

**EDUARDO LUIZ ORTIZ BATISTA**

**FILTRO VOLTERRA ADAPTATIVO:  
ANÁLISE ESTATÍSTICA E  
ALGORITMOS SIMPLIFICADOS**

**FLORIANÓPOLIS**

**2004**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**

**FILTRO VOLTERRA ADAPTATIVO:  
ANÁLISE ESTATÍSTICA E  
ALGORITMOS SIMPLIFICADOS**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia Elétrica.

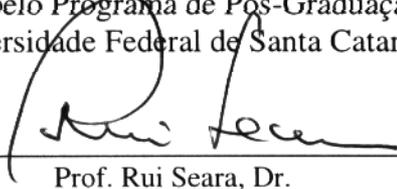
**EDUARDO LUIZ ORTIZ BATISTA**

Florianópolis, Julho de 2004.

# **FILTRO VOLTERRA ADAPTATIVO: ANÁLISE ESTATÍSTICA E ALGORITMOS SIMPLIFICADOS**

Eduardo Luiz Ortiz Batista

'Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Processamento Digital de Sinais, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

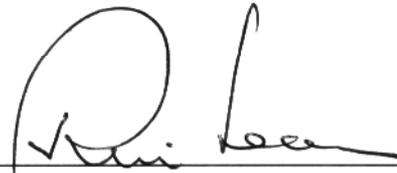


Prof. Rui Seara, Dr.  
Orientador

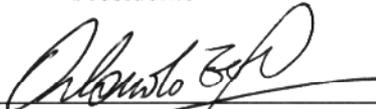


Prof. Denizar Cruz Martins, Dr.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

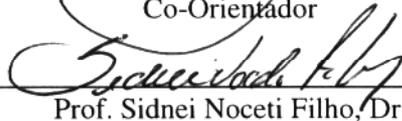
Banca Examinadora:



Prof. Rui Seara, Dr.  
Presidente



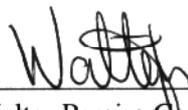
Prof. Orlando J. Tobias, Dr.  
Co-Orientador



Prof. Sidnei Noceti Filho, Dr.



Prof. Hans Helmut Zörn, Ph.D.



Prof. Walter Pereira Carpes Junior, Dr.

## **AGRADECIMENTOS**

- Ao meu orientador, Prof. Rui Seara, e ao meu co-orientador, Prof. Orlando J. Tobias, por todos os ensinamentos, confiança, apoio e excelente relacionamento.
- Aos membros da banca examinadora, por todas as críticas e sugestões que resultaram em aperfeiçoamentos no trabalho final.
- Aos meus pais, por todo amor, incentivo e por tudo o que me ensinaram e continuam ensinando.
- Ao meu irmão, pela amizade e companheirismo.
- À Fran, por todo amor, carinho e compreensão.
- A todos colegas do LINSE e da UFSC, em especial ao Elton Fontão.
- À CAPES, pelo apoio financeiro.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

## **FILTRO VOLTERRA ADAPTATIVO: ANÁLISE ESTATÍSTICA E ALGORITMOS SIMPLIFICADOS**

**Eduardo Luiz Ortiz Batista**

Julho/2004

Orientador: Prof. Rui Seara, Dr.

Co-orientador: Prof. Orlando J. Tobias, Dr.

Área de Concentração: Comunicações e Processamento de Sinais

Palavras-chave: Filtragem adaptativa, filtros não-lineares, filtros Volterra.

Número de páginas: 72

**RESUMO:** Esta dissertação aborda o problema de filtragem adaptativa não-linear, visando uma classe particular de filtros que é a dos Volterra adaptativos. O emprego dessas estruturas de filtragem tem crescido muito devido à grande demanda por soluções específicas para problemas não-lineares em sistemas adaptativos. O filtro Volterra é uma opção interessante, visto que sua operação de filtragem pode ser representada pelo produto interno de dois vetores: um contendo os coeficientes do filtro e o outro, os produtos cruzados das amostras do sinal de entrada. Tal procedimento é análogo ao dos filtros lineares, permitindo a utilização de ferramentas similares às já desenvolvidas para estes últimos. Os estudos aqui apresentados têm por objetivo tratar dois problemas ainda existentes em filtragem adaptativa não-linear, considerando-se os filtros Volterra: a escassez de ferramentas de análise e a alta complexidade computacional exigida por tais filtros. Assim, uma análise estatística desses filtros adaptados pelo algoritmo LMS é desenvolvida. Tal análise é fundamentada na linearidade existente entre os coeficientes e o sinal de saída do filtro adaptativo. Em relação à complexidade computacional, duas estruturas simplificadas são propostas. Tais estruturas constituem-se de um filtro interpolador de entrada seguido por um filtro Volterra adaptativo esparso. Algumas características específicas dessa classe de estruturas são exploradas com o objetivo de melhorar seu desempenho. Finalmente, uma modelagem estatística dos filtros Volterra adaptativos de complexidade reduzida é apresentada, utilizando-se uma abordagem via restrições.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

## **ADAPTIVE VOLTERRA FILTER: STATISTICAL ANALYSIS AND SIMPLIFIED ALGORITHMS**

**Eduardo Luiz Ortiz Batista**

July/2004

Advisor: Prof. Rui Seara, Dr.

Co-advisor: Prof. Orlando J. Tobias, Dr.

Concentration area: Communications and Signal Processing.

Keywords: Adaptive filtering, nonlinear filters, Volterra filters.

Number of pages: 72

**ABSTRACT:** This dissertation addresses the nonlinear adaptive filtering problem, focusing on a particular class of filters called the adaptive Volterra filters. There are a large number of engineering applications requiring the use of these filtering structures. The Volterra filter is an interesting option, since its filtering operation can be represented by the inner product of two vectors: one containing the filter coefficients and the other, the cross products of the input signal. Such an operation is analogous to the one performed in linear adaptive filters, permitting the use of similar tools as those already developed for the latter filters. The use of Volterra structures in adaptive filtering applications has two major drawbacks: the insufficiency of existing analysis tools and the computational burden for its implementation. Regarding the former, a statistical analysis concerning the referred filter is presented. Such an analysis is based on the linearity between the filter coefficients and the corresponding output signal. Secondly, two simplified structures are proposed. Such structures are based on interpolated FIR filters, which consist of an interpolator along with a sparse adaptive Volterra filter. Some particular features of this adaptive structure are also exploited to improve its performance. Finally, a statistical modeling of the reduced complexity adaptive Volterra filters is presented, by using a constrained approach.

# SUMÁRIO

<b>1</b>	<b><i>Introdução</i></b>	<b>1</b>
1.1	Filtros Adaptativos	1
1.2	Filtros Adaptativos Lineares	2
1.2.1	Solução de Wiener	3
1.2.2	Método <i>Steepest Descent</i>	4
1.2.3	Algoritmo LMS	5
1.3	Filtros Adaptativos Não-Lineares	6
1.4	Organização da Dissertação	7
<b>2</b>	<b><i>Filtro Volterra</i></b>	<b>8</b>
2.1	Filtro Volterra como um Operador Pseudolinear	8
2.2	Divisão em Blocos	10
2.3	Número de Coeficientes e Complexidade Computacional	10
2.4	Considerações	11
<b>3</b>	<b><i>Análise Estatística do Filtro Volterra Adaptado pelo Algoritmo LMS</i></b>	<b>12</b>
3.1	Vetor de Coeficientes Ótimo	12
3.2	Comportamento Médio dos Pesos	13
3.3	Vetor de Erro dos Pesos	15
3.4	Estrutura da Matriz de Autocorrelação de Entrada	15
3.5	Erro Quadrático Médio	18
3.6	Aproximação para o Momento de Segunda Ordem	19
3.7	Simulações	21
3.8	Considerações	28
<b>4</b>	<b><i>Filtros Volterra Adaptativos Interpolado e Parcialmente Interpolado</i></b>	<b>29</b>
4.1	Filtro Volterra Adaptativo Interpolado	29
4.2	Filtro Volterra Adaptativo Parcialmente Interpolado	32
4.3	Simulações	33
4.4	Considerações	40
<b>5</b>	<b><i>Análise Estatística com Restrição nos Coeficientes de Filtros Volterra Adaptados pelo Algoritmo LMS e sua Aplicação a Filtros Interpolados e com Atraso</i></b>	<b>41</b>

<b>5.1</b>	<b>Vetor de Coeficientes Ótimo com Restrições</b>	<b>41</b>
<b>5.2</b>	<b>Equação de Atualização dos Pesos com Restrições</b>	<b>43</b>
<b>5.3</b>	<b>Comportamento Médio dos Pesos</b>	<b>44</b>
<b>5.4</b>	<b>Vetor de Erro dos Pesos</b>	<b>45</b>
<b>5.5</b>	<b>Erro Quadrático Médio</b>	<b>45</b>
<b>5.6</b>	<b>Aproximação para o Momento de Segunda Ordem</b>	<b>45</b>
<b>5.7</b>	<b>Aplicações da Análise com Restrições nos Pesos</b>	<b>46</b>
<b>5.7.1</b>	<b>Filtro Volterra Adaptativo Interpolado</b>	<b>47</b>
<b>5.7.2</b>	<b>Filtro Volterra Adaptativo Parcialmente Interpolado</b>	<b>48</b>
<b>5.7.3</b>	<b>Filtro Volterra Adaptativo com Atraso</b>	<b>49</b>
<b>5.7.4</b>	<b>Filtro Volterra Adaptativo Simplificado</b>	<b>50</b>
<b>5.8</b>	<b>Simulações</b>	<b>50</b>
<b>5.9</b>	<b>Considerações</b>	<b>63</b>
<b>6</b>	<b><i>Discussões e Conclusões</i></b>	<b>64</b>
<b>6.1</b>	<b>Sumário e Discussão dos Resultados</b>	<b>64</b>
<b>6.2</b>	<b>Contribuições</b>	<b>65</b>
<b>6.3</b>	<b>Propostas para Trabalhos Futuros</b>	<b>66</b>
	<b><i>Apêndice A - Princípio da Ortogonalidade</i></b>	<b>67</b>
	<b><i>Referências Bibliográficas</i></b>	<b>69</b>

## LISTA DE FIGURAS

Figura 1.1 – Diagrama de blocos de um filtro adaptativo no contexto de estimação de um sinal desejado a partir de um sinal de entrada. _____	2
Figura 2.1 – Decomposição do filtro Volterra em uma estrutura de blocos. _____	10
Figura 3.1 – Diagrama em blocos de um filtro Volterra aplicado a um problema de estimação de sinal. _____	12
Figura 3.2 – Diagrama em blocos de um filtro Volterra adaptativo aplicado a um problema de estimação de sinal. _____	14
Figura 3.3. Exemplo 3.1. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,5\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (Linha sólida) modelo dado por (3.12). _____	22
Figura 3.4. Exemplo 3.1. Evolução de $\xi(n)$ com $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44). ___	23
Figura 3.5. Exemplo 3.2. Evolução de $E[\mathbf{h}_v(n)]$ , com $\mu = 0,2\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (linha sólida) modelo dado por (3.12). _____	24
Figura 3.6. Exemplo 3.2. Evolução de $\xi(n)$ , com $\mu = 0,2\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44). ___	24
Figura 3.7. Exemplo 3.3. Evolução de $E[\mathbf{h}_v(n)]$ , com $\mu = 0,5\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (Linha sólida) modelo dado por (3.12). _____	25
Figura 3.8. Exemplo 3.3. Evolução de $\xi(n)$ , com $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação(média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44). _____	26
Figura 3.9. Exemplo 3.4. Evolução de $E[\mathbf{h}_v(n)]$ , com $\mu = 0,25\mu_{\max}$ . (Linha pontilhada com marcadores) simulação(média de 100 realizações); (Linha sólida) modelo dado por (3.12). _____	27

Figura 3.10. Exemplo 3.4. Evolução de $\xi(n)$ , com $\mu = 0,25\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44). ____	27
Figura 4.1 – Diagrama de blocos do filtro Volterra adaptativo interpolado. _____	31
Figura 4.2 – Diagrama de blocos de um filtro Volterra adaptativo parcialmente interpolado de segunda ordem. _____	32
Figura 4.3. Exemplo 4.1. Evolução do erro quadrático médio (média de 200 realizações). 36	
Figura 4.4. Exemplo 4.1. Evolução do erro quadrático médio (média de 200 realizações). 37	
Figura 4.5. Exemplo 4.2. Evolução do erro quadrático médio (média de 200 realizações). 37	
Figura 4.6. Exemplo 4.2. Evolução do erro quadrático médio (média de 200 realizações). 38	
Figura 4.7. Exemplo 4.3. Evolução do erro quadrático médio (média de 200 realizações). (a) Comparação com o filtro Volterra interpolado. (b) Comparação com o filtro Volterra parcialmente interpolado. _____	39
Figura 5.1 – Diagrama de blocos do filtro Volterra adaptativo interpolado. _____	47
Figura 5.2 – Diagrama de blocos do filtro Volterra adaptativo com atraso aplicado a um problema de estimação de sinal. _____	49
Figura 5.3. Exemplo 5.1. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22). _____	52
Figura 5.4. Exemplo 5.1. Simulação (média de 100 realizações) usando $\mu = 0,5\mu_{\max}$ . (Linha sólida) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ . _____	52
Figura 5.5. Exemplo 5.1. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,25\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22). _____	53
Figura 5.6. Exemplo 5.1. Simulação (média de 100 realizações) usando $\mu = 0,25\mu_{\max}$ . (Linha sólida) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ . _	53
Figura 5.7. Exemplo 5.1. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,05\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22). _____	54

Figura 5.8. Exemplo 5.1. Simulação (média de 100 realizações) usando $\mu = 0,05\mu_{\max}$ . (Linha sólida) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada) $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ .	54
Figura 5.9. Exemplo 5.1. Evolução de $\xi(n)$ com $\mu = 0,5\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	55
Figura 5.10. Exemplo 5.1. Evolução de $\xi(n)$ com $\mu = 0,25\mu_{\max}$ . (Linha preta) simulação(média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	55
Figura 5.11. Exemplo 5.1. Evolução de $\xi(n)$ com $\mu = 0,05\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	56
Figura 5.12. Exemplo 5.2. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,125\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).	57
Figura 5.13. Exemplo 5.2. Evolução de $\xi(n)$ com $\mu = 0,125\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	57
Figura 5.14. Exemplo 5.3. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,2\mu_{\max}$ e sinal branco de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).	59
Figura 5.15. Exemplo 5.3. Evolução de $\xi(n)$ com $\mu = 0,2\mu_{\max}$ e sinal branco de entrada. (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	59
Figura 5.16. Exemplo 5.3. Evolução de $E[\mathbf{h}_v(n)]$ com $\mu = 0,1\mu_{\max}$ e sinal branco de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).	60
Figura 5.17. Exemplo 5.3. Evolução de $\xi(n)$ com $\mu = 0,1\mu_{\max}$ e sinal branco de entrada. (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).	60
Figura 5.18. Exemplo 5.3. Evolução de $E[\mathbf{h}_v(n)]$ , com $\mu = 0,1\mu_{\max}$ e sinal colorido de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).	61

- Figura 5.19. Exemplo 5.3. Evolução de  $\xi(n)$  com  $\mu = 0,1\mu_{\max}$  e sinal colorido de entrada.  
 (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por  
 (5.27) e (5.31). \_\_\_\_\_ 61
- Figura 5.20. Exemplo 5.4. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,1\mu_{\max}$ . (Linha sólida)  
 simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo  
 dado por (5.22). \_\_\_\_\_ 62
- Figura 5.21. Exemplo 5.4. Evolução de  $\xi(n)$  com  $\mu = 0,1\mu_{\max}$ . (Linha preta) simulação  
 (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31). \_\_\_\_\_ 63

## LISTA DE TABELAS

Tabela 4.1 – Complexidade: Filtro Volterra adaptativo comparado com sua versão interpolada _____	31
Tabela 4.2 – Número de coeficientes para as diferentes estruturas com $L = 2$ _____	33

## Introdução

Desde os primeiros trabalhos de pesquisa desenvolvidos por Bernard Widrow [1], que deram origem ao algoritmo *least mean square* (LMS), até os dias de hoje, o uso de filtros adaptativos para resolver os mais diversos problemas em processamento digital de sinais evoluiu muito. A necessidade de lidar com aplicações nas quais a resposta requerida não é conhecida *a priori*, ou ainda quando ela é variante no tempo, levou ao desenvolvimento destes filtros, que possuem a habilidade de acompanhar variações do sistema ao longo do tempo. Outros fatores também contribuíram para a sua difusão, como por exemplo, o grande interesse no desenvolvimento de pesquisas na área de filtragem adaptativa e o surgimento de novos processadores digitais de sinais de alto desempenho. Atualmente os filtros adaptativos são utilizados com sucesso nas mais diversas áreas do processamento digital de sinais. Alguns exemplos são: identificação de sistemas [2], equalização de canais [2], cancelamento de interferências [2], cancelamento de eco acústico [3] e controle ativo de ruído/vibração [4].

### 1.1 Filtros Adaptativos

Em sentido geral, o adjetivo “adaptativo” pode ser compreendido considerando um sistema que procura se ajustar a algum fenômeno que ocorre em suas redondezas, ou seja, o sistema procura acomodar seus parâmetros para atingir um objetivo previamente estabelecido que depende tanto do seu próprio estado quanto do fenômeno em si, realizando assim uma “adaptação”. Já o termo “filtro”, é relativo ao sistema que realiza o processo de adaptação. No contexto do processamento de sinais, filtros podem ser entendidos como dispositivos que removem componentes indesejados de um sinal. Dessa forma, podemos entender os “filtros adaptativos” como sistemas capazes de remover alguns componentes de um sinal de entrada, possuindo ainda a habilidade de se adaptar para satisfazer certos pré-requisitos de desempenho.

A Figura 1.1 mostra um diagrama típico de filtragem adaptativa. Nesse caso, o filtro é empregado para estimar o *senal desejado* a partir de um *senal de entrada*, assumindo que existe alguma correlação entre eles. O processo de adaptação ou de obtenção dos parâmetros (coeficientes) do filtro, é geralmente realizado através da otimização de uma função de desempenho ou função custo previamente definida. Para esse caso, o que se pretende é que o *senal de saída* seja o mais próximo possível do *senal desejado*, com relação a algum dado critério.

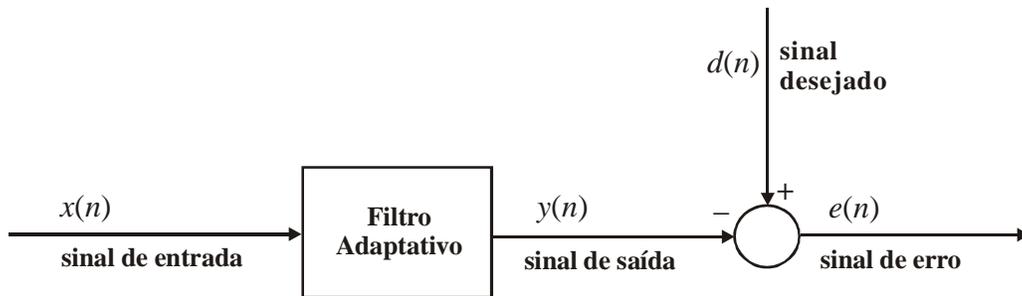


Figura 1.1 – Diagrama de blocos de um filtro adaptativo no contexto de estimação de um sinal desejado a partir de um sinal de entrada.

## 1.2 Filtros Adaptativos Lineares

Os filtros adaptativos lineares são sistemas amplamente utilizados para resolver os mais diversos problemas em procesamento digital de sinais. Sua simplicidade matemática aliada a uma base teórica muito bem fundamentada tornam relativamente simples o seu projeto e implementação. Além disso, os filtros adaptativos lineares oferecem desempenho satisfatório em diversas aplicações, como pode ser verificado por alguns exemplos de aplicações discutidos em [5] e [6].

Um sistema é considerado linear se ele satisfaz o *princípio da superposição* [7]. Essa propriedade leva a resultados interessantes na teoria de sistemas lineares, como o fato de um sistema linear ser completamente caracterizado por sua resposta ao impulso.

A estrutura mais comumente utilizada para a implementação de filtros adaptativos lineares é a estrutura transversal [5]. Trata-se de um filtro linear com resposta ao impulso finita (FIR), com relação entrada e saída dada por

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i), \quad (1.1)$$

onde  $x(n)$  e  $y(n)$  caracterizam a entrada e a saída do filtro, respectivamente; e  $w_i(n)$ , para  $i = 0, 1, \dots, N-1$ , representa os coeficientes do filtro correspondente. Essa relação pode ainda ser escrita na forma vetorial, resultando em

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n), \quad (1.2)$$

com  $\mathbf{w}(n) = [w_0(n) \ w_1(n) \ \dots \ w_{N-1}(n)]^T$  e  $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$ .

Existem ainda filtros lineares adaptativos recursivos, ou com resposta ao impulso infinita (IIR), porém devido às dificuldades envolvidas em seu processo de adaptação, sua aplicação em filtragem adaptativa é bastante restrita [5].

### 1.2.1 Solução de Wiener

Considere-se agora o emprego do filtro transversal descrito por (1.2) usando o diagrama ilustrado na Figura 1.1, com sinal de entrada  $x(n)$  real e estacionário e sinal desejado, denotado por  $d(n)$ , também real e estacionário. Utilizando um vetor de pesos fixo dado por  $\mathbf{w}$  e denotando o sinal de erro por  $e(n)$ , podemos escrever o erro instantâneo como

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^T\mathbf{x}(n). \quad (1.3)$$

O filtro de Wiener utiliza o erro quadrático médio  $\xi = E[e^2(n)]$  como a função custo a ser minimizada [5]. Tal função é escolhida por sua simplicidade matemática e ainda por apresentar um único mínimo global sob algumas condições, como por exemplo, para o caso em que filtros FIR lineares são considerados. Assim, elevando-se (1.3) ao quadrado e tomando-se seu valor esperado, obtém-se

$$\xi = E[d^2(n)] - 2E[\mathbf{w}^T\mathbf{x}(n)d(n)] + E[\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}]. \quad (1.4)$$

Levando-se em conta aqui que  $\mathbf{w}$  não é um vetor aleatório, é possível retirá-lo do operador esperança. Além disso, com  $x(n)$  e  $d(n)$  estacionários, podemos definir a matriz de autocorrelação do sinal de entrada  $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ , e ainda a matriz de correlação entre o sinal de entrada e o sinal desejado  $\mathbf{p} = E[d(n)\mathbf{x}(n)]$ . Dessa forma, (1.4) pode ser reescrita como

$$\xi = E[d^2(n)] - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}. \quad (1.5)$$

Para se obter o valor de  $\mathbf{w}$  que minimize a correspondente função custo, faz-se o seu gradiente em relação aos coeficientes igual a zero, isto é  $\nabla\xi = 0$ . Assim,

$$\nabla\xi = 2\mathbf{R}\mathbf{w} - 2\mathbf{p} = 0. \quad (1.6)$$

Denotando-se o vetor de coeficientes que minimiza a função custo por  $\mathbf{w}_o$ , a partir de (1.6) é possível obter a equação de Wiener-Hopf, dada por

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}. \quad (1.7)$$

Esta equação possui a seguinte solução

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}, \quad (1.8)$$

assumindo-se que exista a matriz inversa de  $\mathbf{R}$ .

### 1.2.2 Método *Steepest Descent*

Uma alternativa para se encontrar o vetor ótimo de coeficientes que minimize a função custo é o uso de um algoritmo que, sendo iniciado com um vetor arbitrário de coeficientes, progressivamente realize uma adaptação em direção ao vetor ótimo. O método *steepest descent* é um dos métodos que realizam tal tarefa. Partindo do princípio que a função custo é convexa, pequenos passos são dados na direção em que essa função decresce mais rapidamente, ou seja, no sentido contrário ao seu gradiente. Assim, de forma iterativa o vetor de coeficientes ótimo pode ser determinado. O *steepest descent* é formalmente descrito através da seguinte equação recursiva:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla_n\xi, \quad (1.9)$$

onde  $\mu$  é o passo de adaptação do algoritmo e  $\nabla_n\xi$  denota o vetor gradiente  $\nabla\xi$  calculado no ponto  $\mathbf{w} = \mathbf{w}(n)$ . Para o caso do filtro transversal descrito por (1.2), temos o gradiente da função custo dado por (1.6). Substituindo-se então (1.6) em (1.9) e isolando-se  $\mathbf{w}(n)$  do lado direito da equação, obtém-se

$$\mathbf{w}(n+1) = (\mathbf{I} - 2\mu\mathbf{R})\mathbf{w}(n) + 2\mu\mathbf{p}. \quad (1.10)$$

Assim, a partir de (1.10) é possível obter o vetor ótimo de pesos iterativamente.

