta}_i^2 \sum_{p=0}^{M_j-1} \sum_{m=0}^{M_j-1} E \left[ s_{jp}^2(n) s_{im}^2(n) \right] \end{aligned} \quad (4.49)$$

Avaliando os somatórios da equação anterior nos casos onde  $i = j$  e  $i \neq j$ , após algumas manipulações algébricas, chegamos ao resultado final  $\mathcal{B}(n)$  dado por

$$\mathcal{B}(n) = \frac{D(n)}{L} \sum_{j=0}^{M-1} M_j \sigma_{s_j}^4 \overline{\beta}_j^2 (3 - M_j) + \frac{\text{tr} \left[ \overline{\beta}^2 \right] D(n)}{L} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 \quad (4.50)$$

onde

$$\text{tr} \left[ \overline{\beta}^2 \right] = \sum_{j=0}^{M-1} M_j \overline{\beta}_j^2 = \frac{\mu^2}{\sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \sum_{j=0}^{M-1} M_j^3 \quad (4.51)$$

Para o último termo da EQ. 4.39, denotado como  $\mathcal{C}(n)$ , é fácil notar que:

$$\begin{aligned}
\mathcal{C}(n) &= \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}} E \left[ \mathbf{s}(n) \mathbf{s}^T(n) \right] \overline{\boldsymbol{\beta}} \right] \\
&= \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}} \mathbf{R}_s \overline{\boldsymbol{\beta}} \right] \\
&= \sigma_r^2 \text{tr} \left[ \overline{\boldsymbol{\beta}}^2 \mathbf{R}_s \right] \\
&= \sigma_r^2 \text{tr} \begin{bmatrix} \ddots & & & \\ & \ddots & & \\ & & \overline{\beta}_j^2 \sigma_{s_j}^2 \mathbf{I}_j & \\ & & & \ddots \end{bmatrix}
\end{aligned}$$

ou seja

$$\mathcal{C}(n) = \sigma_r^2 \sum_{j=0}^{M-1} M_j \sigma_{s_j}^2 \overline{\beta}_j^2 \quad (4.52)$$

Substituindo as EQ. 4.44, EQ. 4.50 e EQ. 4.52 na EQ. 4.39, obtemos a expressão final para  $D(n+1)$  dada por

$$\begin{aligned}
D(n+1) &= D(n) - 2 \frac{D(n)}{L} \sum_{j=0}^{M-1} M_j^2 \sigma_{s_j}^2 \overline{\beta}_j \\
&\quad + \frac{D(n)}{L} \sum_{j=0}^{M-1} M_j \sigma_{s_j}^4 \overline{\beta}_j^2 (3 - M_j) + \frac{\text{tr} \left[ \overline{\boldsymbol{\beta}}^2 \right] D(n)}{L} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 \\
&\quad + \sigma_r^2 \sum_{j=0}^{M-1} M_j \sigma_{s_j}^2 \overline{\beta}_j^2
\end{aligned} \quad (4.53)$$

Colocando  $D(n)$ , em evidência, temos:

$$\begin{aligned}
D(n+1) &= \\
&\left[ 1 - \frac{2}{L} \sum_{j=0}^{M-1} M_j^2 \sigma_{s_j}^2 \overline{\beta}_j + \frac{1}{L} \sum_{j=0}^{M-1} M_j \sigma_{s_j}^4 \overline{\beta}_j^2 (3 - M_j) + \frac{\text{tr} \left[ \overline{\boldsymbol{\beta}}^2 \right]}{L} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 \right] D(n) \\
&\quad + \sigma_r^2 \sum_{j=0}^{M-1} M_j \sigma_{s_j}^2 \overline{\beta}_j^2
\end{aligned} \quad (4.54)$$

Ou, ainda,

$$D(n+1) = \kappa D(n) + \eta, \quad (4.55)$$

na qual

$$\kappa = 1 - \frac{2}{L} \sum_{j=0}^{M-1} M_j^2 \sigma_{s_j}^2 \overline{\beta}_j + \frac{1}{L} \sum_{j=0}^{M-1} M_j \sigma_{s_j}^4 \overline{\beta}_j^2 (3 - M_j) + \frac{\text{tr} \left[ \overline{\boldsymbol{\beta}}^2 \right]}{L} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 \quad (4.56)$$

e

$$\eta = \left( \sigma_r^2 \sum_{j=0}^{M-1} M_j \sigma_{s_j}^2 \bar{\beta}_j^2 \right) \quad (4.57)$$