Como podemos notar em (1.10), a principal desvantagem do método *steepest descent* é a necessidade de conhecer *a priori* a matriz  $\mathbf{R}$  de autocorrelação do sinal de entrada e o vetor  $\mathbf{p}$  de correlação cruzada entre o sinal desejado e o sinal de entrada, o que

difícilmente dispõe-se em aplicações práticas. Assim, uma solução interessante e mais conveniente é o uso do algoritmo *least mean square* (LMS).

### 1.2.3 Algoritmo LMS

O algoritmo *least mean square* (LMS), ou algoritmo do menor valor quadrático médio, é o algoritmo mais comumente utilizado em aplicações de filtragem adaptativa. Isso se deve ao fato de se tratar de um algoritmo relativamente simples de implementar e robusto. Além disso, muitos pesquisadores nas últimas décadas têm se dedicado firmemente ao estudo do algoritmo LMS, resultando em aperfeiçoamentos e modificações que permitem o seu emprego nas mais diversas situações [5].

O algoritmo LMS convencional é uma implementação estocástica do método *steepest descent* [5]. Isso é feito através da substituição da função custo  $\xi = E[e^2(n)]$  por sua estimativa instantânea  $\hat{\xi}(n) = e^2(n)$ . Assim, para o filtro FIR linear, substitui-se  $\xi$  por  $\hat{\xi}(n)$  em (1.9), obtendo-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla e^2(n). \quad (1.11)$$

Como o operador gradiente é definido pelo vetor

$$\nabla = \left[ \frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \dots \quad \frac{\partial}{\partial w_{N-1}} \right]^T, \quad (1.12)$$

tem-se o  $i$ -ésimo elemento do vetor  $\nabla e^2(n)$  dado por

$$\frac{\partial e^2(n)}{\partial w_i} = 2e(n) \frac{\partial e(n)}{\partial w_i}. \quad (1.13)$$

Substituindo-se (1.3) no lado direito de (1.13), obtém-se

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n)x(n-i). \quad (1.14)$$

Assim, tem-se o vetor  $\nabla e^2(n)$  dado por

$$\nabla e^2(n) = -2e(n)\mathbf{x}(n). \quad (1.15)$$

Finalmente, substituindo-se (1.15) em (1.11), a equação recursiva do algoritmo LMS é obtida. Portanto,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n). \quad (1.16)$$

Como pode ser observado, a mais incontestável característica do algoritmo LMS é sua simplicidade. Para a sua implementação são requeridas  $N + 1$  multiplicações para calcular a saída do filtro e obter  $2\mu e(n)$ , e ainda outras  $N$  multiplicações para obter  $2\mu e(n)\mathbf{x}(n)$ , resultando em um total de  $2N + 1$  multiplicações por amostra, somadas às  $2N$  adições também necessárias. Além disso, esse algoritmo é também relativamente robusto, o que significa dizer que pequenas incertezas do modelo e perturbações com pouca energia resultam apenas em pequenos erros de estimação. Por outro lado, a principal desvantagem do algoritmo LMS é sua convergência lenta quando o sinal de entrada é fortemente autocorrelacionado [5], o que pode algumas vezes prejudicar sua aplicação.

### 1.3 Filtros Adaptativos Não-Lineares

Como visto na Seção 1.2, os filtros adaptativos lineares são bastante simples do ponto de vista de implementação e por isso sua aplicação é bastante difundida e estudada. No entanto, muitas das aplicações que requerem o uso de filtros adaptativos possuem elementos não-lineares em seu ambiente. Um bom exemplo são aplicações que envolvem situações de saturação. Para esse tipo de aplicação um filtro adaptativo não-linear passa a ser uma solução muito mais atraente ou mesmo indispensável dependendo dos pré-requisitos de desempenho adotados.

Do ponto de vista prático, a procura pelos parâmetros que otimizem a função custo de um filtro adaptativo não-linear é geralmente lenta e difícil. Isso ocorre pois uma solução analítica simples geralmente não é possível, ao contrário do que acontece com os filtros adaptativos lineares. Assim, as implementações de filtros adaptativos não-lineares normalmente requerem uma alta carga computacional. Com o aumento da capacidade de processamento dos DSPs (*digital signal processors*) ocorrido nas últimas décadas, tornou-se possível a aplicação de filtragem adaptativa não-linear em maior escala. Alguns exemplos podem ser encontrados em [8], [9], [10] e [11].

No entanto, muitos filtros adaptativos não-lineares têm sua complexidade exponencialmente dependente do tamanho da memória de tal forma que, mesmo com uma necessidade de memória relativamente pequena, uma quantidade muito grande de operações por amostra é requerida. Um exemplo de filtros que enfrentam tal problema são os filtros Volterra adaptativos [8] [12]. Outro problema defrontado pelos filtros adaptativos

não-lineares é a escassez de ferramentas de análise e síntese. Para resolver completamente tais problemas muito esforço de pesquisa ainda deve ser feito. O desenvolvimento de implementações de complexidade reduzida, algoritmos com melhor desempenho de convergência e análise teórica do comportamento dos filtros adaptativos não-lineares são bons exemplos do que pode ser feito para prover pesquisadores e engenheiros de ferramentas comparáveis às existentes para filtros lineares.

#### **1.4 Organização da Dissertação**

Nesta dissertação de mestrado, alguns estudos, desenvolvidos com o objetivo de atenuar os problemas existentes no emprego de filtros Volterra adaptativos, são apresentados. No Capítulo 2 o filtro Volterra é descrito procurando enfatizar suas vantagens, desvantagens e particularidades. O Capítulo 3 apresenta uma análise estatística do filtro Volterra adaptado com o algoritmo LMS para sinais de entrada Gaussianos. No Capítulo 4 dois algoritmos simplificados para a implementação subótima de filtros Volterra adaptativos são desenvolvidos. Esses algoritmos procuram explorar uma estrutura semelhante a dos já conhecidos filtros lineares adaptativos interpolados [13], em conjunto com as particularidades do filtro Volterra. O Capítulo 5 apresenta uma análise estatística com restrição nos coeficientes do filtro Volterra adaptativo. Essa análise pode ser aplicada tanto para os algoritmos propostos no Capítulo 4, quanto para outras implementações do filtro Volterra adaptado pelo algoritmo LMS que possuam restrições nos pesos, como os filtros Volterra adaptativos com atraso [14] e os filtros Volterra simplificados [15]. Resultados obtidos através de simulações também são apresentados no decorrer do trabalho para comprovar a validade das expressões e dos algoritmos propostos. Finalmente, no Capítulo 6 as conclusões desta dissertação e as propostas para trabalhos futuros são apresentadas.

## Filtro Volterra

O primeiro problema enfrentado para lidar com sistemas não-lineares em filtragem adaptativa é como especificar e caracterizar esses sistemas. Nesse contexto, o filtro Volterra representa uma das mais interessantes opções. O matemático Vito Volterra foi quem primeiramente introduziu o que hoje é conhecido como série de Volterra [16], e suas primeiras aplicações para análise de sistemas não-lineares foram realizadas por Norbert Wiener [17] no *Massachusetts Institute of Technology* (MIT). A linearidade entre o sinal de saída de tal filtro e seus coeficientes permite o emprego de algoritmos adaptativos já conhecidos e consagrados em seu processo adaptativo. Com isso, torna-se direta sua utilização em diversas aplicações, como, por exemplo, no cancelamento de eco acústico [18], controle ativo de processos [12], e equalização de canais de satélite [19]. Neste capítulo o filtro Volterra é apresentado ressaltando-se suas principais características.

### 2.1 Filtro Volterra como um Operador Pseudolinear

Um filtro Volterra causal discreto pode ser considerado como um filtro linear acrescido de blocos não-lineares. Essa característica fica evidenciada a partir de sua relação de entrada e saída [8], dada por

$$\begin{aligned}
 y(n) = & \sum_{m_1=0}^N h_1(m_1)x(n-m_1) \\
 & + \sum_{m_1=0}^N \sum_{m_2=m_1}^N h_2(m_1, m_2)x(n-m_1)x(n-m_2) + \dots \\
 & + \sum_{m_1=0}^N \dots \sum_{m_p=m_{p-1}}^N h_p(m_1, \dots, m_p)x(n-m_1) \dots x(n-m_p),
 \end{aligned} \tag{2.1}$$

onde  $x(n)$  e  $y(n)$  representam respectivamente o sinal de entrada e o sinal de saída;  $h_p(m_1, \dots, m_p)$  representa os coeficientes de ordem  $p$ ;  $N$  é o tamanho da memória; e  $P$  é a ordem do filtro. É importante ressaltar que os elementos redundantes, sem perda de generalidade [12], são removidos de (2.1). Tem-se, então, um filtro composto por um bloco

de primeira ordem (linear), cujos coeficientes são representados por  $h_1$ , seguido de blocos não-lineares ( $h_2, h_3, \dots, h_p$ ). Denotando-se a saída do bloco de ordem  $p$  por  $y_p(n)$ , podemos reescrever (2.1) como segue:

$$y(n) = \sum_{p=1}^P y_p(n), \quad (2.2)$$

com  $y_p(n)$  dado por

$$y_p(n) = \sum_{m_1=0}^{N-1} \sum_{m_2=m_1}^{N-1} \cdots \sum_{m_p=m_{p-1}}^{N-1} h_p(m_1, m_2, \dots, m_p) \times \prod_{k=1}^p x(n - m_k). \quad (2.3)$$

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

$$y_1(n) = \mathbf{h}_1^T \mathbf{x}_1(n), \quad (2.4)$$

onde  $\mathbf{x}_1(n)$  representa o vetor de entrada de primeira ordem e  $\mathbf{h}_1$  o vetor de coeficientes. Para os demais blocos, um procedimento análogo é possível realizando-se uma operação vetorial pseudolinear. Por exemplo, para um bloco de segunda ordem, pode-se escrever o vetor de entrada e o de coeficientes como

$$\mathbf{x}_2(n) = [x^2(n) \ x(n)x(n-1) \ \cdots \ x(n)x(n-N+1) \ x^2(n-1) \ \cdots \ x^2(n-N+1)]^T, \quad (2.5)$$

$$\mathbf{h}_2 = [h_2(0,0) \ h_2(0,1) \ \cdots \ h_2(0,N-1) \ h_2(1,1) \ \cdots \ h_2(N-1,N-1)]^T, \quad (2.6)$$

resultando em uma relação entrada/saída dada por

$$y_2(n) = \mathbf{h}_2^T \mathbf{x}_2(n). \quad (2.7)$$

A mesma operação pode ser realizada para os outros blocos não-lineares do filtro Volterra, resultando na seguinte relação, agora para um bloco de ordem  $p$  qualquer,

$$y_p(n) = \mathbf{h}_p^T \mathbf{x}_p(n). \quad (2.8)$$

Definindo-se então o vetor de entrada do filtro Volterra como

$$\mathbf{x}_v(n) = [\mathbf{x}_1^T(n) \ \mathbf{x}_2^T(n) \ \cdots \ \mathbf{x}_p^T(n)]^T, \quad (2.9)$$

e o vetor de coeficientes por

$$\mathbf{h}_v = [\mathbf{h}_1^T \ \mathbf{h}_2^T \ \cdots \ \mathbf{h}_p^T]^T, \quad (2.10)$$

temos a relação entrada/saída do filtro Volterra (2.1), reescrita da seguinte forma:

$$y(n) = \mathbf{h}_v^T \mathbf{x}_v(n). \quad (2.11)$$

Em (2.11) tem-se a operação de filtragem do filtro Volterra escrita de maneira pseudolinear como o produto interno entre dois vetores. Essa característica é muito importante, permitindo o desenvolvimento das expressões para os momentos de primeira e segunda ordens do filtro Volterra, discutidas a seguir. Além disso, pode-se notar, nesta expressão, a característica linear entre o sinal de saída e os coeficientes do filtro. Tal propriedade permite a aplicação direta de algoritmos adaptativos, usualmente empregados em filtros lineares, também para os filtros Volterra.

## 2.2 Divisão em Blocos

Outra característica também muito importante é a possibilidade de decomposição do filtro Volterra em uma estrutura paralela de blocos, o que pode ser notado a partir de (2.2). Essa decomposição está ilustrada pela Figura 2.1, onde cada um dos blocos corresponde a um filtro de ordem  $p$ . Tal condição é bastante vantajosa do ponto de vista prático uma vez que, em algumas aplicações, certos blocos podem ser desconsiderados, ou, ainda, estratégias de redução de complexidade podem ser aplicadas a blocos que demandem um maior número de coeficientes.

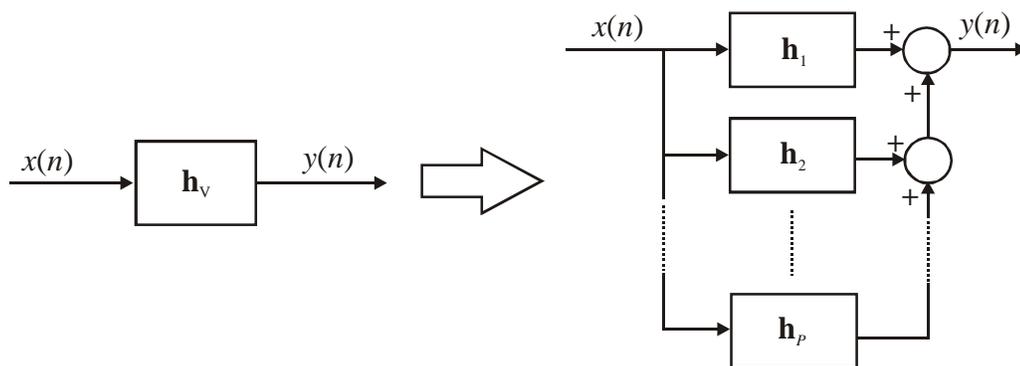


Figura 2.1 – Decomposição do filtro Volterra em uma estrutura de blocos.

## 2.3 Número de Coeficientes e Complexidade Computacional

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. Como é bem conhecido, a complexidade computacional necessária à

implementação de um filtro ou de sua versão adaptativa está intimamente relacionada com o número total de coeficientes desse filtro. Pode-se então notar, a partir de (2.5) e (2.6), que o bloco de segunda ordem de um filtro Volterra apresenta um coeficiente para cada um dos produtos cruzados de segunda ordem das amostras do sinal de entrada, resultando em uma quantidade de coeficientes bastante superior àquela requerida pelo bloco de primeira ordem. Para blocos de ordens superiores a situação se torna ainda mais crítica, pois será necessária uma quantidade de coeficientes proporcional às combinações de ordem  $p$  das amostras do sinal de entrada, com  $p$  representando a ordem do bloco em questão. Assim, algumas vezes o tamanho de memória e a ordem requeridos para um determinado problema podem dificultar ou mesmo inviabilizar o uso de um filtro Volterra. Esse número elevado de coeficientes dificulta também o uso de algoritmos adaptativos de maior complexidade, tais como o algoritmo RLS (*recursive least-squares*) [5] e o algoritmo de projeções afins (AP).

Agora, denotando por  $D_p(N)$  o número de coeficientes de cada bloco de ordem  $p$  de um filtro Volterra com tamanho de memória  $N$ , tem-se [12]

$$D_p(N) = \frac{(N+p-1)!}{(N-1)!p!}. \quad (2.12)$$

Conseqüentemente, o número total de coeficientes de um filtro Volterra de ordem  $P$  e tamanho de memória  $N$  é dado pelo somatório dos coeficientes de cada um dos blocos. Assim, denotando-se tal número por  $D_v(N, P)$ , temos

$$D_v(N, P) = \sum_{p=1}^P D_p(N), \quad (2.13)$$

o que resulta em

$$D_v(N, P) = \frac{(N+P)!}{N!P!} - 1. \quad (2.14)$$

## 2.4 Considerações

Neste capítulo, as principais características do filtro Volterra foram apresentadas. Nos próximos capítulos essas características serão exploradas no desenvolvimento de modelos matemáticos para o comportamento do filtro Volterra adaptativo e de algoritmos de complexidade reduzida.

## Análise Estatística do Filtro Volterra Adaptado pelo Algoritmo LMS

Na literatura, são encontrados diversos trabalhos dedicados à análise do filtro Volterra adaptativo. A maioria deles é orientada a problemas de convergência e estabilidade do algoritmo [20] [21], sendo de grande interesse para avaliar seu desempenho e suas limitações. Outros trabalhos são focados especificamente em filtros que empregam procedimentos de ortogonalização do sinal de entrada [22] ou mesmo filtros com restrição de ordem [20]. Entretanto, nenhum deles apresenta uma análise dos momentos de primeira e segunda ordens para os filtros Volterra adaptados pelo algoritmo LMS.

Neste capítulo, modelos analíticos para os momentos de primeira e segunda ordens do vetor de coeficientes do filtro Volterra adaptado pelo algoritmo LMS são desenvolvidos. Resultados de simulação são apresentados, atestando a validade e a precisão das expressões analíticas obtidas.

### 3.1 Vetor de Coeficientes Ótimo

A Figura 3.1 apresenta o diagrama em blocos de um filtro Volterra no contexto da estimação de um sinal  $d(n)$  estacionário, baseado na excitação  $x(n)$  também estacionária e correlacionada com  $d(n)$ . O sinal de saída do filtro é representado por  $y(n)$  e o sinal de erro por  $e(n)$ . Este esquema é semelhante ao da Figura 1.1.

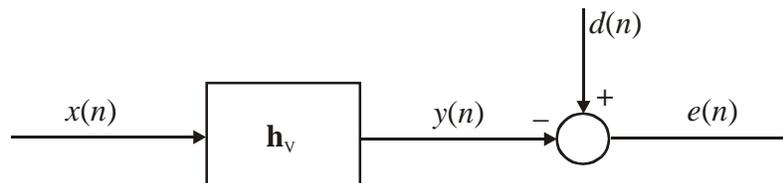


Figura 3.1 – Diagrama em blocos de um filtro Volterra aplicado a um problema de estimação de sinal.

A intenção é obter o vetor de coeficientes que minimize o erro quadrático médio (função custo). Assim, a partir da relação entrada/saída do filtro Volterra escrita na forma vetorial (2.11), temos o erro instantâneo dado por

$$e(n) = d(n) - y(n) = d(n) - \mathbf{h}_v^T \mathbf{x}_v(n). \quad (3.1)$$

Elevando-se (3.1) ao quadrado e tomando-se seu valor esperado, obtém-se

$$\xi = E[e^2(n)] = E[d^2(n)] - 2E[d(n)\mathbf{x}_v^T(n)]\mathbf{h}_v + \mathbf{h}_v^T E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]\mathbf{h}_v. \quad (3.2)$$

Em (3.2) o vetor  $\mathbf{h}_v$  foi retirado do operador esperança por não se tratar aqui de uma variável estatística. Como realizado na Seção 1.2.1, para se obter o vetor ótimo de coeficientes que minimize o erro quadrático médio, faz-se o gradiente da função custo em relação aos coeficientes igual a zero, obtendo-se

$$\nabla \xi = -2E[d(n)\mathbf{x}_v(n)] + 2E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]\mathbf{h}_v = 0. \quad (3.3)$$

Definindo-se a matriz  $\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]$  e o vetor  $\mathbf{p}_v = E[d(n)\mathbf{x}_v(n)]$ , é possível reescrever (3.3) como

$$-2\mathbf{p}_v + 2\mathbf{R}_{vv}\mathbf{h}_v = 0, \quad (3.4)$$

que resulta em uma expressão semelhante à de Wiener-Hopf para o filtro Volterra, dada por

$$\mathbf{R}_{vv}\mathbf{h}_{v_0} = \mathbf{p}_v, \quad (3.5)$$

com  $\mathbf{h}_{v_0}$  denotando o vetor de coeficientes ótimo. Resolvendo (3.5), obtém-se

$$\mathbf{h}_{v_0} = \mathbf{R}_{vv}^{-1}\mathbf{p}_v. \quad (3.6)$$

Esta solução é semelhante àquela obtida para filtros lineares mostrada na Seção 1.2.1. No entanto, é importante ressaltar uma principal diferença existente em tal representação. O vetor de entrada  $\mathbf{x}_v(n)$ , como pode ser observado em (2.9), é distinto daquele para o caso do filtro linear.

### 3.2 Comportamento Médio dos Pesos

Como mencionado anteriormente, a linearidade entre o sinal de saída e os coeficientes do filtro Volterra permite a aplicação direta do algoritmo LMS para efetuar a adaptação de seus pesos (coeficientes). Assim, a equação de atualização dos pesos pode ser expressa como

$$\mathbf{h}_v(n+1) = \mathbf{h}_v(n) + 2\mu e(n)\mathbf{x}_v(n). \quad (3.7)$$

A Figura 3.2 apresenta um diagrama em blocos semelhante ao da Figura 3.1, porém agora temos um filtro Volterra adaptativo aplicado a um problema de estimação de um sinal  $d(n)$  adicionado a um ruído branco  $z(n)$  de média zero e descorrelacionado dos demais sinais do sistema.

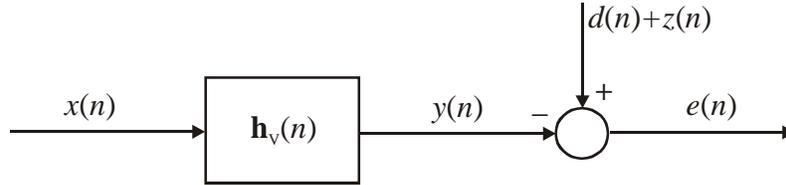


Figura 3.2 – Diagrama em blocos de um filtro Volterra adaptativo aplicado a um problema de estimação de sinal.

Com base na Figura 3.2, tem-se o sinal de erro instantâneo dado por

$$e(n) = d(n) + z(n) - y(n) = d(n) + z(n) - \mathbf{h}_v^T(n) \mathbf{x}_v(n). \quad (3.8)$$

Para obter o momento de primeira ordem dos coeficientes do filtro, substitui-se em (3.7) o valor instantâneo do erro dado por (3.8). A seguir, tomando-se o valor esperado, obtém-se

$$\begin{aligned} E[\mathbf{h}_v(n+1)] &= E[\mathbf{h}_v(n)] + 2\mu E[d(n) \mathbf{x}_v(n)] \\ &\quad + 2\mu E[z(n) \mathbf{x}_v(n)] - 2\mu E[\mathbf{x}_v(n) \mathbf{h}_v^T(n) \mathbf{x}_v(n)]. \end{aligned} \quad (3.9)$$

Como demonstrado em [5], podemos assumir que a correlação entre vetores de entrada é mais importante do que a correlação entre o vetor de coeficientes e o vetor de entrada. Essa aproximação, conhecida como hipótese da independência (*independence assumption*), parte do pressuposto que as amostras atuais dos sinais observados  $\mathbf{x}(n)$ ,  $d(n)$  são independentes de suas amostras anteriores  $\mathbf{x}(n-1)$ ,  $d(n-1)$ ,  $\mathbf{x}(n-2)$ ,  $d(n-2)$ . Dessa forma, é possível argumentar que  $\mathbf{h}_v(n)$  depende apenas das amostras anteriores, podendo ser considerado independente de  $\mathbf{x}_v(n)$  (amostra atual). Através de uma análise mais minuciosa nota-se que em muitos casos práticos a hipótese de independência é questionável. Porém, a experiência com o algoritmo LMS tem demonstrado que as previsões realizadas usando tal consideração se encaixam nos resultados de simulação do algoritmo LMS, principalmente quando pequenos valores de  $\mu$  são utilizados (condição de adaptação lenta). Então, admitindo-se essa fraca dependência entre  $\mathbf{x}_v(n)$  e  $\mathbf{h}_v(n)$  é possível escrever

$$E[\mathbf{x}_v(n) \mathbf{h}_v^T(n) \mathbf{x}_v(n)] \approx E[\mathbf{x}_v(n) \mathbf{x}_v^T(n)] E[\mathbf{h}_v(n)]. \quad (3.10)$$

Assumindo-se, ainda, o sinal de ruído com média zero e decorrelacionado dos demais sinais do sistema, substitui-se (3.10) em (3.9) para obter

$$E[\mathbf{h}_v(n+1)] = E[\mathbf{h}_v(n)] + 2\mu E[d(n)\mathbf{x}_v(n)] - 2\mu E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]. \quad (3.11)$$

Usando-se agora  $\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]$  e  $\mathbf{p}_v = E[d(n)\mathbf{x}_v(n)]$  em (3.11), podemos obter a expressão que descreve o comportamento médio dos pesos de um filtro Volterra adaptado pelo algoritmo LMS. Assim,

$$E[\mathbf{h}_v(n+1)] = (\mathbf{I} - 2\mu\mathbf{R}_{vv})E[\mathbf{h}_v(n)] + 2\mu\mathbf{p}_v, \quad (3.12)$$

onde  $\mathbf{I}$  é a matriz identidade.