Podemos, então, expressar a norma média quadrática dos coeficientes da seguinte maneira:

$$\begin{aligned} D(n+1) &= \kappa D(n) + \eta \\ D(1) &= \kappa D(0) + \eta \\ D(2) &= \kappa(\kappa D(0) + \eta) + \eta = \kappa^2 D(0) + \kappa\eta + \eta \\ D(M) &= \kappa^M D(0) + \sum_{l=1}^M \kappa^{M-l} \eta \end{aligned} \quad (4.58)$$

Analisando-se a equação acima verifica-se que a convergência do algoritmo WTD-LMSM na covariância é assegurada fazendo-se  $|\kappa| < 1$ .

Se esta condição for atendida, em regime permanente, tem-se:

$$D(\infty) = \lim_{n \rightarrow \infty} D(n) = \frac{\eta}{1 - \kappa} \quad (4.59)$$

ou seja:

$$\boxed{D(\infty) = \frac{\sigma_r^2 \sum_{j=0}^{M-1} M_j \sigma_{s_j}^2 \bar{\beta}_j^2}{\frac{2}{L} \sum_{j=0}^{M-1} M_j^2 \sigma_{s_j}^2 \bar{\beta}_j - \frac{1}{L} \sum_{j=0}^{M-1} M_j \sigma_{s_j}^4 \bar{\beta}_j^2 (3 - M_j) - \frac{\text{tr}[\bar{\beta}^2]}{L} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4}} \quad (4.60)$$

A seguir, expressamos a condição para convergência em termos do passo do algoritmo. Da condição de convergência, tem-se

$$|\kappa| = |1 - b\mu + a\mu^2| < 1 \quad (4.61)$$

As raízes de  $1 - b\mu + a\mu^2$  são

$$\mu = \frac{b \pm \sqrt{b^2 - 4a}}{2a} \quad (4.62)$$

sendo

$$b = \frac{2}{L \sum_{i=0}^{M-1} M_i \sigma_{s_i}^2} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^2 \quad (4.63)$$

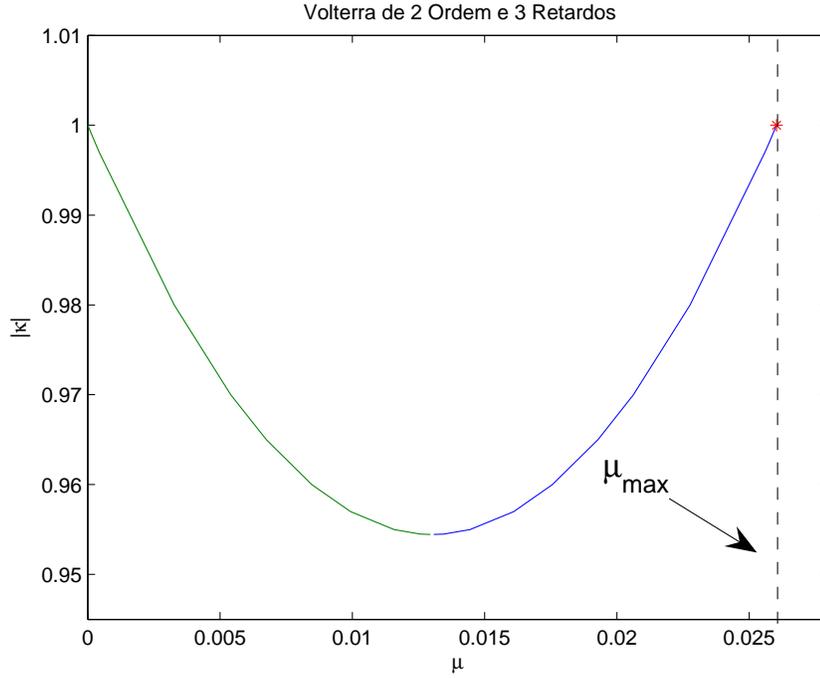


FIG. 4.4: Valores de  $\mu$

e

$$b = \frac{1}{L \left( \sum_{i=0}^{M-1} M_i \sigma_{s_i}^2 \right)^2} \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 (3 - M_j) + \frac{1}{L \left( \sum_{i=0}^{M-1} M_i \sigma_{s_i}^2 \right)^2} \sum_{i=0}^{M-1} M_i^3 \sum_{j=0}^{M-1} M_j^3 \sigma_{s_j}^4 \quad (4.64)$$

Para verificar a faixa de valores de  $\mu$  que garante convergência, a FIG. 4.4 apresenta  $|\kappa|$  em função de  $\mu$  para os valores de  $\mu$  tais que  $|\kappa| \leq 1$ . Os resultados apresentados nesta figura foram obtidos considerando um filtro de Volterra de segunda ordem com três retardos, transformada de Haar e uma SNR de  $-30$  dB.

Os resultados apresentados mostram que, para esta aproximação,  $\mu_{\max} = 0.026029$ . Empregando este valor de  $\mu_{\max}$  e inserindo-o na FIG. 4.2, observa-se na FIG. 4.5 que  $\mu_{\max}$  obtido pela análise na covariância é menor que o  $\mu_{\max}$  obtido pelo análise na média.

Este resultado, além de coerente (o  $\mu_{\max}$ ) da análise na covariância de outros algoritmos é geralmente menor que o encontrado pela análise na média, parece ser uma boa proposta de valor a ser utilizado na prática uma vez que está no início do joelho e proporciona um

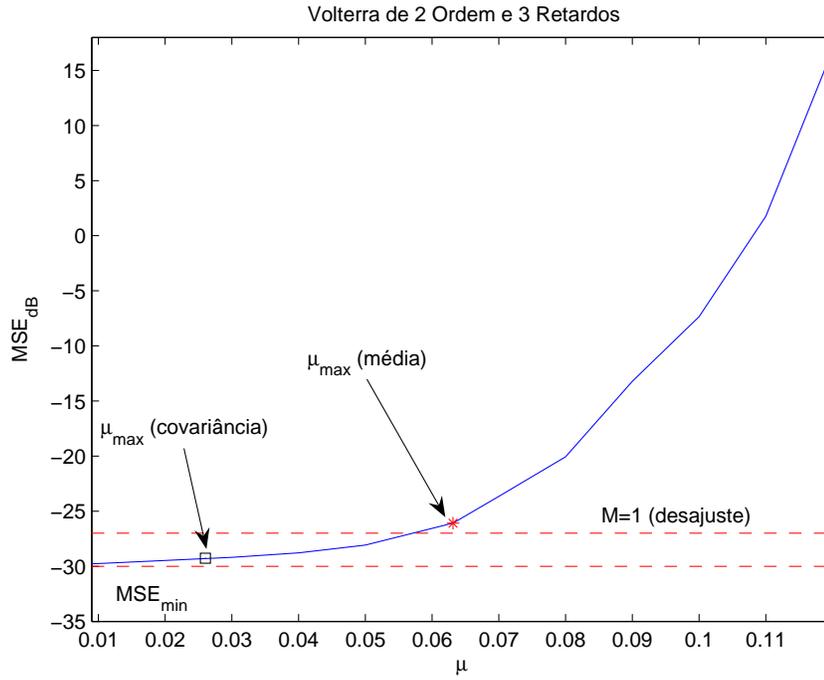


FIG. 4.5: Valores de  $\mu_{\max}$  das duas análises.

desajuste bem menor que 1 (indicado na curva com  $M = 1$ )<sup>2</sup>.

Como se mostrou na FIG. 4.2, o  $\mu_{\max}$  para diferentes SNR, fazemos o mesmo na FIG. 4.6, e como se observa o resultado mostra um resultado razoável que não depende da SNR e menor que o encontrado com a análise na média.

#### 4.5 RESUMO

Neste capítulo, fez-se uma análise do algoritmo WTD-LMS buscando uma escolha adequada do  $\mu$ . Para o algoritmo proposto foi feita uma análise na média; contudo, tendo em vista que a análise na covariância para este algoritmo mostrou-se muito complexa, optou-se por uma simplificação que resultou numa análise aproximada para o algoritmo WTD-LMSM.