### 3.3 Vetor de Erro dos Pesos

Definindo o vetor de erro dos pesos como a diferença entre o vetor de pesos e o vetor ótimo de pesos, temos

$$\mathbf{v}(n) = \mathbf{h}_v(n) - \mathbf{h}_{v_0}. \quad (3.13)$$

A substituição de (3.13) em (3.12) resulta em

$$E[\mathbf{v}(n+1)] + \mathbf{h}_{v_0} = (\mathbf{I} - 2\mu\mathbf{R}_{vv})E[\mathbf{v}(n)] + \mathbf{h}_{v_0} - 2\mu\mathbf{R}_{vv}\mathbf{h}_{v_0} + 2\mu\mathbf{p}_v. \quad (3.14)$$

Utilizando-se (3.5) (equação de Wiener-Hopf para o filtro Volterra) e manipulando (3.14), obtém-se a equação do comportamento do vetor de erro dos pesos, dada por

$$E[\mathbf{v}(n+1)] = (\mathbf{I} - 2\mu\mathbf{R}_{vv})E[\mathbf{v}(n)]. \quad (3.15)$$

### 3.4 Estrutura da Matriz de Autocorrelação de Entrada

Como mencionado anteriormente, a matriz de autocorrelação do sinal de entrada do filtro Volterra adaptativo é definida por

$$\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]. \quad (3.16)$$

A partir do vetor de entrada do filtro Volterra de ordem  $P$ , dado por (2.9), pode-se escrever  $\mathbf{R}_{vv}$  da seguinte forma:

$$\mathbf{R}_{vv} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1P} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{P1} & \mathbf{R}_{P2} & \cdots & \mathbf{R}_{PP} \end{bmatrix}, \quad (3.17)$$

onde

$$\mathbf{R}_{p_1 p_2} = \mathbf{R}_{p_2 p_1}^T = E[\mathbf{x}_{p_1}(n)\mathbf{x}_{p_2}^T(n)]. \quad (3.18)$$

Em (3.18),  $\mathbf{x}_{p_1}(n)$  e  $\mathbf{x}_{p_2}(n)$  representam vetores de entrada dos blocos de ordem  $p_1$  e  $p_2$ , respectivamente.

A matriz  $\mathbf{R}_{11}$  é correspondente à matriz de autocorrelação de entrada de um filtro FIR linear, pois se trata da matriz de autocorrelação de entrada de um bloco de primeira ordem do filtro Volterra. Tem-se então a matriz  $\mathbf{R}_{11}$  escrita como

$$\mathbf{R}_{11} = \begin{bmatrix} E[x(n)x(n)] & E[x(n)x(n-1)] & \cdots & E[x(n)x(n-N+1)] \\ E[x(n-1)x(n)] & E[x(n-1)x(n-1)] & \cdots & E[x(n-1)x(n-N+1)] \\ \vdots & \vdots & \ddots & \vdots \\ E[x(n-N+1)x(n)] & E[x(n-N+1)x(n-1)] & \cdots & E[x(n-N+1)x(n-N+1)] \end{bmatrix}. \quad (3.19)$$

Para um sinal de entrada branco Gaussiano com variância  $\sigma_x^2$  e média zero, a matriz  $\mathbf{R}_{11}$  é dada por

$$\mathbf{R}_{11} = \begin{bmatrix} \sigma_x^2 & 0 & \cdots & 0 \\ 0 & \sigma_x^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_x^2 \end{bmatrix}. \quad (3.20)$$

Analisando a expressão do comportamento médio dos pesos, podemos observar que, para filtros lineares com sinal de entrada branco, o comportamento de cada um dos pesos é independente do comportamento dos demais pesos. Isso pode ser constatado escrevendo-se o comportamento médio do  $i$ -ésimo peso do filtro a partir de (3.12) e (3.20). Assim,

$$E[h_i(n+1)] = (1 - 2\mu\sigma_x^2)E[h_i(n)] + 2\mu E[d(n)x(n-i+1)]. \quad (3.21)$$

Para um filtro Volterra de segunda ordem, ainda devem ser consideradas as matrizes  $\mathbf{R}_{12}$  e  $\mathbf{R}_{22}$  na composição total da matriz de autocorrelação do sinal de entrada. Obtendo-se essas matrizes para um sinal de entrada branco Gaussiano com variância  $\sigma_x^2$  e média zero, tem-se

$$\mathbf{R}_{12} = \mathbf{R}_{21}^T = \mathbf{0} \quad (3.22)$$

e

$$\mathbf{R}_{22} = \begin{bmatrix} 3\sigma_x^2 & 0 & \cdots & 0 & \sigma_x^2 & 0 & \cdots & \sigma_x^2 \\ 0 & \sigma_x^2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_x^2 & 0 & 0 & \cdots & 0 \\ \sigma_x^2 & 0 & \cdots & 0 & 3\sigma_x^2 & 0 & \cdots & \sigma_x^2 \\ 0 & 0 & \cdots & 0 & 0 & \sigma_x^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_x^2 & 0 & \cdots & 0 & \sigma_x^2 & 0 & \cdots & 3\sigma_x^2 \end{bmatrix}. \quad (3.23)$$

Como exemplo, usando-se (3.20), (3.22) e (3.23), tem-se a matriz de autocorrelação do sinal de entrada de um filtro Volterra de segunda ordem com tamanho de memória igual 3 e ruído branco gaussiano de entrada. Assim,

$$\mathbf{R}_{VV} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\sigma_x^2 & 0 & 0 & \sigma_x^2 & 0 & \sigma_x^2 \\ 0 & 0 & 0 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_x^2 & 0 & 0 & 3\sigma_x^2 & 0 & \sigma_x^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_x^2 & 0 \\ 0 & 0 & 0 & \sigma_x^2 & 0 & 0 & \sigma_x^2 & 0 & 3\sigma_x^2 \end{bmatrix}. \quad (3.24)$$

Portanto, podemos verificar que existem elementos também fora da diagonal principal da matriz de autocorrelação do sinal de entrada, mesmo considerando-se um sinal de entrada branco Gaussiano. Isso implica um comportamento não independente dos pesos, existindo, assim, um acoplamento entre eles. Outra consequência desse fato é um espalhamento maior dos autovalores desta matriz, degradando sobremaneira a velocidade de convergência do algoritmo LMS [5]. Ainda, podemos observar que, como a matriz  $\mathbf{R}_{12}$  é nula, existe um desacoplamento total entre o bloco de primeira e o de segunda ordem. Assim, podemos encontrar limites de estabilidade independentes, levando à escolha de diferentes passos de adaptação para cada bloco, uma vez que tais parâmetros dependem da matriz de autocorrelação do sinal de entrada [5].

Analisando-se filtros Volterra de ordens superiores, podemos observar que a condição de independência entre blocos de diferentes ordens, para sinal branco Gaussiano

de entrada, não mais ocorre. Como exemplo, um filtro Volterra adaptativo de terceira ordem com tamanho de memória igual a 3 e sinal de entrada branco Gaussiano, com variância  $\sigma_x^2$ , apresenta as seguintes matrizes adicionais na construção da matriz de autocorrelação de entrada:

$$\mathbf{R}_{13} = \begin{bmatrix} 3\sigma_x^2 & 0 & 0 & \sigma_x^2 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 3\sigma_x^2 & 0 & \sigma_x^2 & 0 \\ 0 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & \sigma_x^2 & 0 & 3\sigma_x^2 \end{bmatrix}, \quad (3.25)$$

$$\mathbf{R}_{23} = \mathbf{R}_{32}^T = \mathbf{0}, \quad (3.26)$$

e

$$\mathbf{R}_{33} = \begin{bmatrix} 15\sigma_x^2 & 0 & 0 & 3\sigma_x^2 & 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 & 3\sigma_x^2 & 0 & \sigma_x^2 & 0 \\ 0 & 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 & \sigma_x^2 & 0 & 3\sigma_x^2 \\ 3\sigma_x^2 & 0 & 0 & 3\sigma_x^2 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 3\sigma_x^2 & 0 & 0 & \sigma_x^2 & 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 & 15\sigma_x^2 & 0 & 3\sigma_x^2 & 0 \\ 0 & 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 3\sigma_x^2 & 0 & 3\sigma_x^2 \\ 0 & \sigma_x^2 & 0 & 0 & 0 & 0 & 3\sigma_x^2 & 0 & 3\sigma_x^2 & 0 \\ 0 & 0 & 3\sigma_x^2 & 0 & 0 & 0 & 0 & 3\sigma_x^2 & 0 & 15\sigma_x^2 \end{bmatrix}, \quad (3.27)$$

o que resulta em um acoplamento entre os blocos de primeira e terceira ordens em tal filtro.

### 3.5 Erro Quadrático Médio

Para determinar uma expressão analítica para a curva de aprendizagem do algoritmo, deve-se inicialmente substituir a definição do vetor de erro dos pesos (3.13) em (3.8), obtendo-se

$$e(n) = d(n) + z(n) - \mathbf{v}^T(n)\mathbf{x}_V(n) - \mathbf{h}_{V_0}^T \mathbf{x}_V(n). \quad (3.28)$$

Definindo o sinal de erro instantâneo ótimo por

$$e_o(n) = d(n) + z(n) - \mathbf{h}_{V_0}^T \mathbf{x}_V(n) \quad (3.29)$$

pode-se reescrever (3.28) como

$$e(n) = e_o(n) - \mathbf{v}^T(n)\mathbf{x}_V(n). \quad (3.30)$$

Elevando-se (3.30) ao quadrado e tomando seu valor esperado, obtém-se

$$E[e^2(n)] = E[e_o^2(n)] - E[e_o(n)\mathbf{v}^T(n)\mathbf{x}_v(n)] + E\{[\mathbf{v}^T(n)\mathbf{x}_v(n)]^2\}. \quad (3.31)$$

De [5], tem-se o princípio da ortogonalidade que estabelece que o erro de estimação ótimo é decorrelacionado do sinal de entrada, ou seja,  $E[\mathbf{x}_v(n)e_o(n)] = 0$ . Uma breve demonstração desse princípio é apresentada no Apêndice A. Usando-se tal princípio em conjunto com a hipótese da independência, dada por (3.10) e válida também para o vetor de erro dos pesos, tem-se

$$E[e_o(n)\mathbf{v}^T(n)\mathbf{x}_v(n)] = E[\mathbf{v}^T(n)]E[e_o(n)\mathbf{x}_v(n)] = 0. \quad (3.32)$$

Novamente utilizando-se a hipótese da independência (3.10), é possível escrever o último termo de (3.31) como

$$\begin{aligned} E\{[\mathbf{v}^T(n)\mathbf{x}_v(n)]^2\} &= E[\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)] \\ &= E\left\{\mathbf{v}^T(n)E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]\mathbf{v}(n)\right\} \\ &= E[\mathbf{v}^T(n)\mathbf{R}_{vv}\mathbf{v}(n)]. \end{aligned} \quad (3.33)$$

Dado que  $E[\mathbf{v}^T(n)\mathbf{R}_{vv}\mathbf{v}(n)]$  é dimensionalmente um escalar, o resultado de (3.33) pode ser reescrito por

$$\begin{aligned} E[\mathbf{v}^T(n)\mathbf{R}_{vv}\mathbf{v}(n)] &= \text{tr}\left\{E[\mathbf{v}^T(n)\mathbf{R}_{vv}\mathbf{v}(n)]\right\} \\ &= E\left\{\text{tr}[\mathbf{v}^T(n)\mathbf{R}_{vv}\mathbf{v}(n)]\right\} \\ &= E\left\{\text{tr}[\mathbf{v}^T(n)\mathbf{v}(n)\mathbf{R}_{vv}]\right\} \\ &= \text{tr}\left\{E[\mathbf{v}^T(n)\mathbf{v}(n)]\mathbf{R}_{vv}\right\}, \end{aligned} \quad (3.34)$$

onde  $\text{tr}\{\cdot\}$  representa o operador traço da matriz. Denotando-se agora o erro quadrático médio por  $\xi(n) = E[e^2(n)]$ , o seu valor mínimo por  $\xi_{\min} = E[e_o^2(n)]$ , e definindo-se a matriz de autocorrelação do vetor de erro dos pesos por  $\mathbf{K}(n) = E[\mathbf{v}(n)\mathbf{v}^T(n)]$ , a expressão para a curva de aprendizagem é reescrita como

$$\xi(n) = \xi_{\min} + \text{tr}\left\{\mathbf{K}(n)\mathbf{R}_{vv}\right\}. \quad (3.35)$$

### 3.6 Aproximação para o Momento de Segunda Ordem

Para possibilitar a obtenção da curva de aprendizagem, uma expressão recursiva para o momento de segunda ordem dos pesos [matriz  $\mathbf{K}(n)$ ] é requerida. Para tal, através da substituição de (3.13) em (3.7), obtém-se

$$\mathbf{v}(n+1) = \mathbf{v}(n) + 2\mu e(n)\mathbf{x}_v(n). \quad (3.36)$$

Substituindo-se agora (3.30) em (3.36), tem-se

$$\mathbf{v}(n+1) = \mathbf{v}(n) + 2\mu\mathbf{x}_v(n)e_o(n) - 2\mu\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n). \quad (3.37)$$

Multiplicando-se o vetor  $\mathbf{v}(n+1)$  por sua versão transposta e tomando-se o seu valor esperado, obtém-se

$$\begin{aligned} E[\mathbf{v}(n+1)\mathbf{v}^T(n+1)] = & \\ & E[\mathbf{v}(n)\mathbf{v}^T(n)] \\ & + 2\mu E[\mathbf{v}(n)e_o(n)\mathbf{x}_v^T(n)] \\ & - 2\mu E[\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)] \\ & + 2\mu E[e_o(n)\mathbf{x}_v(n)\mathbf{v}^T(n)] \\ & + 4\mu^2 E[\mathbf{x}_v(n)e_o(n)e_o(n)\mathbf{x}_v^T(n)] \\ & - 4\mu^2 E[e_o(n)\mathbf{x}_v(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)] \\ & - 2\mu E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)\mathbf{v}^T(n)] \\ & - 4\mu^2 E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)e_o(n)\mathbf{x}_v^T(n)] \\ & + 4\mu^2 E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)]. \end{aligned} \quad (3.38)$$

A resolução de (3.38) consiste em um trabalho matemático considerável, especialmente devido à existência de elementos dependentes de momentos de ordem elevada do vetor de entrada. No entanto, uma aproximação satisfatória pode ser obtida desconsiderando os termos dependentes de  $\mu^2$ , visto que para uma condição de adaptação lenta comum em casos práticos, tal valor torna-se muito pequeno. Dessa forma, preservam-se os elementos dominantes em (3.38) como também facilita-se o tratamento matemático. Levando-se em conta a hipótese da independência e o princípio da ortogonalidade, temos os termos de (3.38) não dependentes de  $\mu^2$  dados por

$$E[\mathbf{v}(n)\mathbf{v}^T(n)] = \mathbf{K}(n), \quad (3.39)$$

$$2\mu E[\mathbf{v}(n)e_o(n)\mathbf{x}_v^T(n)] = 2\mu E[\mathbf{v}(n)]E[e_o(n)\mathbf{x}_v^T(n)] = 0, \quad (3.40)$$

$$-2\mu E[\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)] = -2\mu\mathbf{K}(n)\mathbf{R}_{vv}, \quad (3.41)$$

$$2\mu E[e_o(n)\mathbf{x}_v(n)\mathbf{v}^T(n)] = 0, \quad (3.42)$$

$$-2\mu E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)\mathbf{v}^T(n)] = -2\mu\mathbf{R}_{vv}\mathbf{K}(n), \quad (3.43)$$

Com isso, obtém-se uma aproximação para expressão recursiva do momento de segunda ordem, dada por

$$\mathbf{K}(n+1) = \mathbf{K}(n) - 2\mu(\mathbf{K}(n)\mathbf{R}_{VV} + \mathbf{R}_{VV}\mathbf{K}(n)). \quad (3.44)$$

Utilizando-se (3.44) em conjunto com (3.35), pode-se obter o comportamento aproximado do erro quadrático médio do filtro Volterra adaptado pelo algoritmo LMS.

### 3.7 Simulações

Para verificar a precisão das expressões obtidas, alguns resultados de simulação são apresentados. Os exemplos desenvolvidos consideram um problema de identificação de sistemas usando sinais de entrada Gaussianos branco e colorido. O ruído de medição  $z(n)$  usado é descorrelacionado dos demais sinais do sistema, tendo uma variância  $\sigma_z^2 = 0,001$ . Os resultados são apresentados para diferentes valores do passo de adaptação, todos relacionados ao valor máximo de  $\mu$  ( $\mu_{\max}$ ) determinado experimentalmente, para o qual o algoritmo ainda converge. Isso permite verificar as aproximações efetuadas, levando-se em conta diferentes velocidades de convergência.

*Exemplo 3.1:* Neste exemplo, um filtro Volterra de segunda ordem e tamanho de memória igual a 7 é usado para identificar uma planta com a seguinte relação entrada/saída:

$$\begin{aligned} d(n) = & x(n) + 0,5x(n-1) + 0,1x(n-2) - 0,2x(n-3) - 0,3x(n-4) \\ & - 0,1x(n-5) + 0,08x(n-6) + 0,7x^2(n) + 0,3x(n)x(n-1) \\ & - 0,1x(n)x(n-3) + 0,05x(n)x(n-6) + 0,5x^2(n-1) \\ & + 0,2x(n-1)x(n-2) - 0,2x(n-1)x(n-5) - 0,35x^2(n-2) \\ & - 0,05x(n-2)x(n-4) + 0,02x(n-2)x(n-5) - 0,2x^2(n-3) \\ & + 0,05x(n-3)x(n-5) + 0,1x^2(n-4) + 0,05x(n-4)x(n-5) \\ & + 0,02x(n-4)x(n-6) - 0,1x^2(n-5) + 0,05x^2(n-6). \end{aligned}$$

O sinal de entrada é branco Gaussiano e com variância unitária. O passo de adaptação é igual a  $0,5\mu_{\max}$ , com  $\mu_{\max} = 0,005$ . A Figura 3.3 apresenta o comportamento médio de alguns dos coeficientes obtido por simulação Monte Carlo (média de 100 realizações) comparado com o resultado obtido por (3.12). Através desta figura, podemos constatar a muito boa precisão do modelo obtido, exceto por uma pequena diferença existente na fase

transitória. Tal diferença é devido à aproximação considerada em (3.10) (hipótese da independência). A Figura 3.4 apresenta a curva de aprendizagem obtida com (3.35) e (3.44) comparada com a obtida por simulação (média de 100 realizações). Neste caso, uma maior discrepância é verificada devido à aproximação usada na obtenção do momento de segunda ordem. Com a redução do valor do passo de adaptação (condição de adaptação lenta) ambas as diferenças diminuem, como será observado nos próximos exemplos.

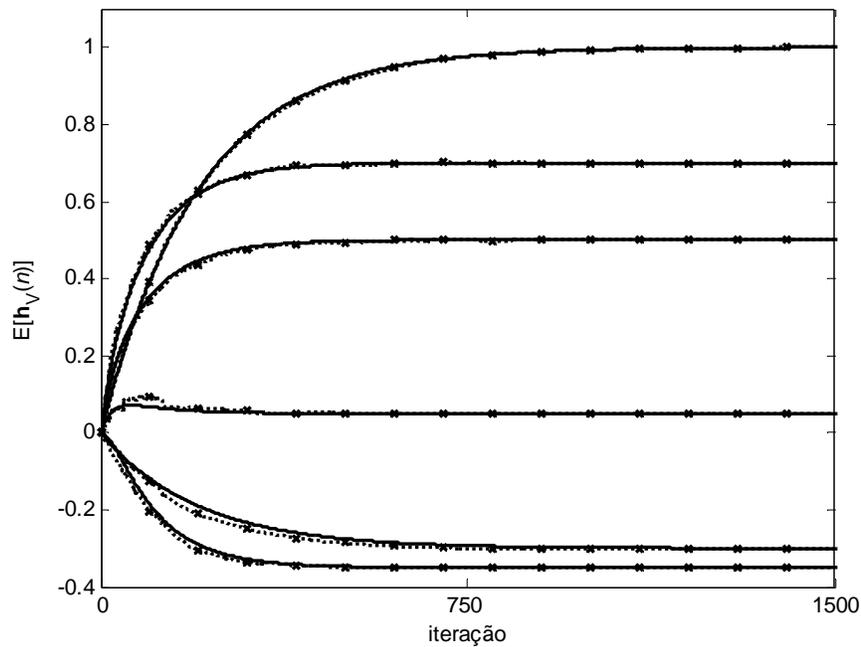


Figura 3.3. Exemplo 3.1. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,5\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (Linha sólida) modelo dado por (3.12).

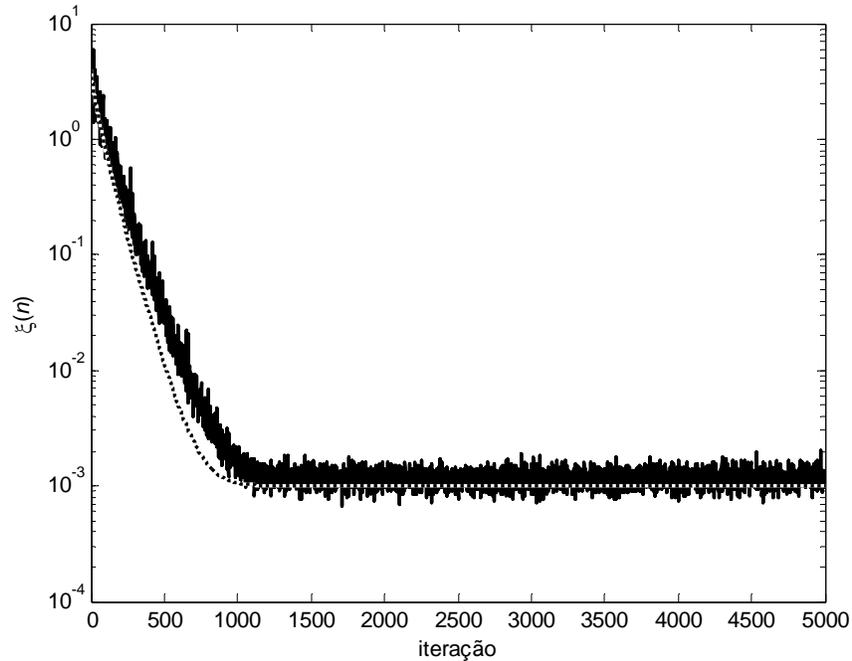


Figura 3.4. Exemplo 3.1. Evolução de  $\xi(n)$  com  $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44).

*Exemplo 3.2:* O mesmo filtro do Exemplo 3.1 é considerado, modificando-se apenas o valor do passo de adaptação para  $0,2\mu_{\max}$ . A Figura 3.5 mostra os resultados de maneira similar aos da Figura 3.3, enquanto a Figura 3.6 ilustra o comportamento do erro quadrático médio seguindo o mesmo padrão da Figura 3.4. Neste exemplo, podemos constatar a excelente precisão da modelagem desenvolvida, o que pode ser verificado através da similaridade entre as curvas obtidas para uma condição de adaptação lenta.

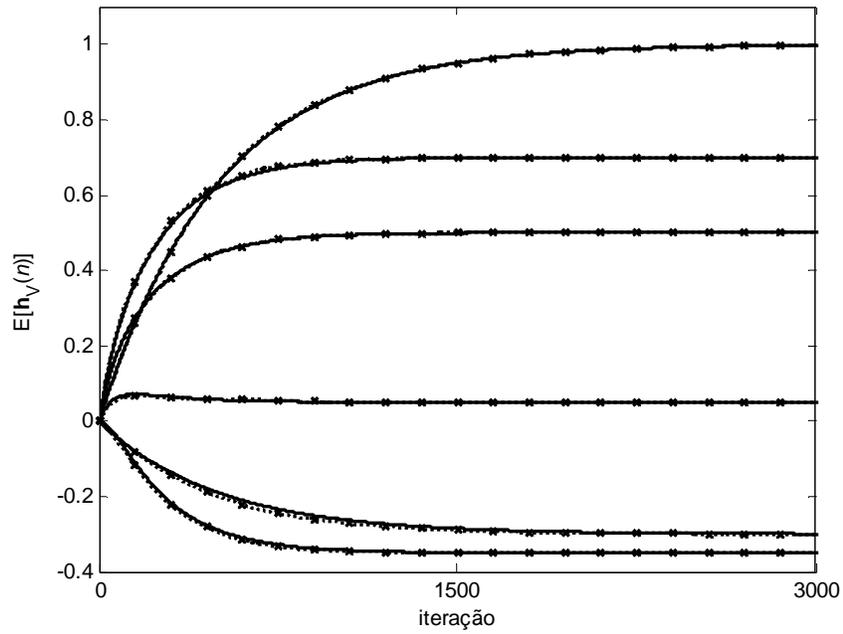


Figura 3.5. Exemplo 3.2. Evolução de  $E[\mathbf{h}_v(n)]$ , com  $\mu = 0,2\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (linha sólida) modelo dado por (3.12).

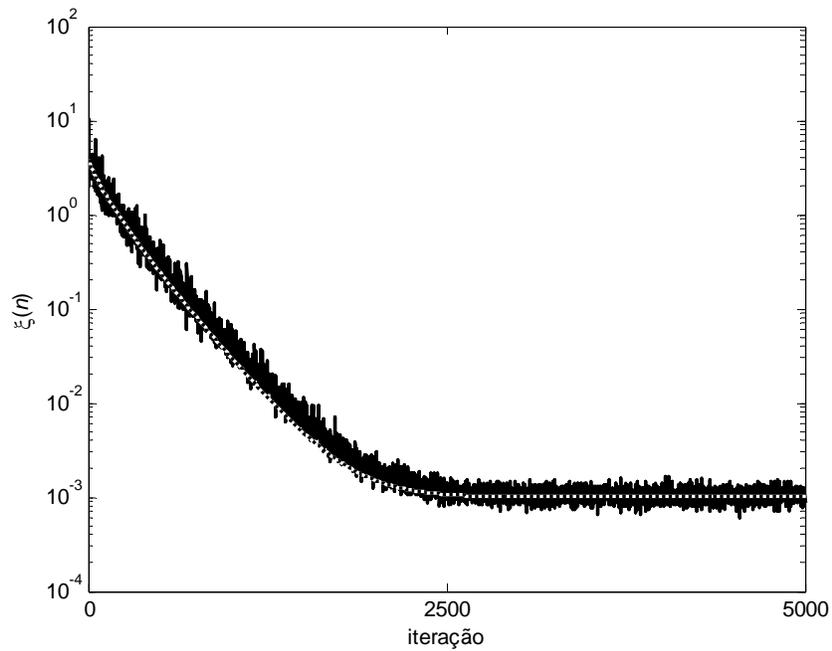


Figura 3.6. Exemplo 3.2. Evolução de  $\xi(n)$ , com  $\mu = 0,2\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44).

*Exemplo 3.3:* Novamente, o filtro do Exemplo 3.1 é considerado. Agora, o sinal de entrada é um sinal colorido obtido a partir de um processo auto-regressivo dado por  $x(n) = \alpha x(n-1) + \sqrt{1-\alpha^2} u(n)$ , onde  $u(n)$  é um ruído branco Gaussiano com variância unitária e  $\alpha = 0,5$ . O passo de adaptação é  $0,5\mu_{\max}$  com  $\mu_{\max} = 0,0015$ . Os resultados do comportamento médio dos coeficientes são mostrados na Figura 3.7, enquanto a curva de aprendizagem é ilustrada na Figura 3.8. Novamente, ótimos resultados são obtidos, comprovando tanto a validade das expressões analíticas desenvolvidas quanto as aproximações consideradas.

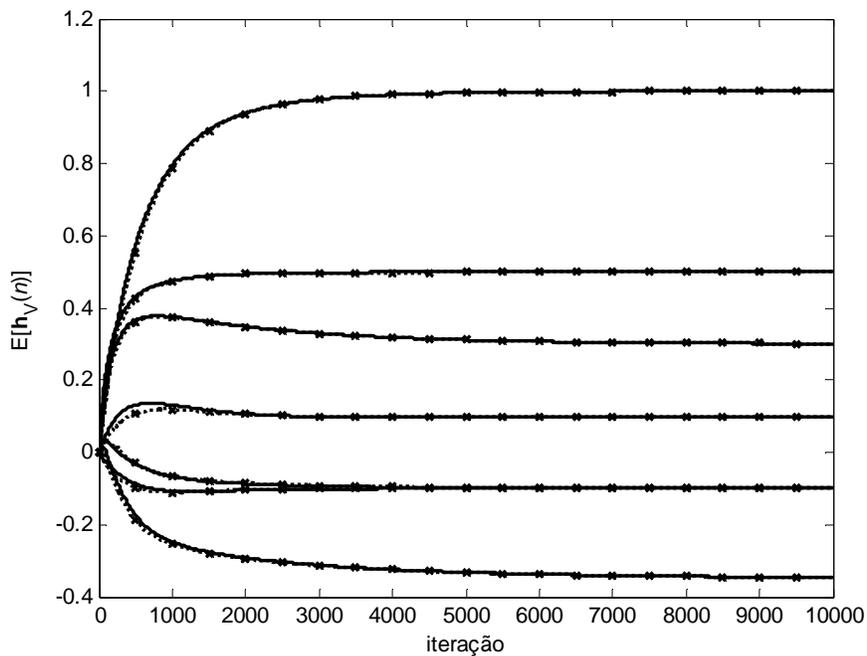


Figura 3.7. Exemplo 3.3. Evolução de  $E[\mathbf{h}_v(n)]$ , com  $\mu = 0,5\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (Linha sólida) modelo dado por (3.12).

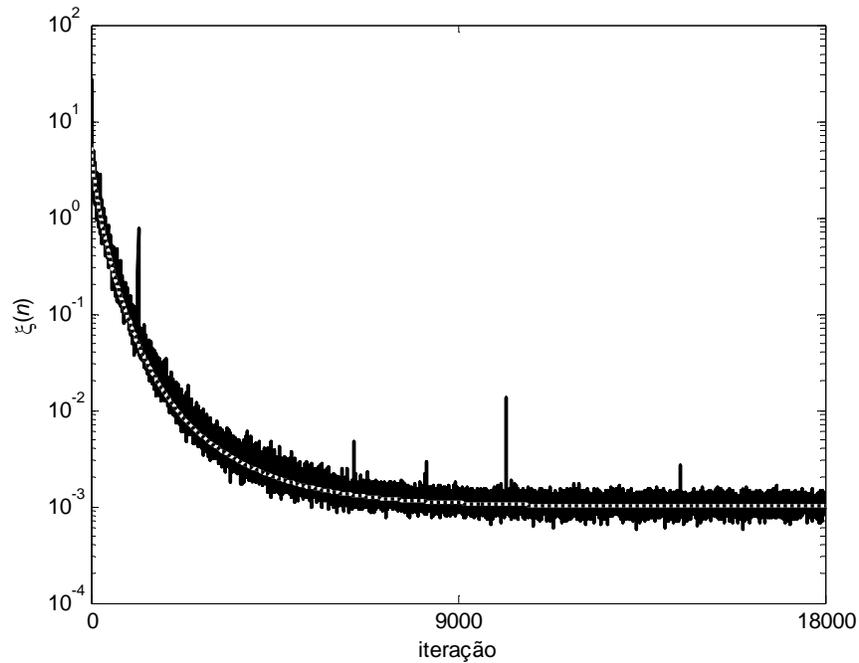


Figura 3.8. Exemplo 3.3. Evolução de  $\xi(n)$ , com  $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44).

*Exemplo 3.4:* Neste exemplo, uma planta Volterra de terceira ordem e tamanho de memória igual a 4 é considerada, possuindo os seguintes vetores de coeficientes:

$$\mathbf{h}_1 = [0,75 \quad 0,3 \quad -0,1 \quad -0,3]^T;$$

$$\mathbf{h}_2 = [-0,37 \quad -0,33 \quad -0,28 \quad -0,23 \quad 0,04 \quad 0,07 \quad 0,09 \quad 0,13 \quad 0,07 \quad 0,06]^T;$$

$$\mathbf{h}_3 = [0,27 \quad -0,16 \quad 0,06 \quad -0,01 \quad 0,22 \quad 0,14 \quad -0,02 \quad -0,25 \quad 0,17 \quad -0,03 \quad 0,2 \quad 0,11 \quad -0,15 \quad -0,04 \quad 0,2 \quad 0,19 \quad -0,04 \quad 0,17 \quad -0,19 \quad -0,06]^T.$$

Esta planta é identificada usando um filtro Volterra adaptativo também de terceira ordem e com tamanho de memória igual a 4. O valor do passo de adaptação é  $0,25\mu_{\max}$  com  $\mu_{\max} = 0,001$ , e o sinal de entrada é branco Gaussiano com variância unitária. As Figuras 3.9 e 3.10 mostram os resultados de simulação obtidos usando o método de Monte Carlo (média de 100 realizações) e através das expressões analíticas desenvolvidas. Novamente, observa-se uma concordância muito boa entre os resultados de simulação e os correspondentes aos modelos.

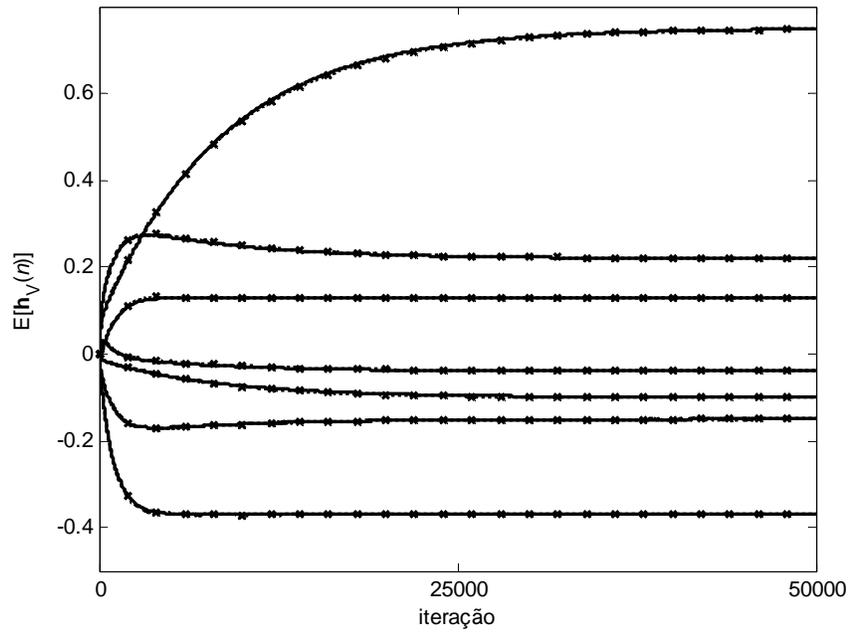


Figura 3.9. Exemplo 3.4. Evolução de  $E[\mathbf{h}_v(n)]$ , com  $\mu = 0,25\mu_{\max}$ . (Linha pontilhada com marcadores) simulação (média de 100 realizações); (Linha sólida) modelo dado por (3.12).

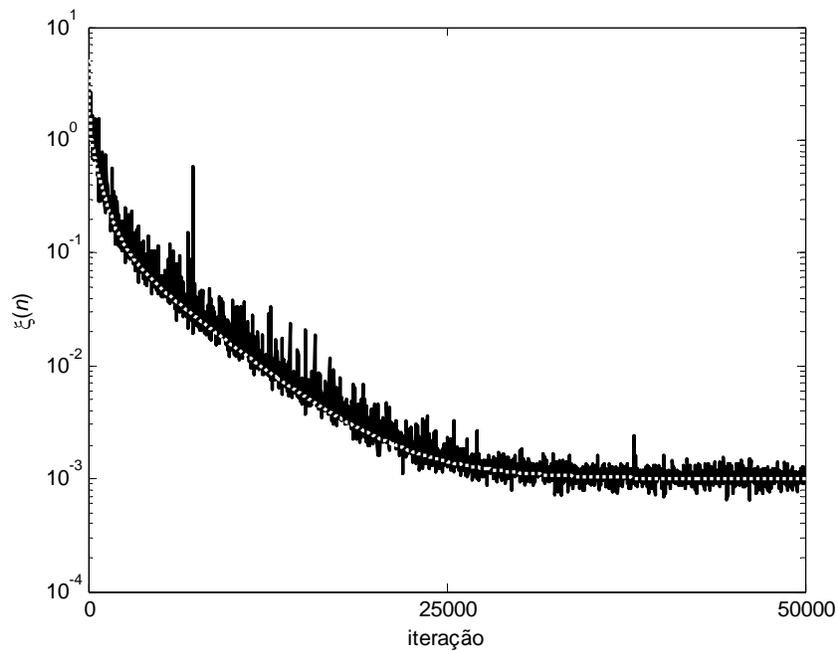


Figura 3.10. Exemplo 3.4. Evolução de  $\xi(n)$ , com  $\mu = 0,25\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada) modelo dado por (3.35) e (3.44).

### **3.8 Considerações**

Neste Capítulo, uma análise estatística do filtro Volterra adaptado pelo algoritmo LMS foi apresentada. O modelo pseudolinear da operação de filtragem Volterra é explorado, permitindo a derivação de expressões analíticas para o seu comportamento. Através dos resultados de simulação, foi possível comprovar a validade da modelagem analítica proposta. O resultado obtido neste capítulo deu origem a uma publicação [23].

## Filtros Volterra Adaptativos Interpolado e Parcialmente Interpolado

Apesar do sucesso obtido com o uso de filtros Volterra adaptativos em diversas aplicações, a alta complexidade computacional requerida, mesmo quando tamanhos de memória relativamente pequenos são considerados, ainda é um fator que impede a expansão de sua utilização. Como comentado na Seção 2.3, a complexidade computacional necessária para implementação de um filtro Volterra adaptativo cresce exponencialmente com o aumento do tamanho da memória do filtro, restringindo severamente sua aplicação uma vez que os recursos computacionais disponíveis são na maioria das vezes limitados. Algumas alternativas têm sido propostas para sobrepujar tal problema. Como exemplos de implementação que apresentam complexidade reduzida, tem-se: filtros Volterra no domínio da frequência [24], filtros Volterra com atraso [14], e filtros Volterra simplificados [15].

Neste capítulo, duas estruturas para implementação subótima do filtro Volterra adaptativo são discutidas. A idéia é utilizar um filtro Volterra esparso adaptativo em conjunto com um interpolador de tal forma que esse conjunto se comporte como um filtro adaptativo completo. Essa abordagem tem sido aplicada com sucesso a filtros lineares [25], [26], [27], [28], [29]. Aqui, tal princípio é usado tanto para o filtro em sua totalidade quanto apenas para os blocos de ordem superior, considerando que esses últimos são os principais responsáveis pelo aumento exponencial no número global de coeficientes. Essas estruturas são denominadas filtro Volterra adaptativo interpolado e filtro Volterra adaptativo parcialmente interpolado, respectivamente.

### 4.1 Filtro Volterra Adaptativo Interpolado

A partir da Seção 2.3, nota-se que com a redução do tamanho da memória de um filtro Volterra, podemos obter uma diminuição substancial no seu número de coeficientes. Uma maneira de fazer essa redução é utilizando filtros esparsos. Como em [28], podemos

denotar o fator de interpolação por  $L$  e considerar que o vetor de entrada esparsa de primeira ordem  $\mathbf{x}_{1s}(n)$  é obtido removendo  $(L-1)$  amostras de cada  $L$  amostras consecutivas do sinal de entrada original. Assim temos

$$\mathbf{x}_{1s}(n) = \{x(n) \ x(n-L) \ x(n-2L) \ \cdots \ x[n-(N_s-1)L]\}^T \quad (4.1)$$

com  $N_s$  denotando o tamanho de memória do filtro esparsa, dado por

$$N_s = \left\lfloor \frac{N-1}{L} \right\rfloor + 1, \quad (4.2)$$

onde  $\lfloor \cdot \rfloor$  denota a operação de truncamento. O vetor de coeficientes esparsa de primeira ordem correspondente fica assim dado por

$$\mathbf{h}_{1s} = \{h_1(0) \ h_1(L) \ h_1(2L) \ \cdots \ h_1[(N_s-1)L]\}^T. \quad (4.3)$$

Os demais vetores de entrada e de coeficientes de ordem superior são obtidos a partir dos de primeira ordem como apresentado na Seção 2.1.

Entretando, usando um filtro esparsa estamos considerando uma estrutura sub-ótima com significante perda de desempenho. Um filtro interpolador de entrada pode então ser utilizado para reduzir o efeito dos coeficientes removidos. Esse procedimento leva ao filtro Volterra interpolado, demonstrado na Figura 4.1. Nesta figura,  $\mathbf{h}_{vi}$  representa o filtro Volterra adaptativo esparsa, enquanto o interpolador é representado por um filtro FIR  $\mathbf{I} = [i_0 \ i_1 \ \cdots \ i_{M-1}]^T$ , com  $M$  coeficientes. O sinal de entrada e o sinal de entrada interpolado são denotados por  $x(n)$  e  $x_i(n)$ , respectivamente, onde

$$x_i(n) = \sum_{j=0}^{M-1} i_j x(n-j). \quad (4.4)$$

O sinal a ser estimado  $d(n)$  é somado a um ruído de medição  $z(n)$  de média zero e descorrelacionado dos demais sinais do sistema. Os sinais  $y(n)$  e  $e(n)$  representam o sinal de saída do filtro esparsa e o sinal de erro, respectivamente. Assim, o vetor de entrada de primeira ordem para o filtro Volterra interpolado fica dado por

$$\mathbf{x}_{1i}(n) = \{x_i(n) \ x_i(n-L) \ x_i(n-2L) \ \cdots \ x_i[n-(N_s-1)L]\}^T. \quad (4.5)$$

Gerando-se os vetores de entrada de ordens superiores a partir de (4.5), o vetor de entrada completo, denotado por  $\mathbf{x}_{vi}(n)$ , é construído de maneira similar à (2.9).

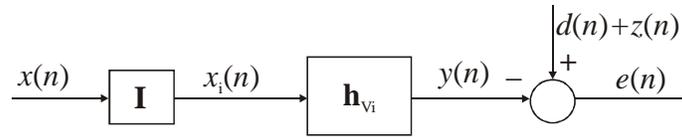


Figura 4.1 – Diagrama de blocos do filtro Volterra interpolado.

A estrutura total do filtro Volterra interpolado pode ser entendida como um filtro Volterra completo com restrições nos coeficientes. Considerando esse fato no contexto adaptativo, temos um filtro Volterra com restrições necessitando apenas realizar a adaptação dos coeficientes do filtro esparsos, o que é muito interessante dada a alta complexidade computacional requerida para adaptação de um filtro Volterra. O algoritmo adaptativo efetua então o ajuste dos coeficientes na direção dos seus valores ótimos. Assim, a equação de adaptação dos coeficientes para o filtro Volterra adaptado pelo algoritmo LMS é dada por

$$\mathbf{h}_{v_i}(n+1) = \mathbf{h}_{v_i}(n) + 2\mu e(n)\mathbf{x}_{v_i}(n), \quad (4.6)$$

onde  $\mathbf{h}_{v_i}(n)$  é o vetor de coeficientes correspondente ao vetor de entrada interpolado  $\mathbf{x}_{v_i}(n)$ . A Tabela 4.1 compara o número de coeficientes necessário tanto para o filtro Volterra adaptativo convencional quanto para sua versão interpolada, destacando a diferença de complexidade entre as estruturas consideradas.

Tabela 4.1 – Complexidade: Filtro Volterra adaptativo comparado com sua versão interpolada

$N$	$P$	$L$	$N_s$	$D_v(N, P)$	$D_v(N_s, P)$	Redução (%)
3	2	2	2	9	5	44,44%
10	2	2	5	65	20	69,23%
25	2	2	13	350	104	70,29%
15	3	2	8	815	164	79,88%
30	3	2	15	5455	815	85,06%
30	3	4	8	5455	164	96,99%

Entretanto, é bem conhecido que filtros interpolados não apresentam bom desempenho para a modelagem de plantas com baixo grau de similaridade entre coeficientes [29]. Para tal caso, uma solução mais apropriada pode ser o uso de um filtro Volterra adaptativo parcialmente interpolado, descrito na próxima seção.

## 4.2 Filtro Volterra Adaptativo Parcialmente Interpolado

Uma segunda versão simplificada do filtro Volterra adaptativo é obtida através do uso da interpolação apenas nos blocos de maior ordem. Fazendo-se isso, é possível obter uma redução de complexidade bem próxima à do filtro interpolado, uma vez que esses blocos são os que possuem o maior número de coeficientes. A Figura 4.2 mostra a estrutura de um filtro Volterra adaptativo parcialmente interpolado de segunda ordem. Nesta figura,  $\mathbf{h}_1$  e  $\mathbf{h}_{2i}$  representam o bloco de primeira ordem e o bloco esparso de segunda ordem, respectivamente, com  $y_1(n)$  e  $y_2(n)$  representando seus sinais de saída. Os demais sinais da Figura 4.2 são equivalentes aos da Figura 4.1.

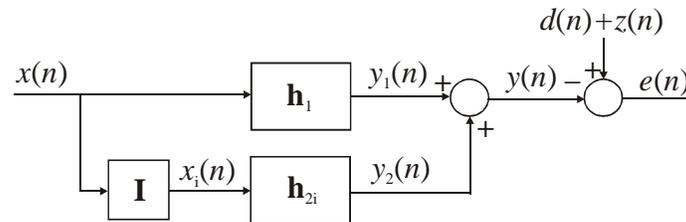


Figura 4.2 – Diagrama de blocos de um filtro Volterra adaptativo parcialmente interpolado de segunda ordem.

Para a estrutura da Figura 4.2, os coeficientes são atualizados da seguinte forma:

$$\begin{aligned} \mathbf{h}_1(n+1) &= \mathbf{h}_1(n) + 2\mu_1 e(n) \mathbf{x}_1(n) \\ \mathbf{h}_{2i}(n+1) &= \mathbf{h}_{2i}(n) + 2\mu_2 e(n) \mathbf{x}_{2i}(n), \end{aligned} \quad (4.7)$$

onde  $\mathbf{x}_1(n)$  é o vetor de entrada de primeira ordem; e  $\mathbf{x}_{2i}(n)$  é o vetor de entrada esparso de segunda ordem, obtido a partir do vetor  $\mathbf{x}_{1i}(n)$  (4.5). Os passos de atualização do algoritmo adaptativo para o bloco de primeira ordem e para o bloco esparso de segunda ordem são dados por  $\mu_1$  e  $\mu_2$ , respectivamente.

O número total de coeficientes para este filtro é determinado somando o número de coeficientes para cada um dos blocos, obtido através de (2.12). É importante lembrar que o tamanho de memória dos blocos interpolados é reduzido de acordo com (4.2). Na Tabela 4.2, as diferentes estruturas são comparadas em número requerido de coeficientes. Podemos observar, a partir desta tabela, que o número de coeficientes obtido com a abordagem parcialmente interpolada é bem próximo daquele alcançado com uma estrutura empregando interpolação total.