Os resultados teóricos obtidos apresentaram boa concordância com aqueles obtidos a partir de simulações de Monte-Carlo, indicando que as hipóteses simplificadoras adotadas

---

<sup>2</sup>A definição de desajuste é dada por  $M = \frac{\xi_{\text{exc}}}{\xi_{\text{min}}}$  onde  $\xi_{\text{exc}} = \lim_{n \rightarrow \infty} \xi(n) - \xi_{\text{min}}$  e  $\xi_{\text{min}}$  é o menor valor possível de MSE ( $\text{MSE}_{\text{min}}$ ).

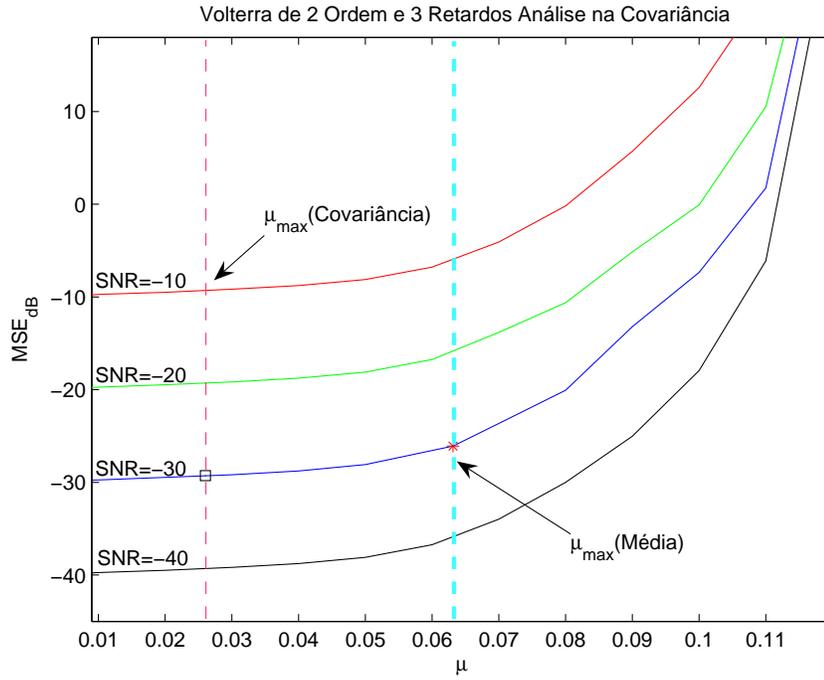


FIG. 4.6: Valores de  $\mu_{max}$  das duas análises diferentes SNR.

foram razoáveis.

A partir da análise para o algoritmo WTD-LMSM, foi proposta uma maneira de inicializar o passo por nível de decomposição para este algoritmo (dado pela EQ. 4.26). Isto foi uma contribuição adicional pois ajustar os  $M$  valores de  $\mu$  era uma tarefa, até então, baseada em tentativa e erro.

Além disso se observo que os resultados de  $\mu_{max}$  na covariância como na média são válidos, para qualquer SNR, sem ser prejudicados pelo incremento o diminuição do SNR.

## 5 CONCLUSÕES E COMENTÁRIOS FINAIS

### 5.1 CONCLUSÕES

Nesta dissertação, após uma breve introdução e revisão de conceitos e algoritmos básicos (lineares e especialmente os não-lineares) e uma pequena explicação sobre a transformada *wavelet*, foi feita uma avaliação do desempenho de vários algoritmos adaptativos no domínio da transformada *wavelet*, genericamente denominado como WTD-LMS. Uma das principais dificuldades desta família de algoritmos é a escolha do valor do passo uma vez que, à medida que aumenta o grau do polinômio de Volterra, aumenta a quantidade de passos, um por cada ordem do polinômio. Para minimizar este problema, foi proposto uma nova versão deste algoritmo onde somente um valor é necessário para a inicialização do algoritmo.

Esta nova abordagem resultou numa melhor velocidade de convergência. Como é sabido, os algoritmos que empregam núcleos de Volterra tem a desvantagem da sua elevada complexidade computacional. Este fato motivou a proposta de redução de ordem dos coeficientes no domínio da transformada, mediante o emprego da propriedade de concentração de energia da wavelet; esta proposta foi denotada como RO-WTD-LMS (Reduced Order WTD-LMS).

Continuando com o estudo do algoritmo WTD-LMS proposto, empregou-se como ferramenta de análise a esperança do vetor de erros nos coeficientes, tendo como motivação determinar o valor de passo máximo ( $\mu_{\max}$ ). Isto foi feito com êxito de modo que pudemos definir um limite dentro da zona de convergência e facilitar a escolha do passo.

Também procedeu-se uma análise na covariância; contudo, devido à complexidade de tal análise para o algoritmo proposto, optou-se por uma simplificação resultando na análise do algoritmo WTD-LMSM. Como resultado desta análise, restringiu-se um pouco mais o valor do  $\mu_{\max}$ .

Pôde-se concluir que o algoritmo estudado tem boa velocidade de convergência comparado aos algoritmos LMS e NLMS, além de ter uma baixa complexidade computacional frente ao algoritmo RLS. O algoritmo proposto foi usado com sucesso em filtragem de Volterra. Em qualquer caso, a análise realizada, ao estabelecer limites para o valor do

passo, torna sua aplicação mais fácil.

## 5.2 TRABALHOS FUTUROS

Algoritmos adaptativos que sejam suficientemente rápidos e estáveis em aplicações não-lineares, a um custo computacional atraente, continuam sendo um desafio a ser investigado; além disto, realizar análises mais completas de algoritmos da família WTD-LMS constituem um fértil campo de pesquisa. Com base nisto, propomos os seguintes tópicos para pesquisas futuras:

- Investigar a possibilidade de realizar uma análise na covariância sem empregar as aproximações mencionada nesta dissertação, e comparar com os resultados obtidos;
- Do algoritmo RO-WTD-LMS, seria interessante determinar a quantidade de coeficientes a ser eliminada mediante o emprego de uma expressão analítica e não só de uma maneira empírica. Isto seria uma atraente opção quando se necessita reduzir o tamanho do núcleo de Volterra;
- Determinar, mediante a análise da matriz  $\mathbf{T}$ , se o desempenho do algoritmo WTD-LMS depende da família de *wavelets* empregadas bem como explorar, na construção dessa matriz novas transformadas tais como pacotes *wavelets*;
- Testar o algoritmo em sistemas esparsos, ou seja, aqueles onde os coeficientes tem grande quantidade de zeros.

## 5.3 COMENTÁRIOS FINAIS

Esta dissertação enfatizou um assunto que tem sido pouco estudado devido à falta de ferramentas para sua análise; por isto, este trabalho, pelos seus bons resultados, oferece novas perspectivas para investigações futuras.

Também a partir deste trabalho, vislumbra-se um estudo mais detalhado, incluindo análise e implementações em diferentes cenários, dos algoritmos com redução de coeficientes, necessários em aplicações que requerem de complexidade computacional reduzida.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

- ATTALLAH, S. **The Wavelet Transform-Domain LMS Algorithm: a More Practical Approach.** *In: IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 47(3):209–213, March 2000.
- BATISTA, E. L. O., TOBIAS, O. J. e SEARA, R. **Filtros Volterra Adaptativos: Interpolado e Parcialmente Interpolado.** *XXI Simpósio Brasileiro de Telecomunicações, SBT 2004*, Setembro 2004.
- BERNAL O., C. P., GALDINO, J. F. e APOLINÁRIO JR., J. A. **O Algoritmo LMS no Domínio da Transformada Wavelet Aplicado a Filtros de Volterra.** *XXII Simpósio Brasileiro de Telecomunicações, SBT 2005*, Setembro 2005.
- BURRUS, C. S., GOPINATH, R. A. e GUA, H. **Introduction to Wavelets and Wavelet Transforms - A Primer.** Prentice Hall, New Jersey, 1998.
- CHANG, P. S. e WILLSON JR., A. N. **Analysis of Conjugate Gradient Algorithms for Adaptive Filtering.** *In: IEEE Transactions on Signal Processing*, 48:409–418, 2000.
- CHANG, S. L. e OGUNFUNMI, T. **Stochastic Gradient Based Third-Order Volterra System Identification by Using Nonlinear Wiener Adaptive Algorithm.** *In: IEE Proceedings Vision, Image & Signal Processing*, 150(2):90–98, April 2003.
- CHENG, C. e POWERS, E. J. **Optimal Volterra Kernel Estimation Algorithms for a Nonlinear Communication System For PSK and QAM Inputs.** *In: IEEE Transactions on Signal Processing*, 49(1):147–163, January 2001.
- DAUBECHIES, I. **Ten Lectures on Wavelets.** SIAM, Philadelphia, PA, 1992. Notes From the 1990 CBMS-NSF Conference on *Wavelets* and Applications at Lowell, Ma., 1992.
- DINIZ, P. S. R. **Adaptive Filtering: Algorithms and Practical Implementations.** 2nd Edition, Kluwer Academic Publishers, Boston, 2002.
- FERMO, A., CARINI, A. e SICURANZA, L. **Simplified Volterra Filters For Acoustic Echo Cancellation in GSM Receivers.** *In: Proc. EUSIPCO 2000, Tampere, Finland*, September 2000.
- GALDINO, J. F., PINTO, E. L. e DE ALENCAR, M. S. **Desempenho do Algoritmo LMS na Identificação de Canais Variantes no Tempo e seu Emprego em Esquemas de Recepção MLSE-PSP.** *In: Revista da Sociedade Brasileira de Telecomunicações - SBrT*, 18:273–284, 2003.