Tabela 4.2 – Número de coeficientes para as diferentes estruturas com  $L = 2$ 

Memória	Ordem	Número de coeficientes		
		Volterra	Interpolado	Parc. Interpolado
3	2	9	5	6
10	2	65	20	25
25	2	350	104	116
50	2	1325	350	375
10	3	285	55	60
25	3	3275	559	571

Além da redução de complexidade computacional obtida, o filtro Volterra parcialmente interpolado é menos sensível a plantas com baixo grau de similaridade entre coeficientes do que a versão totalmente interpolada. A estrutura da Figura 4.2, formada por um filtro linear adaptativo não-interpolado em paralelo com blocos de ordem superior interpolados, em geral, apresenta um desempenho melhor ou, no mínimo, igual ao obtido por uma solução puramente linear.

### 4.3 Simulações

Para ilustrar o desempenho das estruturas propostas, são apresentados resultados de simulação considerando um problema de identificação de sistema. Através de exemplos, o filtro Volterra adaptativo interpolado e o parcialmente interpolado são comparados, em relação ao comportamento do erro quadrático médio, com filtros Volterra adaptativos convencionais e filtros FIR lineares adaptativos. Nos exemplos que apresentam filtros de segunda ordem, também são realizadas comparações com implementações do filtro Volterra adaptativo simplificado [15]. Esse último consiste de uma implementação esparsa do filtro Volterra adaptativo de segunda ordem que despreza os coeficientes de segunda ordem mais distantes da diagonal principal, uma vez que esses coeficientes, em muitas aplicações práticas, são de menor importância [15]. Isso pode ser mais bem observado considerando-se a relação entrada/saída do bloco de segunda ordem, dada por

$$y_2(n) = \sum_{k=0}^K \sum_{m=0}^{N-k-1} h_2(m, m+k)x(n-m)x(n-m-k), \quad (4.8)$$

onde  $y_2(n)$  é o sinal de saída do bloco de segunda ordem;  $x(n)$  denota o sinal de entrada;  $h_2(m, m+k)$  representa os coeficientes de segunda ordem; e  $N$  é o tamanho de memória do filtro. Em (4.8) tem-se ainda o fator  $K$  que determina a quantidade de coeficientes do

filtro Volterra que é desprezada na implementação simplificada. Escolhendo  $K = N - 1$  tem-se a implementação de um filtro Volterra completo. Reduzindo-se esse valor estaremos desprezando os coeficientes mais afastados da diagonal principal [15]. O passo de adaptação usado para as diferentes estruturas está indicado em cada um dos exemplos considerados. O fator de interpolação utilizado é  $L = 2$  e a resposta ao impulso do interpolador é dada por  $\mathbf{I} = [0,5 \ 1 \ 0,5]^T$ . A variância do ruído  $z(n)$  é  $\sigma_z^2 = 0,001$ .

*Exemplo 4.1:* Para este exemplo, a planta é um filtro linear com resposta ao impulso dada por  $[-0,05 \ -0,1 \ 0 \ 0,15 \ 0,32 \ 0,4 \ 0,32 \ 0,15 \ 0 \ -0,1 \ -0,05]^T$  seguido por uma não-linearidade sem memória, de tal forma que o sinal desejado é obtido através de  $d(n) = y_f(n) + 0,3y_f^2(n)$ , onde  $y_f(n)$  é a saída do filtro linear. O sinal de entrada é branco e Gaussiano, com variância unitária. A escolha do passo de adaptação é feita levando-se em consideração um limite superior, denotado aqui por  $\mu_{\max}$ , dado por [5]

$$\mu_{\max} = \frac{1}{3\text{tr}[\mathbf{R}]}, \quad (4.9)$$

onde  $\mathbf{R}$  é a matriz de autocorrelação do sinal de entrada e  $\text{tr}[\cdot]$  corresponde ao operador traço da matriz. Apesar de (4.9) ser obtida de forma aproximada [5], esse limite é bastante próximo daquele obtido na prática. Também são consideradas as particularidades das matrizes de correlação do sinal de entrada para cada caso. Com isso, os valores do passo de adaptação usados e as dimensões de cada um dos filtros são:  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,03$  para o filtro linear de 11 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,0033$  para o filtro Volterra convencional de 77 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,05$  para o filtro Volterra interpolado de 27 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,04$  para o filtro Volterra parcialmente interpolado de 32 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,006$  para o filtro Volterra simplificado com  $K = 2$  e 32 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,005$  para o filtro Volterra simplificado com  $K = 3$  e 41 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,0046$  para o filtro Volterra simplificado com  $K = 4$  e 49 coeficientes. As curvas do erro quadrático médio foram obtidas através de simulação Monte Carlo (média de 200 realizações). A Figura 4.3 mostra uma comparação entre o filtro linear adaptativo, o filtro Volterra

adaptativo convencional e as estruturas propostas neste trabalho. Levando-se em conta a redução de complexidade, as abordagens interpoladas apresentaram um desempenho comparável ao do filtro Volterra convencional, especialmente o do filtro Volterra adaptativo parcialmente interpolado. A Figura 4.4 mostra uma comparação entre as implementações do filtro Volterra adaptativo simplificado [15] e o filtro parcialmente interpolado. Podemos observar que, para este exemplo, o filtro simplificado tem um desempenho inferior ao do filtro parcialmente interpolado, mesmo usando um número maior de coeficientes (49 contra 32).

*Exemplo 4.2:* Para este exemplo, a planta usada é semelhante à do exemplo anterior, com diferença apenas na parte linear, que possui uma resposta ao impulso dada por  $[0,4 \ -0,2 \ 0,1 \ 0,15 \ -0,05 \ -0,1 \ 0 \ 0,2 \ 0,3 \ -0,25 \ -0,02]^T$ . É possível notar que tal planta possui baixo grau de similaridade entre as amostras, afetando sobremaneira o desempenho de uma abordagem interpolada. A não-linearidade sem memória e o sinal de entrada são os mesmos do Exemplo 4.1. O número de coeficientes e os valores do passo de adaptação são também idênticos aos do Exemplo 4.1, exceto para os filtros Volterra simplificados que apresentam:  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,005$  para o filtro Volterra simplificado com  $K = 7$  e 67 coeficientes;  $0,2\mu_{\max}$  com  $\mu_{\max} = 0,0034$  para o filtro Volterra simplificado com  $K = 9$  e 74 coeficientes. A Figura 4.5 ilustra os resultados obtidos através de simulação Monte Carlo (média de 200 realizações). Como esperado, o filtro Volterra adaptativo interpolado apresenta pior desempenho do que o filtro linear adaptativo, devido às características da planta utilizada. Por outro lado, o filtro Volterra adaptativo parcialmente interpolado tem desempenho um pouco melhor do que o filtro linear, confirmando as suposições apresentadas na Seção 4.2. Através da comparação com o filtro Volterra simplificado [15] da Figura 4.6, podemos notar que o desempenho do filtro parcialmente interpolado (32 coeficientes além de 3 coeficientes do interpolador) é bem próximo ao do filtro Volterra simplificado de 67 coeficientes.

*Exemplo 4.3:* Neste exemplo, a não-linearidade sem memória é dada por  $d(n) = 2 \tanh[2,5y_f(n)]$ , onde  $y_f(n)$  é a saída de uma planta linear com coeficientes

$[0,1 \ 0,3 \ 0,32 \ 0,3 \ 0,1]^T$ . Agora, os filtros Volterra adaptativos possuem blocos de primeira e de terceira ordens devido ao tipo de não-linearidade da planta usada. O tamanho de memória dos filtros adaptativos é igual a 5 e o sinal de entrada é branco Gaussiano com variância  $\sigma_x^2 = 0,5$ . Os valores do passo de adaptação são determinados experimentalmente levando-se em conta o limite de estabilidade do algoritmo. Os passos de adaptação e o número de coeficientes para cada estrutura simulada são:  $\mu = 0,0067$  para o filtro linear de 5 coeficientes;  $\mu = 0,0005$  para o filtro Volterra convencional de 40 coeficientes;  $\mu = 0,0001$  para o filtro Volterra interpolado de 13 coeficientes;  $\mu = 0,00015$  para o filtro Volterra parcialmente interpolado de 15 coeficientes. A Figura 4.7 apresenta a comparação das curvas do erro quadrático médio do filtro linear adaptativo e do filtro Volterra adaptativo com o filtro Volterra adaptativo interpolado (a) e com o parcialmente interpolado (b). Novamente, é possível verificar o bom desempenho das estruturas propostas, especialmente o do filtro Volterra adaptativo parcialmente interpolado, que apresenta resultados muito bons para este caso.

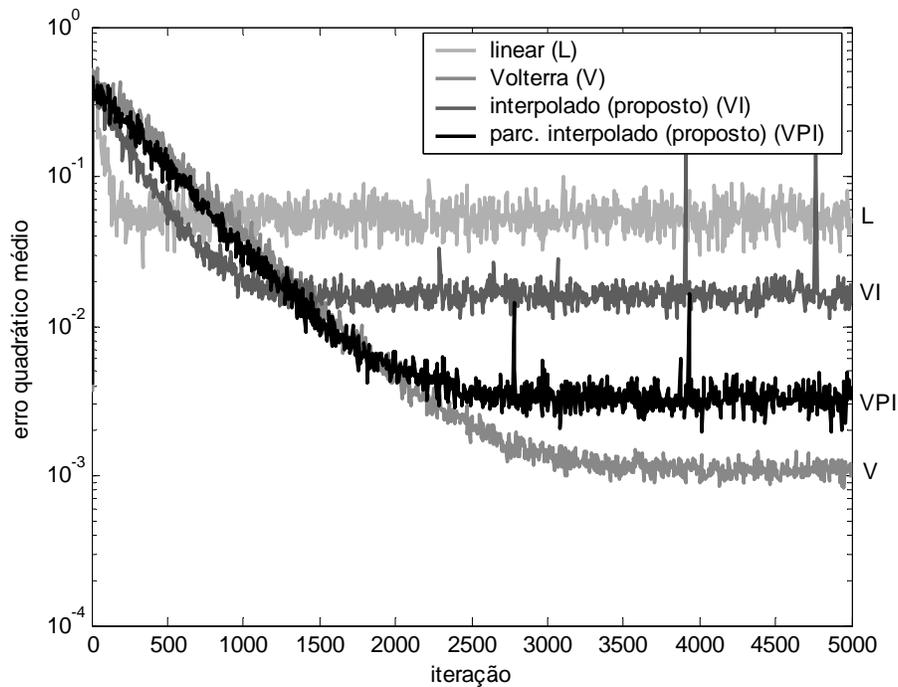


Figura 4.3. Exemplo 4.1. Evolução do erro quadrático médio (média de 200 realizações).

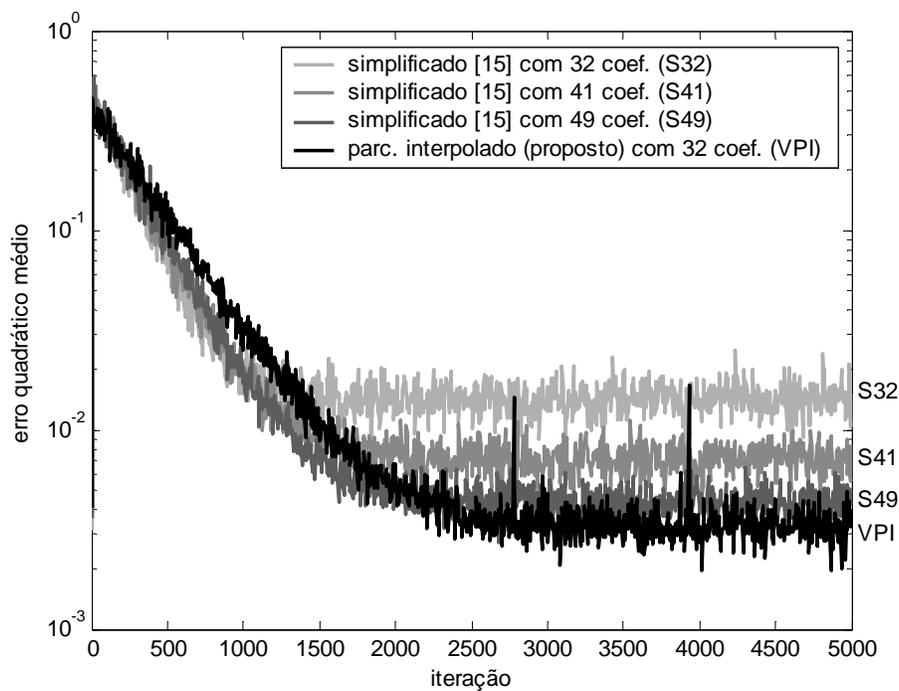


Figura 4.4. Exemplo 4.1. Evolução do erro quadrático médio (média de 200 realizações).

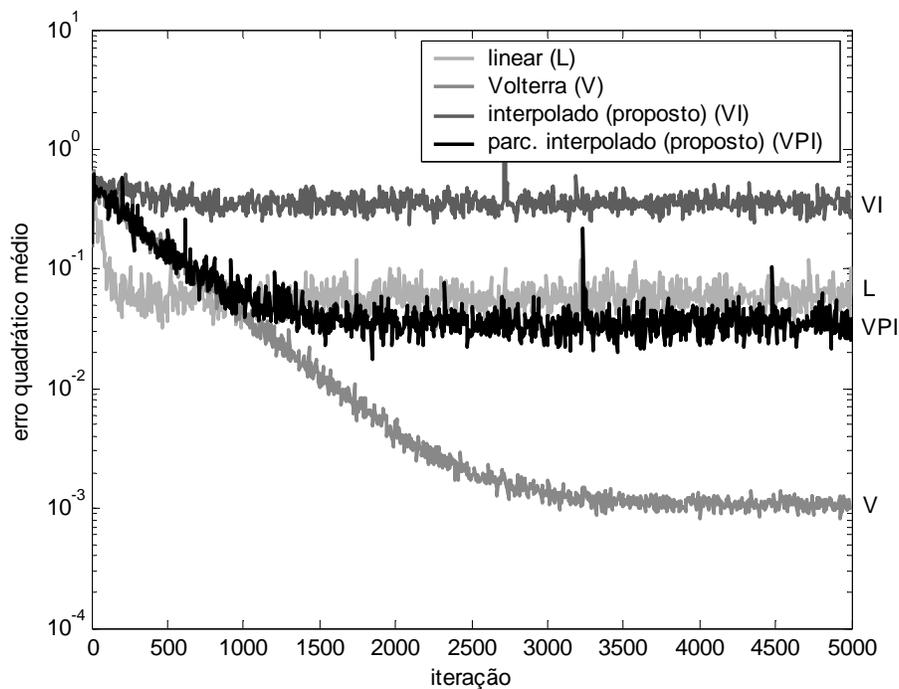


Figura 4.5. Exemplo 4.2. Evolução do erro quadrático médio (média de 200 realizações).

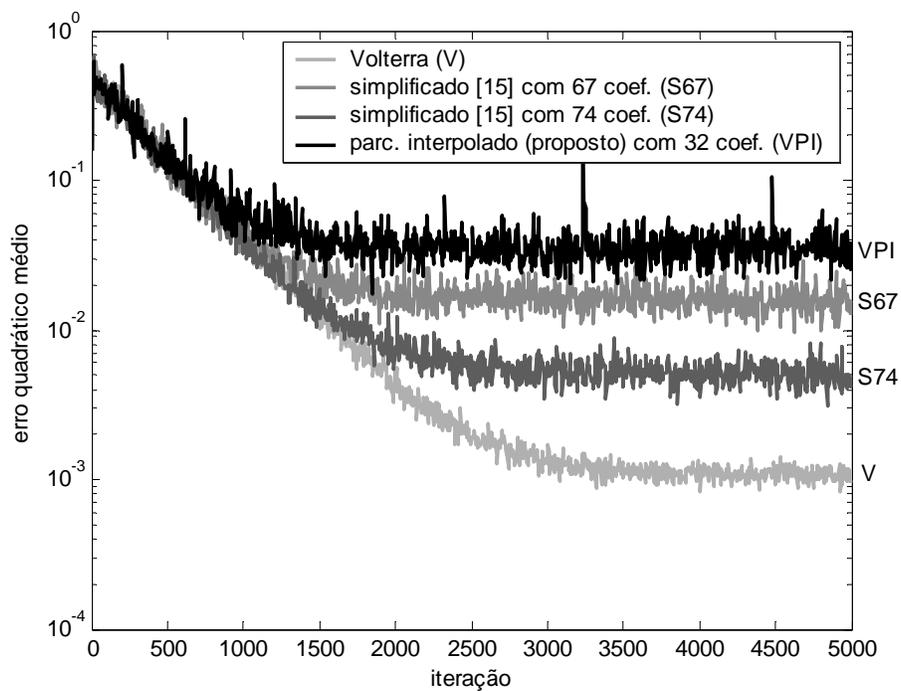


Figura 4.6. Exemplo 4.2. Evolução do erro quadrático médio (média de 200 realizações).

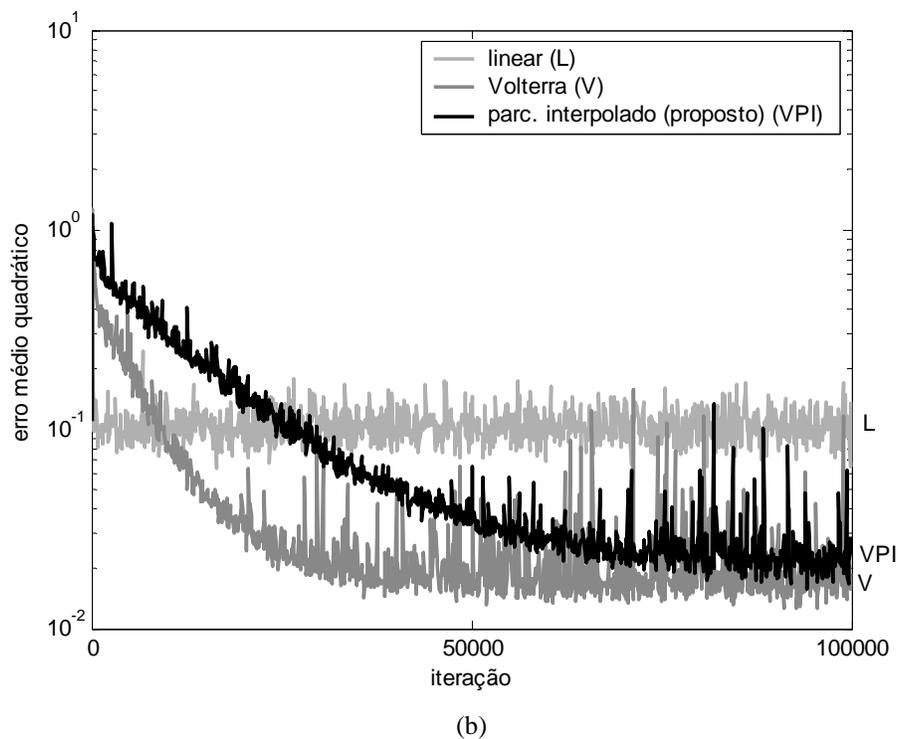
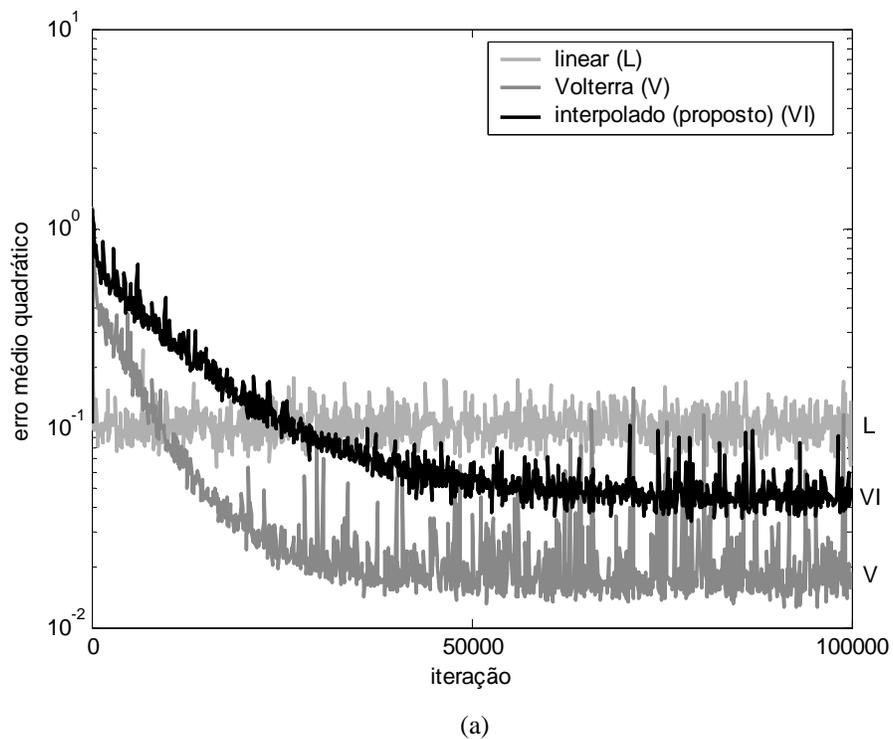


Figura 4.7. Exemplo 4.3. Evolução do erro quadrático médio (média de 200 realizações). (a) Comparação com o filtro Volterra interpolado. (b) Comparação com o filtro Volterra parcialmente interpolado.

#### **4.4 Considerações**

Neste capítulo, foram apresentadas estruturas simplificadas para a implementação de filtros Volterra adaptativos. O uso de uma abordagem interpolada permitiu a implementação de soluções subótimas com considerável redução de complexidade computacional. Através de simulações, foi possível verificar o desempenho e a aplicabilidade dos algoritmos propostos, particularmente, quando restrições de complexidade são consideradas. Tais resultados também originaram uma publicação [30], considerando as estruturas propostas.

## Análise Estatística com Restrição nos Coeficientes de Filtros Volterra Adaptados pelo Algoritmo LMS e sua Aplicação a Filtros Interpolados e com Atraso

Neste capítulo apresenta-se uma análise estatística do filtro Volterra adaptativo com restrição nos pesos. O objetivo é tratar dos filtros Volterra adaptativos interpolados e parcialmente interpolados (Capítulo 4), dos filtros Volterra adaptativos com atraso [14], e ainda dos filtros Volterra adaptativos simplificados [15]. Expressões para o vetor ótimo dos pesos e os momentos de primeira e segunda ordem são desenvolvidas levando-se em conta as restrições nos coeficientes apresentadas por esses filtros devido à sua topologia esparsa. Resultados de simulações são apresentados para verificar a precisão das expressões obtidas.