- GALDINO, J. F., PINTO, E. L. e DE ALENCAR, M. S. **Analytical Performance of the LMS Algorithm on the Estimation of Wide Sense Stationary Channels.** *In: IEEE Transactions on Communications*, 52(6):982–991, June 2004.
- HAYKIN, S. ***Adaptive Filter Theory***. New Jersey: Prentice-Hall, Englewood-Cliffs, 1996.
- HERLEY, C. e VETTERLI, M. **Orthogonal Time-Varying Filter Banks and Wavelet Packets.** *In: IEEE Transactions on Signal Processing*, 42(10):2650–2663, October 1994.
- HOSUR, S. e TEWFIK, A. H. **Wavelet Transform Domain FIR Filtering.** *In: IEEE Transactions on signal Processing*, 45(3):617–630, March 1997.
- KALLURI, S. e ARCE, G. R. **A General Class of Nonlinear Normalized Adaptive Filtering Algorithms.** *In: IEEE Transactions on Signal Processing*, 47(8):2262–2272, August 1999.
- MALLAT, S. ***A Wavelet Tour of Signal Processing***. Second Edition, Academic Press, 1998.
- MATHEWS, V. J. **Adaptive Polynomial Filters.** *In: IEEE Signal Processing Magazine*, 6:10–26, November 1991.
- MATHEWS, V. J. e SICURANZA, G. L. ***Polynomial Signal Processing***. Wiley-Intercience: John Wiley and Sons, 2000.
- NARAYAN, S. S., PETERSON, A. M. e NARASIMHA, M. J. **Transform Domain LMS Algorithm.** *In: IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(3):609–615, June 1983.
- OPPENHEIM, A. P., SCHAFER, R. W. e BUCK, J. R. ***Discrete-Time Signal Processing***. Allan V. Oppenheim: Prentice Hall, 1999.
- PAPOULIS, A. ***Probability, Random Variables, and Stochastic Processes***. McGraw-Hill, 1991.
- SAMSON, C. G. e REDDY, V. U. **Fixed Point Error Analysis of the Normalised Ladder Algorithm.** *In: IEEE Transactions Acoustics, Speech, and Signal Processing*, ASSP-31(5):1177–1191, October 1983.
- SCHETZEN, M. ***The Volterra and Wiener Theories of Nonlinear Systems***. John Wiley & Sons, Inc., Estados Unidos, 1980.
- TAN, L. e JIANG, J. **Adaptive Volterra Filters for Active Control of Nonlinear Noise Processes.** *In: IEEE Transactions on Signal Processing*, 49(8):1667–1676, August 2001.
- TEWFIK, A. H. e KIM, M. **Fast Positive Definite Linear System Solvers.** *In: IEEE Transactions on Signal Processing*, 42(3):572–584, March 1994.

VIDAKOVIC, B. *Statistical Modeling by Wavelets*. John Wiley, United States, 1999.

ZANUY, M. F. *Modelo Predictivo no Lineal de la Señal de Voz aplicado a Codificación y Reconocimiento de Locutor*. Tesis Doctoral–UPC/Universitat Politècnica de Catalunya, Barcelona, España, 1998.