### 5.1 Vetor de Coeficientes Ótimo com Restrições

O interesse aqui é lidar com filtros Volterra adaptativos com restrições nos pesos, que apresentam algum tipo de esparsidade. Como em [31] e [28], as restrições impostas ao filtro podem então ser expressas da seguinte forma:

$$\mathbf{C}^T \mathbf{h}_v = \mathbf{f}, \quad (5.1)$$

onde  $\mathbf{C}$  é a matriz de restrição,  $\mathbf{f}$  o vetor de resposta e  $\mathbf{h}_v$  o vetor de coeficientes do filtro Volterra. Temos agora um problema de otimização definido pela minimização do erro quadrático médio submetido às restrições dadas por (5.1), ou seja,

$$\begin{aligned} &\text{minimizar } E[e^2(n)] \\ &\text{submetido a } \mathbf{C}^T \mathbf{h}_v = \mathbf{f}. \end{aligned}$$

Supõe-se agora um filtro Volterra adaptativo no contexto da estimação de um sinal desejado  $d(n)$  baseado em um sinal de excitação  $x(n)$ . Esse esquema está ilustrado na Figura 3.1. Temos a partir dessa figura que o sinal de erro instantâneo  $e(n)$  é dado por

$$e(n) = d(n) - y(n) = d(n) - \mathbf{h}_v^T \mathbf{x}_v(n), \quad (5.2)$$

onde  $y(n)$  é o sinal de saída do filtro Volterra adaptativo,  $\mathbf{x}_v(n)$  o vetor de entrada do filtro Volterra e  $\mathbf{h}_v$  o vetor de coeficientes. Elevando-se (5.2) ao quadrado e tomando-se seu valor esperado, obtém-se

$$E[e^2(n)] = E[d^2(n)] - 2E[d(n)\mathbf{x}_v^T(n)]\mathbf{h}_v + \mathbf{h}_v^T E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]\mathbf{h}_v. \quad (5.3)$$

Substituindo-se  $\mathbf{p} = E[d(n)\mathbf{x}_v(n)]$  e  $\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]$  em (5.3), a seguinte expressão para o erro quadrático médio é obtida:

$$\xi(n) = E[e^2(n)] = E[d^2(n)] - 2\mathbf{p}_v^T \mathbf{h}_v + \mathbf{h}_v^T \mathbf{R}_{vv} \mathbf{h}_v. \quad (5.4)$$

Para incluir as restrições na função custo (erro quadrático médio), o método dos multiplicadores de Lagrange é utilizado [32] [31] [29]. Assim, adicionando-se a (5.4) o vetor de multiplicadores de Lagrange  $\lambda$ , obtém-se a seguinte função custo:

$$\begin{aligned} J &= E[e^2(n)] + \lambda^T (\mathbf{C}^T \mathbf{h}_v - \mathbf{f}) \\ &= E[d^2(n)] - 2\mathbf{p}_v^T \mathbf{h}_v + \mathbf{h}_v^T \mathbf{R}_{vv} \mathbf{h}_v + \lambda^T (\mathbf{C}^T \mathbf{h}_v - \mathbf{f}). \end{aligned} \quad (5.5)$$

Para obter o valor de  $\mathbf{h}_v$  com restrições que minimize o erro quadrático médio, faz-se  $\nabla_{\mathbf{h}_v} J = 0$ . Assim,

$$\nabla_{\mathbf{h}_v} J = -2\mathbf{p}_v + 2\mathbf{R}_{vv} \mathbf{h}_v + \mathbf{C}\lambda = 0, \quad (5.6)$$

de onde se obtém o valor ótimo com restrições dos pesos em função de  $\lambda$ , dado por

$$\mathbf{h}_{v_o} = \frac{1}{2} \mathbf{R}_{vv}^{-1} (2\mathbf{p}_v - \mathbf{C}\lambda). \quad (5.7)$$

Como (5.7) deve também satisfazer às restrições dadas por (5.1), podemos escrever

$$\mathbf{C}^T \mathbf{h}_{v_o} = \mathbf{f} = \frac{1}{2} \mathbf{C}^T \mathbf{R}_{vv}^{-1} (2\mathbf{p}_v - \mathbf{C}\lambda). \quad (5.8)$$

Isolando-se  $\lambda$  em (5.8), obtém-se o vetor dos multiplicadores de Lagrange, que é dado por

$$\lambda = -2 \left[ \mathbf{C}^T \mathbf{R}_{vv}^{-1} \mathbf{C} \right]^{-1} \cdot (\mathbf{f} - \mathbf{C}^T \mathbf{R}_{vv}^{-1} \mathbf{p}_v). \quad (5.9)$$

Finalmente, através da substituição de (5.9) em (5.7), o vetor de pesos ótimo com restrições é obtido. Assim,

$$\mathbf{h}_{v_o} = \mathbf{R}_{vv}^{-1} \mathbf{p}_v + \mathbf{R}_{vv}^{-1} \mathbf{C} \left( \mathbf{C}^T \mathbf{R}_{vv}^{-1} \mathbf{C} \right)^{-1} (\mathbf{f} - \mathbf{C}^T \mathbf{R}_{vv}^{-1} \mathbf{p}_v). \quad (5.10)$$

Para os filtros Volterra interpolados, com atraso e simplificados, o vetor de resposta  $\mathbf{f}$  é um vetor de zeros. Isso ocorre pois os coeficientes que sofrem restrições são anulados, ou seja, fixados em zero. Com isso, é possível reescrever (5.10) como

$$\mathbf{h}_{v_0} = \left[ \mathbf{I} - \mathbf{R}_{VV}^{-1} \mathbf{C} (\mathbf{C}^T \mathbf{R}_{VV}^{-1} \mathbf{C})^{-1} \mathbf{C}^T \right] \mathbf{R}_{VV}^{-1} \mathbf{p}_V, \quad (5.11)$$

onde  $\mathbf{I}$  é a matriz identidade com dimensão idêntica à  $\mathbf{R}_{VV}$ .

## 5.2 Equação de Atualização dos Pesos com Restrições

Como descrito na Seção 1.2.3, o algoritmo LMS é uma implementação estocástica do método *steepest descent* obtida através da substituição da função custo por sua estimativa instantânea. Do algoritmo *steepest descent* (Seção 1.2.2), temos ainda que a atualização dos pesos é feita na direção oposta ao gradiente da função custo. Considerando-se a função custo com restrições, a atualização dos pesos é realizada como segue:

$$\mathbf{h}_V(n+1) = \mathbf{h}_V(n) - \mu \nabla_{\mathbf{h}_V} J. \quad (5.12)$$

Escrevendo-se o valor instantâneo da função custo (5.5) como

$$\begin{aligned} J(n) = & d^2(n) - 2d(n)\mathbf{x}_V(n)\mathbf{h}_V(n) \\ & + \mathbf{h}_V^T(n)\mathbf{x}_V(n)\mathbf{x}_V^T(n)\mathbf{h}_V(n) + \lambda^T(\mathbf{C}^T\mathbf{h}_V(n) - \mathbf{f}), \end{aligned} \quad (5.13)$$

temos seu gradiente instantâneo dado por

$$\nabla_{\mathbf{h}_V} J(n) = -2d(n)\mathbf{x}_V(n) - 2z(n)\mathbf{x}_V(n) + 2\mathbf{x}_V(n)\mathbf{x}_V^T(n)\mathbf{h}_V(n) + \mathbf{C}\lambda(n). \quad (5.14)$$

Substituindo-se agora (5.14) em (5.12) e ainda com  $e(n) = d(n) - \mathbf{x}_V^T(n)\mathbf{h}_V(n)$ , obtém-se

$$\mathbf{h}_V(n+1) = \mathbf{h}_V(n) + 2\mu e(n)\mathbf{x}_V(n) - \mu \mathbf{C}\lambda(n). \quad (5.15)$$

Os multiplicadores de Lagrange são obtidos impondo a restrição de (5.1) a (5.15), resultando em

$$\mathbf{f} = \mathbf{C}^T\mathbf{h}_V(n+1) = \mathbf{C}^T\mathbf{h}_V(n) + 2\mu\mathbf{C}^T e(n)\mathbf{x}_V(n) - \mu\mathbf{C}^T\mathbf{C}\lambda(n). \quad (5.16)$$

Como  $\mathbf{C}^T\mathbf{C} = \mathbf{I}$  (matriz identidade), isolamos  $\lambda(n)$  em (5.16), obtendo-se

$$\lambda(n) = -\frac{1}{\mu} \left[ \mathbf{f} - \mathbf{C}^T\mathbf{h}_V(n) - 2\mu\mathbf{C}^T e(n)\mathbf{x}_V(n) \right]. \quad (5.17)$$

Finalmente, substituindo-se (5.17) em (5.15), obtém-se a equação de atualização dos coeficientes com restrições, dada por

$$\mathbf{h}_V(n+1) = \mathbf{P} \left[ \mathbf{h}_V(n) + 2\mu e(n)\mathbf{x}_V(n) \right] + \mathbf{C}\mathbf{f}, \quad (5.18)$$

onde  $\mathbf{P} = \mathbf{I} - \mathbf{C}\mathbf{C}^T$ .

Analisando-se a matriz  $\mathbf{P}$ , é possível constatar que ela possui elementos apenas na diagonal principal e, ainda, que esses elementos são iguais a um quando correspondentes aos pesos onde não há restrição e zero quando correspondentes aos pesos onde são impostas as restrições. Assim, como o vetor  $\mathbf{f}$  é um vetor de zeros para as implementações aqui consideradas, podemos observar que a equação de atualização (5.18) nada mais é do que a aplicação do algoritmo LMS apenas aos pesos não-restritos.

### 5.3 Comportamento Médio dos Pesos

Para obter uma expressão recursiva para o comportamento médio dos pesos, considera-se primeiramente um sinal de ruído descorrelacionado, com média zero, adicionado ao sinal de erro instantâneo de (5.2). Com isso, tem-se

$$e(n) = d(n) + z(n) - \mathbf{h}_v^T(n) \mathbf{x}_v(n). \quad (5.19)$$

Substituindo-se (5.19) na equação de atualização dos pesos com restrições, dada por (5.18), e tomando-se seu valor esperado, obtém-se

$$\begin{aligned} E[\mathbf{h}_v(n+1)] = & \mathbf{P}\{E[\mathbf{h}_v(n)] + 2\mu E[d(n)\mathbf{x}_v(n)] \\ & + 2\mu E[z(n)\mathbf{x}_v(n)] - 2\mu E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]\} + \mathbf{Cf}. \end{aligned} \quad (5.20)$$

O problema para prosseguir com esta análise é a resolução de  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ . Como considerado na Seção 3.2, é possível assumir uma fraca dependência entre o vetor de pesos e o vetor de entrada para pequenos valores do passo de atualização [5] [28], resultando em (3.10). Como o ruído  $z(n)$  é descorrelacionado dos demais sinais do sistema e possui média zero, temos  $E[z(n)\mathbf{x}_v(n)] = 0$ . Fazendo-se  $\mathbf{p}_v = E[d(n)\mathbf{x}_v(n)]$ ,  $\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]$  e substituindo-se (3.10) em (5.20), obtém-se a expressão para o comportamento médio dos pesos, dada por

$$E[\mathbf{h}_v(n+1)] = \mathbf{P}(\mathbf{I} - 2\mu\mathbf{R}_{vv})E[\mathbf{h}_v(n)] + 2\mu\mathbf{P}\mathbf{p}_v + \mathbf{Cf}. \quad (5.21)$$

Para os casos já comentados onde  $\mathbf{f}$  é um vetor nulo, é possível reescrever (5.21) como

$$E[\mathbf{h}_v(n+1)] = \mathbf{P}(\mathbf{I} - 2\mu\mathbf{R}_{vv})E[\mathbf{h}_v(n)] + 2\mu\mathbf{P}\mathbf{p}_v. \quad (5.22)$$

#### 5.4 Vetor de Erro dos Pesos

Com o vetor de erro dos pesos, definido por  $\mathbf{v}(n) = \mathbf{h}_v(n) - \mathbf{h}_{v_0}$ , manipulamos (5.21) para obter

$$\begin{aligned} E[\mathbf{v}(n+1)] + \mathbf{h}_{v_0} &= \mathbf{P}(\mathbf{I} - 2\mu\mathbf{R}_{VV})E[\mathbf{v}(n)] \\ &+ \mathbf{P}\mathbf{h}_{v_0} - 2\mu\mathbf{P}\mathbf{R}_{VV}\mathbf{h}_{v_0} + 2\mu\mathbf{P}\mathbf{p}_v + \mathbf{Cf}. \end{aligned} \quad (5.23)$$

Podemos verificar que  $\mathbf{h}_{v_0} = \mathbf{P}\mathbf{h}_{v_0}$ , visto que, multiplicando-se o vetor de pesos ótimo pela matriz  $\mathbf{P}$ , apenas são multiplicados por zero os elementos nulos desse vetor e por um os demais elementos. Também temos que  $\mathbf{P}\mathbf{R}_{VV}\mathbf{h}_{v_0} = \mathbf{P}\mathbf{p}_v$ . Isso ocorre pois as diferenças entre os vetores  $\mathbf{R}_{VV}\mathbf{h}_{v_0}$  e  $\mathbf{p}_v$  estão justamente nos elementos que são anulados quando tais elementos são multiplicados pela matriz  $\mathbf{P}$ . Com essas considerações e manipulando-se a expressão (5.23), obtém-se a expressão recursiva para o comportamento do vetor de erro dos pesos, dada por

$$E[\mathbf{v}(n+1)] = \mathbf{P}(\mathbf{I} - 2\mu\mathbf{R}_{VV})E[\mathbf{v}(n)]. \quad (5.24)$$

#### 5.5 Erro Quadrático Médio

A expressão para o erro quadrático médio é obtida de maneira similar ao que foi realizado na Seção 3.5. Primeiramente o erro instantâneo é escrito como

$$e(n) = e_o(n) - \mathbf{v}^T(n)\mathbf{x}_v(n). \quad (5.25)$$

Elevando-se (5.25) ao quadrado e tomando-se seu valor esperado, obtém-se

$$E[e^2(n)] = E[e_o^2(n)] - 2E[e_o(n)\mathbf{v}^T(n)\mathbf{x}_v(n)] + E[\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)]. \quad (5.26)$$

Realizando-se o mesmo procedimento da Seção 3.5, a expressão para curva de aprendizagem é obtida. Assim,

$$\xi(n) = \xi_{\min} + \text{tr}\{\mathbf{K}(n)\mathbf{R}_{VV}\}. \quad (5.27)$$

#### 5.6 Aproximação para o Momento de Segunda Ordem

De maneira similar à Seção 3.6, substituindo-se  $\mathbf{v}(n) = \mathbf{h}_v(n) - \mathbf{h}_{v_0}$  em (5.18), obtém-se

$$\mathbf{v}(n+1) + \mathbf{h}_{v_0} = \mathbf{P}[\mathbf{v}(n) + \mathbf{h}_{v_0} + 2\mu e(n)\mathbf{x}_v(n)] + \mathbf{Cf}. \quad (5.28)$$

Como mencionado anteriormente,  $\mathbf{Cf}$  é uma matriz nula para os filtros interpolados, os filtros com atraso e os filtros simplificados. Assim, considerando-se  $\mathbf{P}\mathbf{h}_{v_0} = \mathbf{h}_{v_0}$  e  $e(n) = e_o(n) - \mathbf{v}^T(n)\mathbf{x}_v(n)$ , é possível reescrever (5.28) como

$$\mathbf{v}(n+1) = \mathbf{P} \left[ \mathbf{v}(n) + 2\mu e_o(n)\mathbf{x}_v(n) - 2\mu \mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n) \right]. \quad (5.29)$$

Levando-se em conta que  $\mathbf{P} = \mathbf{P}^T$ , multiplicando-se (5.29) por sua versão transposta, obtém-se

$$\begin{aligned} \mathbf{v}(n+1)\mathbf{v}^T(n+1) &= \\ &\mathbf{P}\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{P} \\ &+ 2\mu\mathbf{P}\mathbf{v}(n)e_o(n)\mathbf{x}_v^T(n)\mathbf{P} \\ &- 2\mu\mathbf{P}\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{P} \\ &+ 2\mu\mathbf{P}e_o(n)\mathbf{x}_v(n)\mathbf{v}^T(n)\mathbf{P} \\ &+ 4\mu^2\mathbf{P}\mathbf{x}_v(n)e_o(n)e_o(n)\mathbf{x}_v^T(n)\mathbf{P} \\ &- 4\mu^2\mathbf{P}e_o(n)\mathbf{x}_v(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{P} \\ &- 2\mu\mathbf{P}\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{P} \\ &- 4\mu^2\mathbf{P}\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)e_o(n)\mathbf{x}_v^T(n)\mathbf{P} \\ &+ 4\mu^2\mathbf{P}\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{P}. \end{aligned} \quad (5.30)$$

Por razões de tratabilidade matemática e simplicidade de resultados, adotamos aqui novamente a aproximação considerada na Seção 3.6. Assim, desprezando-se os termos dependentes de  $\mu^2$  em (5.30) e tomando-se seu valor esperado, uma expressão recursiva aproximada para  $\mathbf{K}(n)$  é obtida. Portanto,

$$\mathbf{K}(n+1) = \mathbf{P}[\mathbf{K}(n) - 2\mu\mathbf{K}(n)\mathbf{R}_{vv} + 2\mu\mathbf{R}_{vv}\mathbf{K}(n)]\mathbf{P}. \quad (5.31)$$

Finalmente, com (5.31) e (5.27) é possível obter a curva de aprendizagem do filtro Volterra adaptativo com restrições nos pesos.

## 5.7 Aplicações da Análise com Restrições nos Pesos

Como comentado no início deste capítulo, a análise com restrições aqui apresentada permite verificar o comportamento dos momentos de primeira e segunda ordens de filtros Volterra adaptados pelo algoritmo LMS considerando-se restrições nos pesos. Nesta seção, a aplicação da análise com restrições nos pesos é descrita para as implementações do filtro Volterra adaptativo mencionadas anteriormente.

### 5.7.1 Filtro Volterra Adaptativo Interpolado

A Figura 5.1 apresenta a estrutura em blocos de um filtro Volterra interpolado adaptativo considerando-se um esquema de identificação de um sinal baseado na excitação  $x(n)$ . Tal estrutura consiste de um filtro interpolador de entrada seguido por um filtro Volterra adaptativo esparsos, como descrito no Capítulo 4. Na Figura 5.1, **I** representa o interpolador,  $x_i(n)$  o sinal de entrada interpolado e  $\mathbf{h}_{vi}(n)$  o filtro Volterra adaptativo esparsos. Os demais elementos são os mesmos considerados na Figura 3.2.

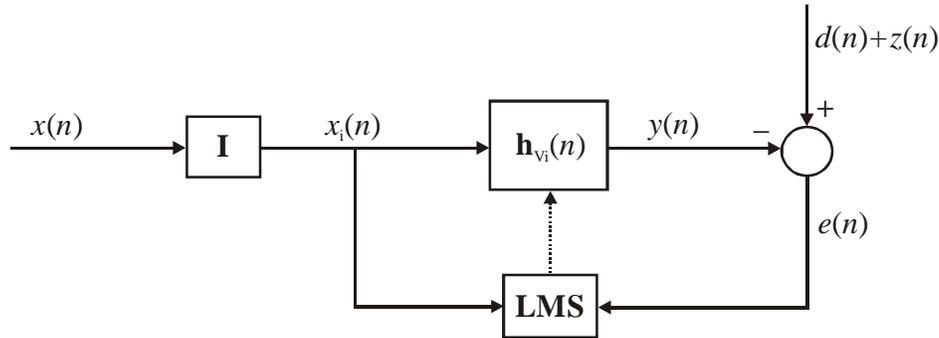


Figura 5.1 – Diagrama de blocos do filtro Volterra adaptativo interpolado.

Para realizar a análise deste filtro, precisamos inicialmente montar a matriz de restrições  $\mathbf{C}$  levando-se em conta quais pesos do filtro vão sofrer restrição. Para exemplificar o procedimento, é feita a análise de um filtro Volterra adaptativo de segunda ordem com tamanho de memória igual a três. Para esse filtro, o vetor de entrada é dado por

$$\mathbf{x}_v(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad x^2(n) \quad x(n)x(n-1) \quad x(n)x(n-2) \quad x^2(n-1) \quad x(n-1)x(n-2) \quad x^2(n-2)]. \quad (5.32)$$

A versão interpolada do filtro Volterra adaptativo com fator de interpolação  $L=2$ , apresenta um vetor de entrada gerado a partir do sinal de entrada interpolado  $x_i(n)$ . As restrições são impostas aos elementos dependentes de  $x(n-1)$  para este caso. Com isso, tem-se o vetor de entrada

$$\mathbf{x}_v(n) = [x_i(n) \quad \underline{x_i(n-1)} \quad x_i(n-2) \quad x_i^2(n) \quad \underline{x_i(n)x_i(n-1)} \quad x_i(n)x_i(n-2) \quad \underline{x_i^2(n-1)} \quad \underline{x_i(n-1)x_i(n-2)} \quad x_i^2(n-2)], \quad (5.33)$$

onde os elementos que sofrem as restrições estão sublinhados. Podemos observar que quatro restrições são impostas, no segundo, quinto, sétimo e oitavo elementos do vetor de

entrada e, conseqüentemente, do vetor de coeficientes do filtro. A matriz de restrição e o vetor de resposta são então dados por

$$\mathbf{C}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ e } \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (5.34)$$

É importante ressaltar que o sinal de entrada para o filtro Volterra adaptativo sujeito a tal análise é  $x_i(n)$ ; portanto, a matriz de correlação de entrada  $\mathbf{R}_{\mathbf{v}_V}$  e o vetor  $\mathbf{p}_V$  devem ser gerados a partir desse sinal e de (5.33). A análise do comportamento do filtro Volterra adaptativo interpolado é então realizada aplicando-se as expressões desenvolvidas nas seções anteriores deste capítulo.

### 5.7.2 Filtro Volterra Adaptativo Parcialmente Interpolado

Como descrito na Seção 4.2, o filtro Volterra adaptativo parcialmente interpolado é uma versão do filtro Volterra adaptativo interpolado que emprega interpolação e esparsidade apenas nos blocos, e conseqüentemente nos pesos, de ordem superior do filtro Volterra. Assim, o vetor de entrada de um filtro de 2ª ordem e com tamanho de memória igual três é

$$\mathbf{x}_V(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad \underline{x_1^2(n)} \quad \underline{x_1(n)x_1(n-1)} \quad \underline{x_1(n)x_1(n-2)} \quad \underline{x_1^2(n-1)} \quad \underline{x_1(n-1)x_1(n-2)} \quad x_1^2(n-2)]. \quad (5.35)$$

Podemos observar que o bloco de segunda ordem do filtro emprega interpolação e esparsidade, resultando em restrições nos elementos sublinhados do vetor. Já o bloco de primeira ordem não utiliza o sinal interpolado e também não possui restrições. Com as três restrições indicadas em (5.35), temos a seguinte matriz de restrições e vetor de resposta:

$$\mathbf{C}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ e } \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (5.36)$$

A análise é então realizada usando-se as expressões desenvolvidas nas seções anteriores, sempre tomando-se o cuidado de montar a matriz  $\mathbf{R}_{\mathbf{v}_V}$  e o vetor  $\mathbf{p}_V$  a partir do vetor definido por (5.35).

### 5.7.3 Filtro Volterra Adaptativo com Atraso

A Figura 5.2 mostra o diagrama em blocos de um filtro Volterra adaptativo com atraso. Esse filtro está descrito em detalhes em [14] e consiste de um bloco de atraso na entrada ( $z^{-d_1}$ ) seguido por um filtro Volterra adaptativo. A primeira vista, a análise de seu comportamento pode ser obtida usando-se apropriadamente uma análise sem restrições, como àquela discutida no capítulo 4. Porém, como descrito em [14], muitas vezes vários desses filtros são utilizados em paralelo, resultando em um filtro com restrições determinadas pelos atrasos e tamanhos de memória de cada um deles. Para tal situação, a análise com restrições se torna bastante vantajosa.

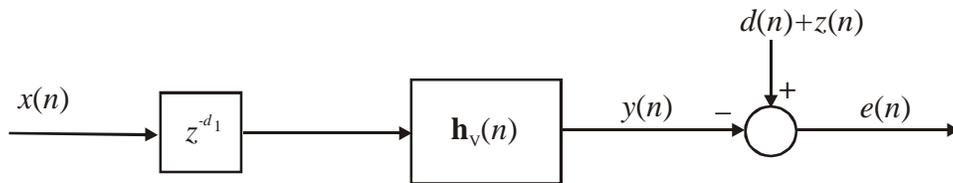


Figura 5.2 – Diagrama de blocos do filtro Volterra adaptativo com atraso aplicado a um problema de estimação de sinal.

Para ilustrar o emprego da análise com restrições, um exemplo simples é mostrado. Assim, a partir do vetor de entrada de um filtro Volterra de segunda ordem com tamanho de memória igual a três (5.32), temos o vetor de entrada de um filtro Volterra adaptativo com atraso de uma amostra na entrada dado por

$$\mathbf{x}_V(n) = [\underline{x(n)} \quad x(n-1) \quad x(n-2) \quad \underline{x^2(n)} \quad \underline{x(n)x(n-1)} \quad \underline{x(n)x(n-2)} \quad x^2(n-1) \quad x(n-1)x(n-2) \quad x^2(n-2)], \quad (5.37)$$

onde os elementos sublinhados são os que sofrem restrição. Assim, temos a matriz de restrição e o vetor de resposta dados por

$$\mathbf{C}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (5.38)$$

Com  $\mathbf{C}$ ,  $\mathbf{f}$  e as expressões obtidas neste capítulo, podemos obter o comportamento dos momentos de primeira e segunda ordens dos coeficientes deste filtro.

### 5.7.4 Filtro Volterra Adaptativo Simplificado

Como comentado na Seção 4.3, o filtro Volterra adaptativo simplificado consiste de uma implementação esparsa do filtro Volterra adaptativo de segunda ordem que apresenta restrição nos coeficientes mais afastados da diagonal principal [15]. Sua relação entrada/saída é descrita por (4.8). Para exemplificar a análise do seu comportamento, consideramos um filtro Volterra adaptativo simplificado com tamanho de memória igual a três e  $K = 1$ . O vetor de entrada é então dado por

$$\mathbf{x}_v(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad x^2(n) \quad \underline{x(n)x(n-1)} \\ \underline{x(n)x(n-2)} \quad x^2(n-1) \quad \underline{x(n-1)x(n-2)} \quad x^2(n-2)],$$

com os elementos que sofrem restrição sublinhados. De forma similar às seções anteriores, a matriz de restrição e o vetor de respostas são dados por

$$\mathbf{C}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Com isso basta usar as expressões desenvolvidas para avaliar o comportamento do filtro adaptativo.

## 5.8 Simulações

Nesta seção são apresentados resultados obtidos por simulação com o objetivo de comprovar a validade da análise com restrição nos pesos. Nessas simulações, os resultados obtidos através das expressões desenvolvidas são comparados com aqueles obtidos através do método de Monte Carlo. Os valores do passo de adaptação usados são relacionados ao valor máximo do passo  $\mu_{\max}$  determinado experimentalmente. Para todos os casos, o ruído aditivo  $z(n)$  possui variância  $\sigma_z^2 = 0,001$ .

*Exemplo 5.1:* O filtro a ser avaliado nesse exemplo é um filtro Volterra adaptativo interpolado de segunda ordem com tamanho de memória igual 3 e fator de interpolação  $L = 2$ . A planta a ser modelada pelo filtro adaptativo possui relação entrada/saída dada por

$$\begin{aligned}
d(n) = & x(n) + 0,5x(n-1) + 0,2x(n-2) + \\
& + 0,5x^2(n) + 0,3x(n)x(n-1) + 0,05x(n)x(n-2) \\
& - 0,1x^2(n-1) - 0,2x(n-1)x(n-2) - 0,05x^2(n-2).
\end{aligned}$$

O sinal de entrada é branco Gaussiano com variância unitária. Neste exemplo, o filtro foi simulado para três valores de  $\mu$ , dados por  $\mu_{\max}/2$ ,  $\mu_{\max}/4$  e  $\mu_{\max}/20$ , com  $\mu_{\max} = 0,02$ . Também foram obtidos por simulação os valores de  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$  e  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$  com a finalidade de verificar a aproximação adotada através de (3.10) (hipótese da independência). A Figura 5.3 apresenta o comportamento médio de alguns dos pesos obtido através de (3.12) em comparação com o resultado obtido através do método de Monte Carlo (média de 100 realizações). Nesta figura podemos notar que existe uma pequena diferença entre a simulação e o resultado do modelo. Isto ocorre, pois para valores grandes do passo de adaptação, a aproximação considerada em (3.10) não apresenta bons resultados. Na Figura 5.4 temos os valores dos vetores  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$  e  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$  obtidos por simulação comprovando tal diferença para o valor do passo de adaptação usado. Com a redução do passo de adaptação para  $\mu_{\max}/4$ , resultados melhores são obtidos, o que pode ser constatado através das Figuras 5.5 e 5.6, seguindo o mesmo padrão das Figuras 5.3 e 5.4. Reduzindo ainda mais o passo de adaptação ( $\mu_{\max}/20$ ) as diferenças praticamente desaparecem, como podemos verificar através das Figuras 5.7 e 5.8. Nas Figuras 5.9, 5.10 e 5.11 estão apresentadas as curvas de aprendizagem para os diferentes valores de  $\mu$ . Novamente os resultados teóricos são comparados com resultados de simulação (média de 500 realizações). Para o maior valor do passo de adaptação ( $\mu_{\max}/2$ ), uma diferença muito pequena pode ser notada. Para os outros casos, podemos observar um casamento muito bom nos resultados obtidos, confirmando as conjecturas previamente apontadas.

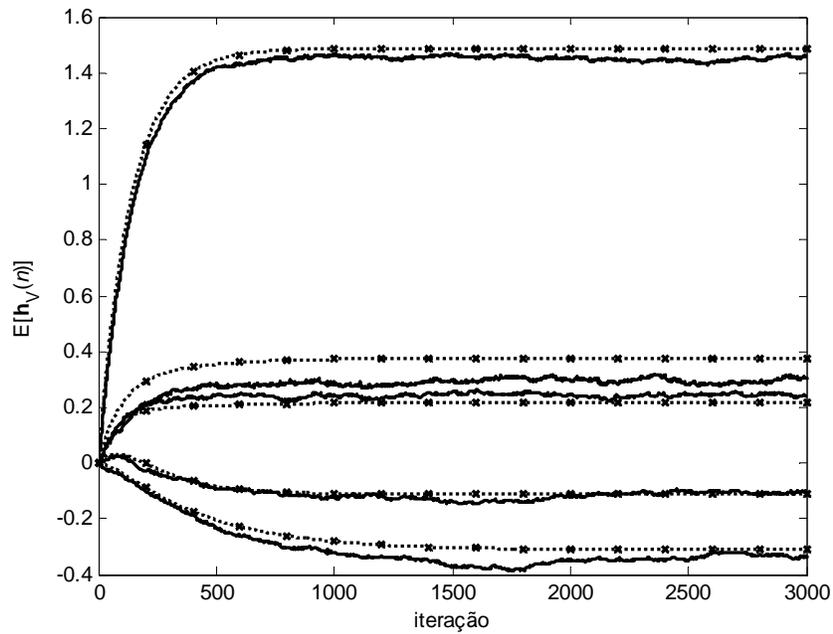


Figura 5.3. Exemplo 5.1. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,5\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

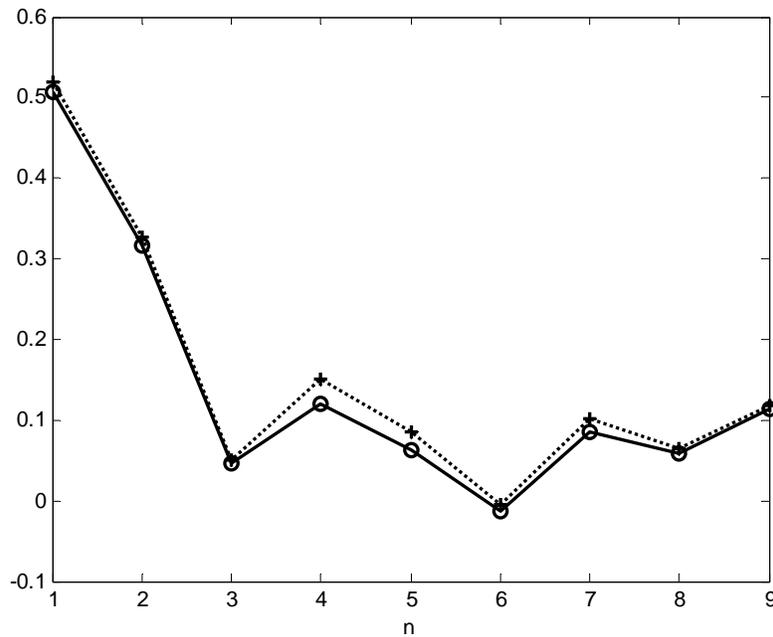


Figura 5.4. Exemplo 5.1. Simulação (média de 100 realizações) usando  $\mu = 0,5\mu_{\max}$ . (Linha sólida)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ .

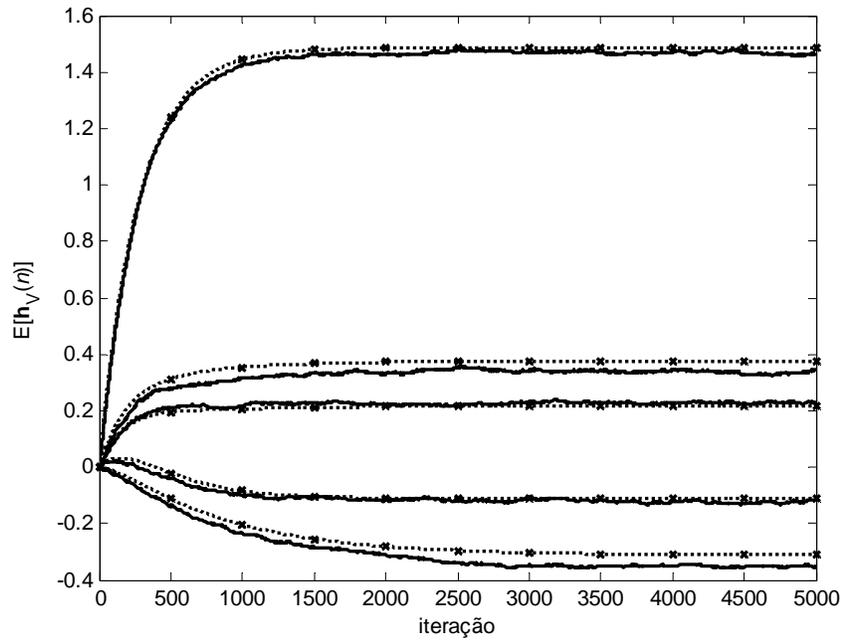


Figura 5.5. Exemplo 5.1. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,25\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

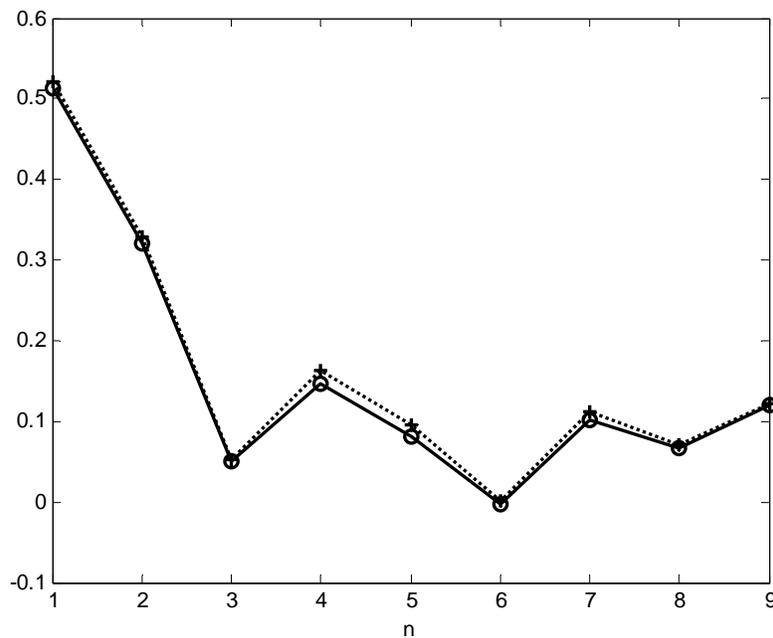


Figura 5.6. Exemplo 5.1. Simulação (média de 100 realizações) usando  $\mu = 0,25\mu_{\max}$ . (Linha sólida)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ .

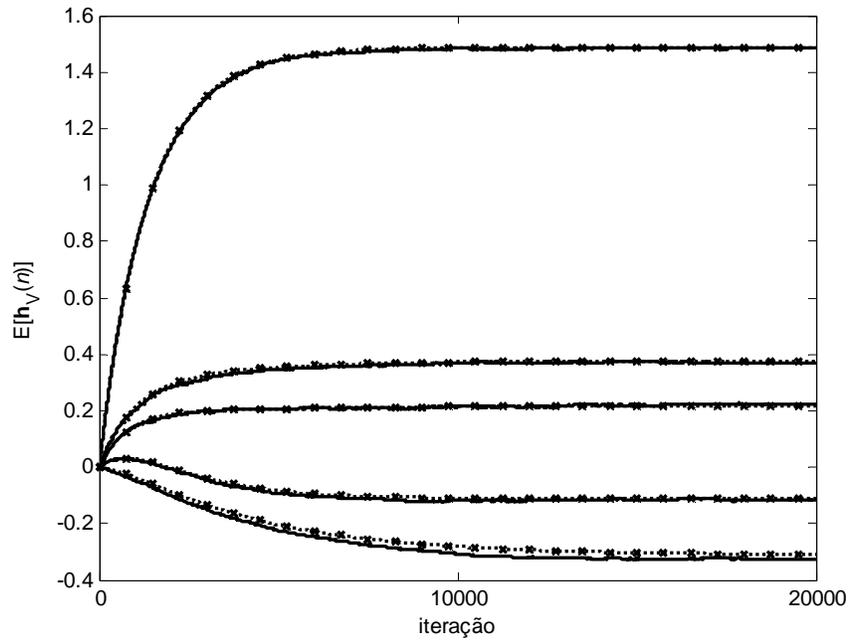


Figura 5.7. Exemplo 5.1. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,05\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

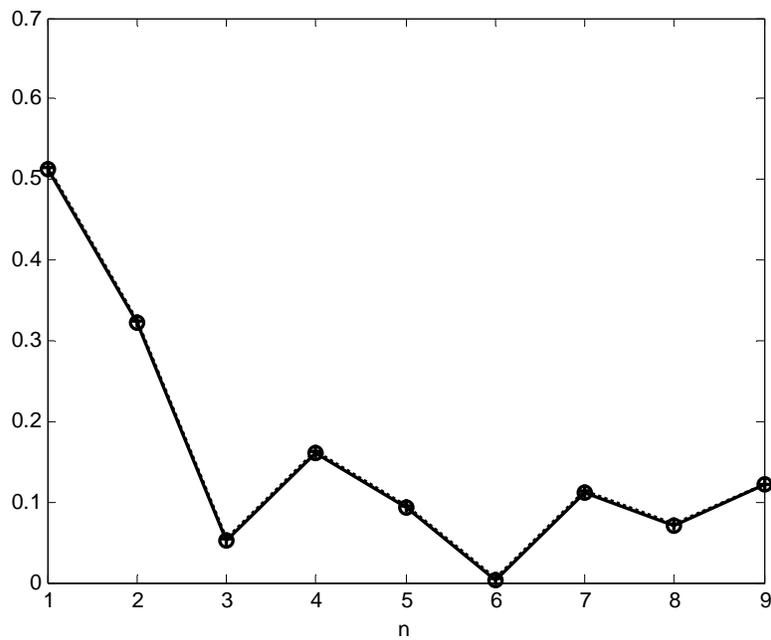


Figura 5.8. Exemplo 5.1. Simulação (média de 100 realizações) usando  $\mu = 0,05\mu_{\max}$ . (Linha sólida)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]E[\mathbf{h}_v(n)]$ ; (Linha pontilhada)  $E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)\mathbf{h}_v(n)]$ .

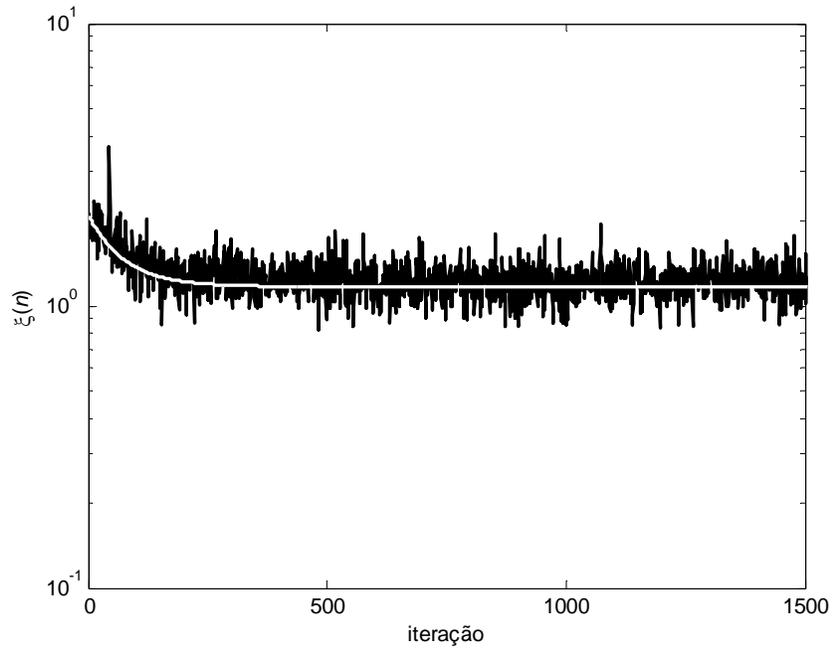


Figura 5.9. Exemplo 5.1. Evolução de  $\xi(n)$  com  $\mu = 0,5\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

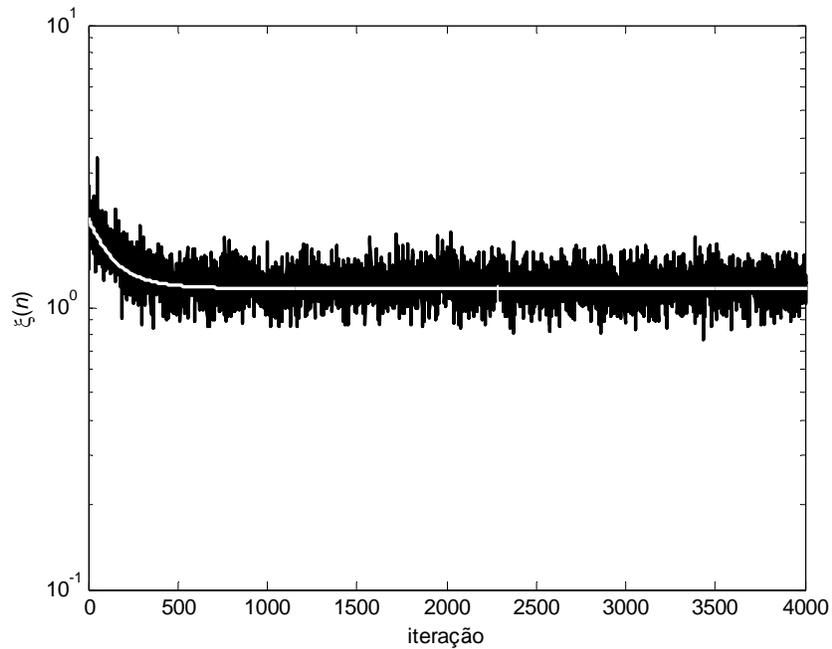


Figura 5.10. Exemplo 5.1. Evolução de  $\xi(n)$  com  $\mu = 0,25\mu_{\max}$ . (Linha preta) simulação(média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

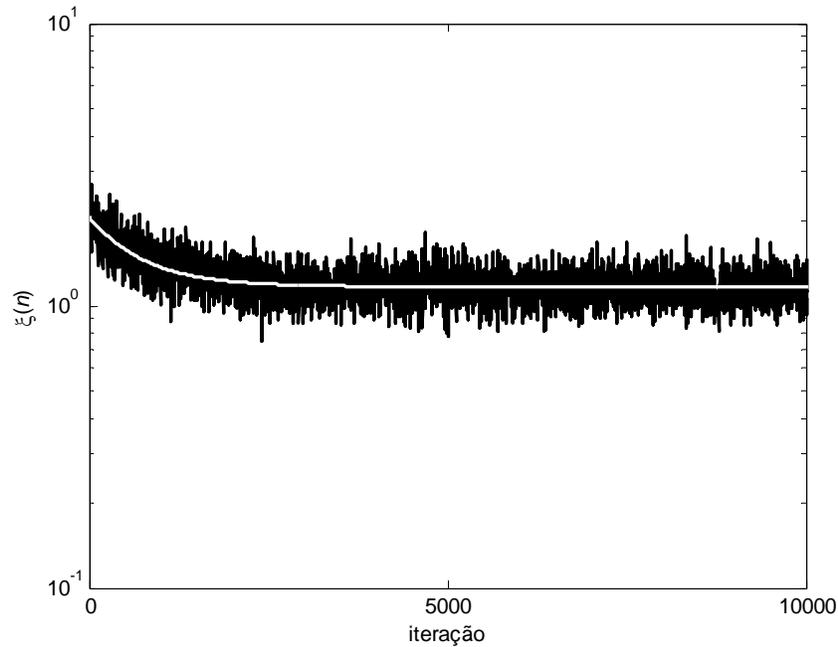


Figura 5.11. Exemplo 5.1. Evolução de  $\xi(n)$  com  $\mu = 0,05\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

*Exemplo 5.2:* O filtro adaptativo deste exemplo é um filtro Volterra parcialmente interpolado (Seção 4.2) com tamanho de memória igual a 5, possuindo um bloco de primeira ordem e um bloco esparsa de terceira ordem. Esse filtro é usado para modelar uma planta Volterra, também com tamanho de memória 5, possuindo os seguintes vetores de coeficientes:

$$\mathbf{h}_1 = [0,12 \ 0,3 \ 0,41 \ 0,3 \ 0,12],$$

$$\begin{aligned} \mathbf{h}_3 = & [0,135 \ -0,08 \ 0,0300 \ -0,005 \ 0,11 \ 0,07 \ -0,01 \ -0,125 \ 0,0850 \\ & -0,015 \ 0,0300 \ 0,07 \ 0,1 \ 0,055 \ -0,075 \ -0,02 \ 0,1 \ 0,095 \ -0,02 \\ & 0,085 \ -0,095 \ -0,03 \ 0,065 \ -0,1 \ -0,07 \ -0,06 \ -0,06 \ 0,02 \\ & -0,04 \ -0,055 \ -0,085 \ 0,045 \ -0,01 \ 0,075 \ -0,005]. \end{aligned}$$

O sinal de entrada é branco Gaussiano com variância unitária e o valor do passo de adaptação utilizado é  $\mu_{\max} / 8$  com  $\mu_{\max} = 0,008$ . A Figura 5.12 apresenta o comportamento médio de alguns dos pesos obtidos pelo modelo proposto, comparando os resultados com aqueles obtidos por simulação (média de 100 realizações). Já a Figura 5.13 mostra a curva de aprendizagem teórica em comparação com a simulação (média de 500 realizações). Novamente, podemos constatar a eficiência da modelagem desenvolvida.

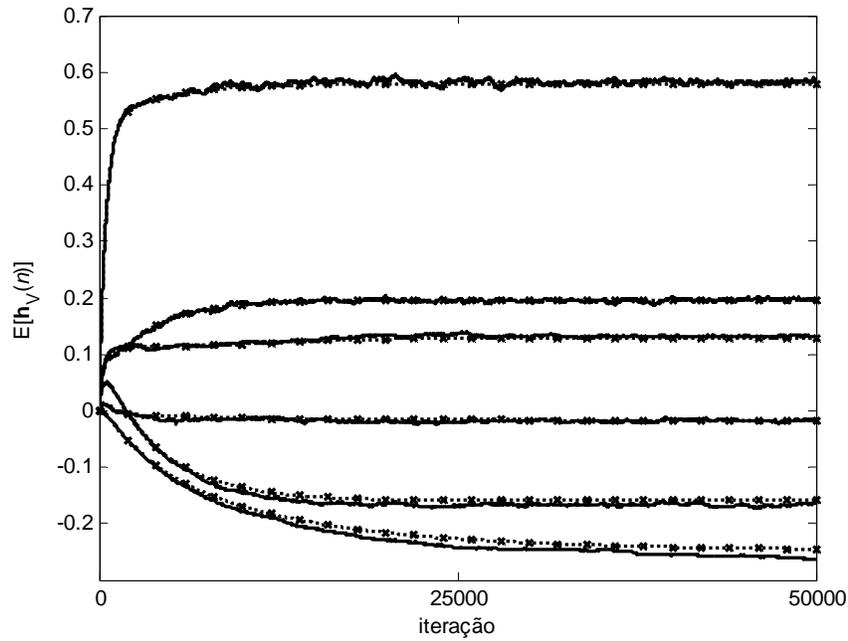


Figura 5.12. Exemplo 5.2. Evolução de  $E[h_v(n)]$  com  $\mu = 0,125\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

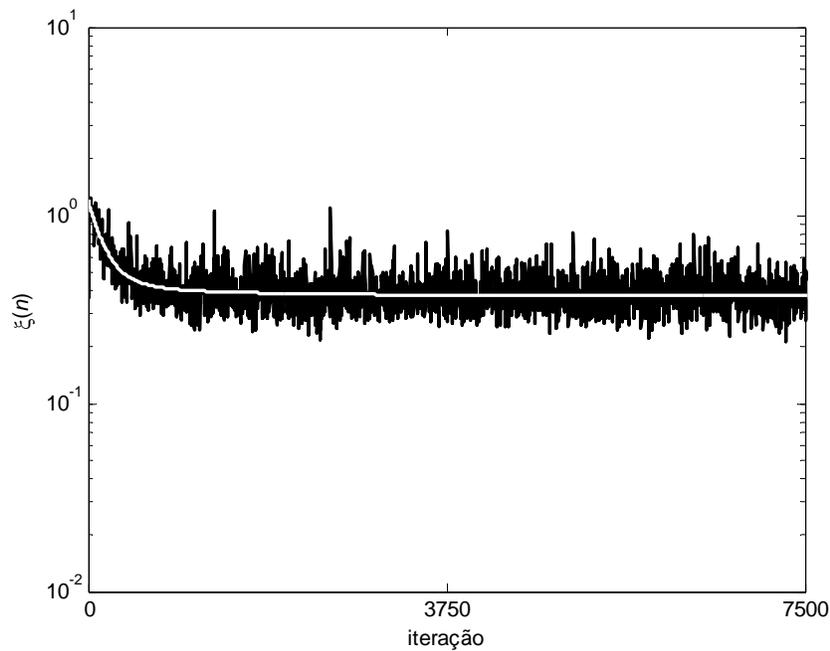


Figura 5.13. Exemplo 5.2. Evolução de  $\xi(n)$  com  $\mu = 0,125\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

*Exemplo 5.3:* Neste exemplo o filtro adaptativo utilizado é um filtro Volterra adaptativo de segunda ordem com atraso [14] e tamanho de memória igual a 11. Para esse caso, foi empregado um filtro com um atraso de duas amostras na entrada seguido de um bloco com tamanho de memória igual a 4, seguido ainda de mais um atraso de duas amostras e outro bloco com tamanho de memória igual a 3. Isso resulta em restrições nos pesos dependentes de  $x(n)$ ,  $x(n-1)$ ,  $x(n-6)$  e  $x(n-7)$ , como discutido na Seção 5.7.3. A planta possui os seguintes vetores de coeficientes:

$$\mathbf{h}_1 = [0,1 \ 0,15 \ 0,35 \ 0,5 \ 0,37 \ 0,28 \ 0,2 \ 0,1 \ 0,2 \ 0,4 \ 0,3],$$

$$\mathbf{h}_2 = [-0,13 \ 0,025 \ 0,22 \ -0,08 \ 0,08 \ -0,05 \ 0,06 \ 0,1 \ -0,05 \ -0,04 \ 0,08 \\ 0,17 \ -0,06 \ -0,04 \ 0,05 \ 0,03 \ 0,11 \ 0,006 \ 0,14 \ -0,005 \ -0,16 \ 0,10 \\ 0,25 \ 0,15 \ 0,10 \ 0 \ -0,2 \ 0,082 \ -0,07 \ -0,18 \ 0,03 \ 0,17 \ 0,09 \ 0,25 \\ 0,231 \ -0,221 \ -0,07 \ 0,025 \ -0,12 \ 0,05 \ -0,23 \ 0,03 \ 0,10 \ 0,23 \ 0,13 \\ 0,12 \ -0,034 \ 0,07 \ 0,15 \ -0,21 \ 0,22 \ 0,21 \ 0,05 \ -0,12 \ 0,18 \ 0 \ 0,11 \\ -0,04 \ 0,23 \ -0,21 \ 0,03 \ -0,10 \ 0,18 \ -0,08 \ 0,09 \ -0,03].$$

Foram realizadas simulações para diferentes valores do passo de adaptação e também com sinais Guassianos branco e colorido na entrada. O sinal colorido foi obtido através de um processo AR dado por  $x(n) = \alpha x(n-1) + \sqrt{1-\alpha^2} u(n)$ , onde  $u(n)$  é um ruído branco Gaussiano com variância unitária e  $\alpha = 0,5$ . Os valores de  $\mu$  utilizados são  $\mu_{\max}/5$  e  $\mu_{\max}/10$  com  $\mu_{\max} = 0,005$  para o filtro com ruído branco na entrada e  $\mu_{\max}/10$  com  $\mu_{\max} = 0,001$  para o filtro com ruído colorido de entrada. A Figura 5.14 mostra o comportamento médio de alguns dos pesos e a Figura 5.15 a curva de aprendizagem do filtro com sinal branco de entrada e passo de adaptação  $\mu_{\max}/5$ . Como podemos observar, os resultados obtidos são satisfatórios, apenas uma pequena diferença na curva de aprendizagem pode ser notada. Reduzindo-se o passo de adaptação para  $\mu_{\max}/10$  tal diferença tende a desaparecer, como podemos constatar nas Figuras 5.16 e 5.17. Nas Figuras 5.18 e 5.19 temos os resultados do filtro com sinal colorido na entrada e passo de adaptação  $\mu_{\max}/10$ . Muito bons resultados são obtidos, confirmando a validade das expressões desenvolvidas também para filtros com sinal colorido de entrada.

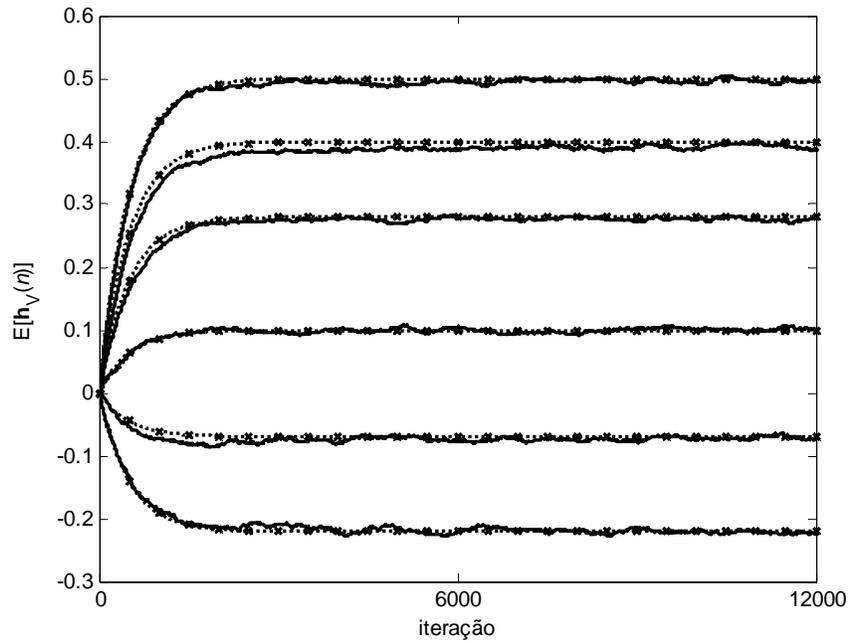


Figura 5.14. Exemplo 5.3. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0, 2\mu_{\max}$  e sinal branco de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

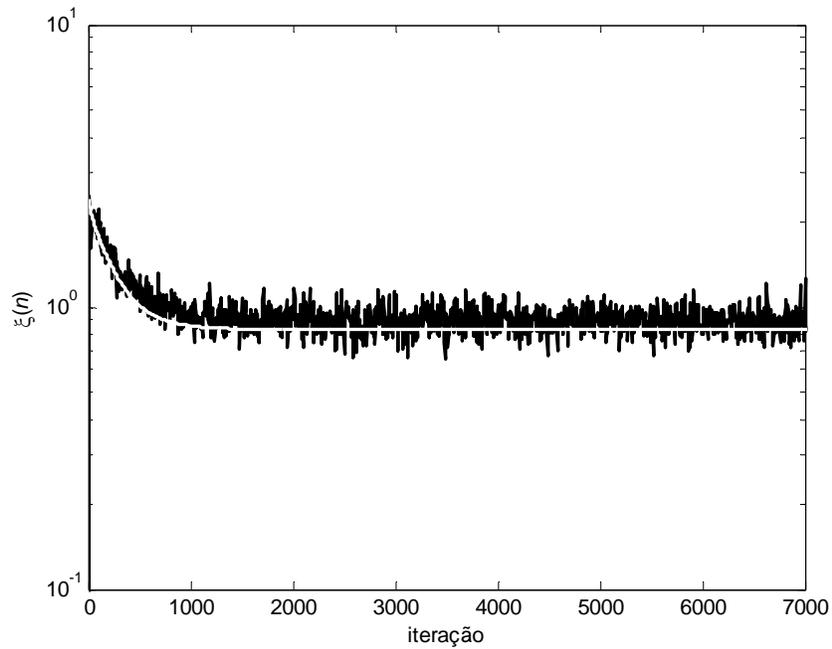


Figura 5.15. Exemplo 5.3. Evolução de  $\xi(n)$  com  $\mu = 0, 2\mu_{\max}$  e sinal branco de entrada. (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

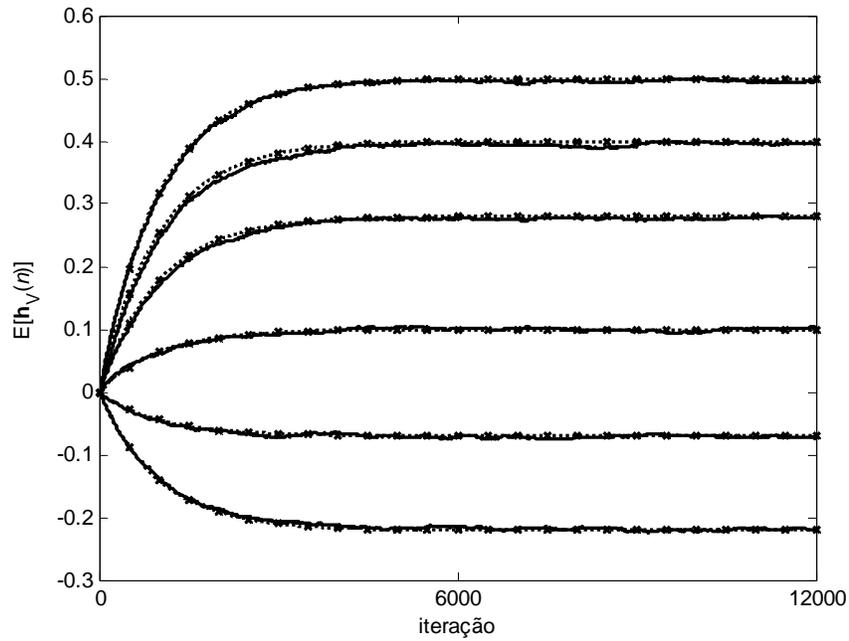


Figura 5.16. Exemplo 5.3. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0, 1\mu_{\max}$  e sinal branco de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

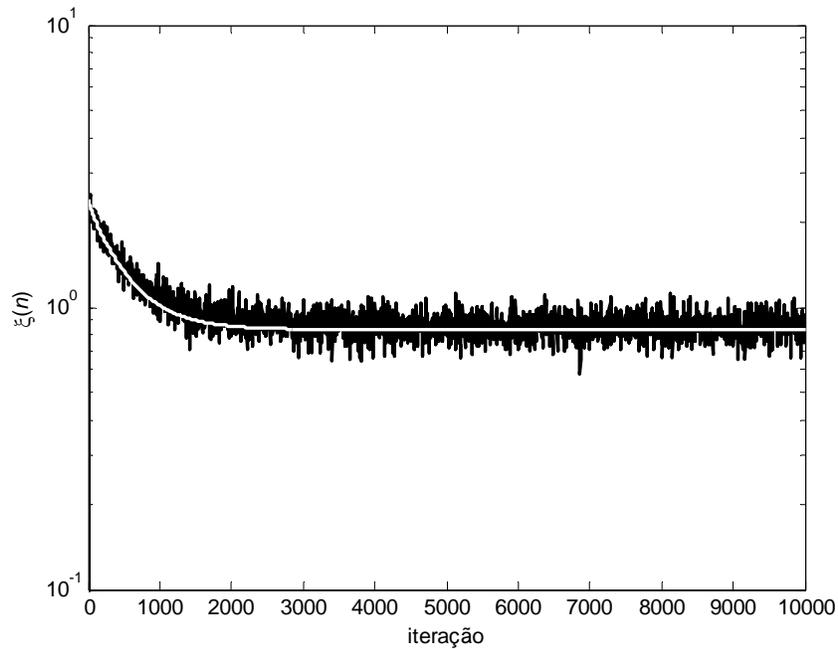


Figura 5.17. Exemplo 5.3. Evolução de  $\xi(n)$  com  $\mu = 0, 1\mu_{\max}$  e sinal branco de entrada. (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

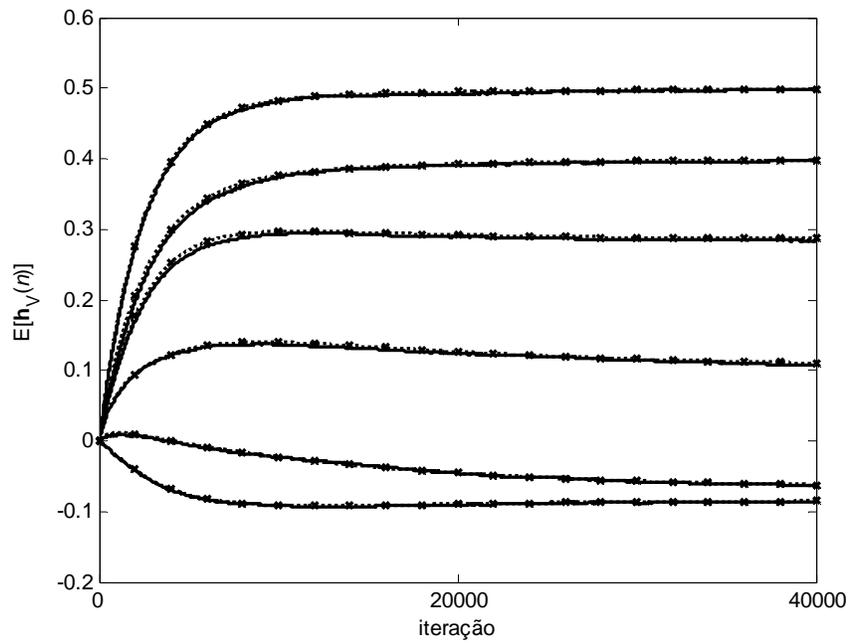


Figura 5.18. Exemplo 5.3. Evolução de  $E[\mathbf{h}_v(n)]$ , com  $\mu = 0,1\mu_{\max}$  e sinal colorido de entrada. (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

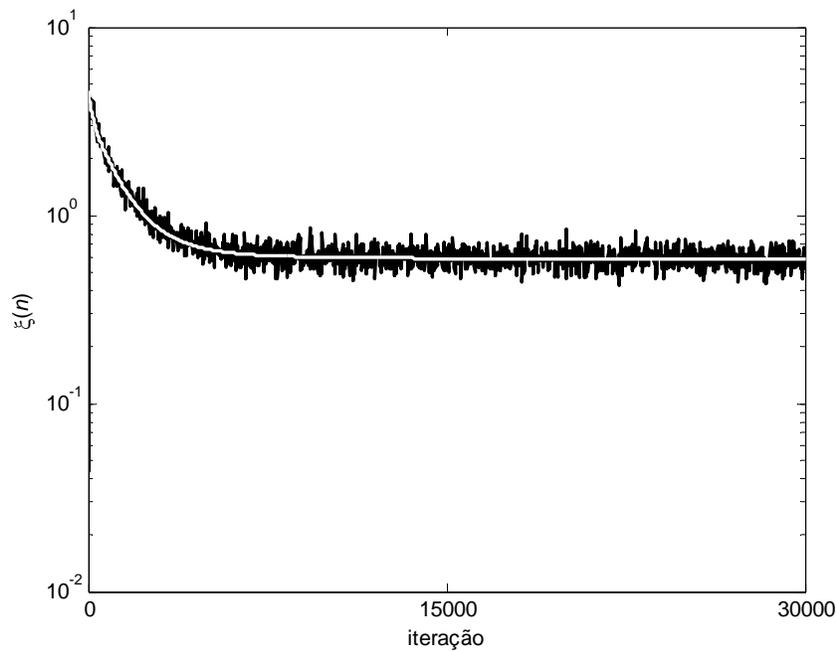


Figura 5.19. Exemplo 5.3. Evolução de  $\xi(n)$  com  $\mu = 0,1\mu_{\max}$  e sinal colorido de entrada. (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

*Exemplo 5.4:* Neste exemplo o filtro a ser analisado é um filtro Volterra adaptativo simplificado [15], com tamanho de memória igual a 5 e  $K = 2$ . Temos ainda uma planta Volterra, também com tamanho de memória igual a 5, com os vetores de coeficientes dados por

$$\mathbf{h}_1 = [1 \ 0,7 \ 0,4 \ 0,1 \ -0,1],$$

$$\mathbf{h}_2 = [0,4 \ 0,3 \ 0,05 \ -0,02 \ -0,04 \ -0,25 \ -0,3 \ -0,05$$

$$\quad -0,03 \ -0,1 \ -0,05 \ 0,01 \ 0,06 \ 0,05 \ 0,03].$$

O sinal de entrada é branco Gaussiano com variância unitária e o valor do passo de adaptação é  $\mu_{\max} / 10$  com  $\mu_{\max} = 0,01$ . A Figura 5.20 apresenta o comportamento médio de alguns dos pesos comparando o resultado obtido por simulação (média de 100 realizações) com o obtido das expressões desenvolvidas. Na Figura 5.21 temos a comparação da curva de aprendizagem obtida por simulação (média de 500 realizações) com os resultados teóricos. Os resultados apresentados novamente comprovam a eficiência das expressões aqui desenvolvidas.

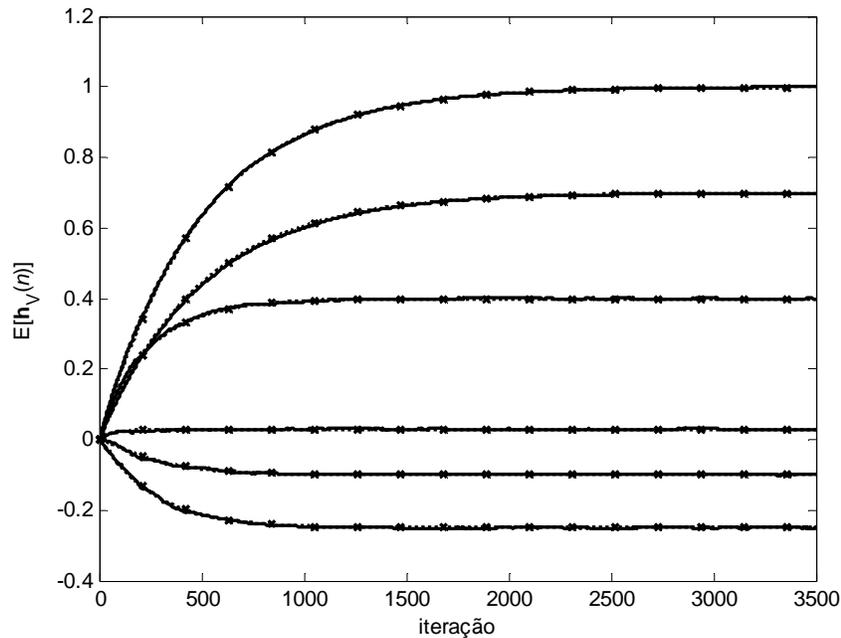


Figura 5.20. Exemplo 5.4. Evolução de  $E[\mathbf{h}_v(n)]$  com  $\mu = 0,1\mu_{\max}$ . (Linha sólida) simulação (média de 100 realizações); (Linha pontilhada com marcadores) modelo dado por (5.22).

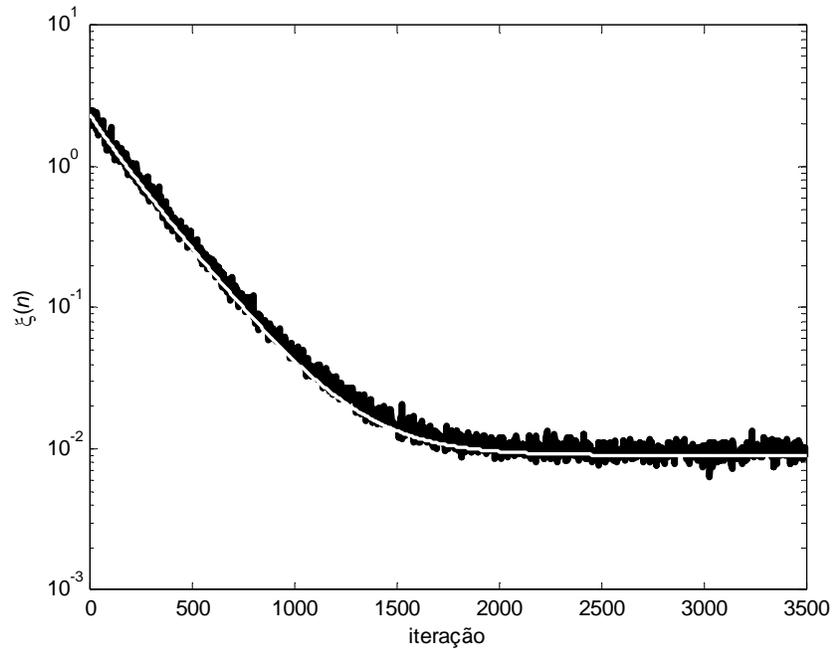


Figura 5.21. Exemplo 5.4. Evolução de  $\xi(n)$  com  $\mu = 0,1\mu_{\max}$ . (Linha preta) simulação (média de 500 realizações); (Linha branca) modelo dado por (5.27) e (5.31).

## 5.9 Considerações

Uma modelagem estatística do comportamento do filtro Volterra adaptativo com restrição nos pesos foi apresentada neste capítulo. A sua aplicação está relacionada às implementações de complexidade reduzida do filtro Volterra adaptativo que apresentam esparsidade em sua estrutura. Exemplos foram descritos e simulações foram realizadas para demonstrar a efetividade da modelagem aqui desenvolvida.

### Discussões e Conclusões

Neste trabalho, um estudo detalhado do filtro Volterra adaptativo foi apresentado. Os principais resultados obtidos são discutidos, neste capítulo, destacando também as contribuições alcançadas. Finalmente, algumas propostas de trabalhos futuros são elencadas.

#### 6.1 Sumário e Discussão dos Resultados

Após uma breve discussão introdutória sobre a base teórica da teoria de filtragem adaptativa, as principais características do filtro Volterra são discutidas no Capítulo 2. A primeira característica destacada é a linearidade entre os coeficientes e o sinal de saída deste filtro. Observando isso, é possível escrever a relação entrada/saída do filtro Volterra através de uma operação pseudolinear, permitindo, assim, tratar matematicamente o processo de filtragem com maior elegância e simplicidade. A segunda característica realçada é a possibilidade de decomposição em blocos do filtro Volterra. Do ponto de vista de implementação, grandes vantagens podem ser extraídas de tal característica. Uma delas é a possibilidade de eliminar blocos eventualmente desnecessários para algumas aplicações, com conseqüente redução de complexidade computacional. Ainda no Capítulo 2, considerações sobre complexidade computacional e número de coeficientes são discutidas.

No Capítulo 3, uma análise estatística do comportamento do filtro Volterra adaptado pelo algoritmo LMS é apresentada. A análise se baseia no princípio da pseudolinearidade da operação de filtragem do filtro Volterra, discutida no Capítulo 2. Essa característica permite a obtenção de expressões para o comportamento dos momentos de primeira e segunda ordens dos coeficientes do filtro bem como sua curva de aprendizagem. As aproximações consideradas para o desenvolvimento de tais expressões são também destacadas. Os resultados obtidos são validados, comparando-se as curvas obtidas através da teoria desenvolvida com as obtidas por simulações.

Como discutido no Capítulo 2, o filtro Volterra adaptativo requer uma grande quantidade de coeficientes mesmo para implementações com tamanho de memória relativamente pequeno. Assim, no Capítulo 4, são discutidas estruturas simplificadas para implementação do filtro Volterra adaptativo. Com isso, algumas soluções subótimas, levando a uma considerável redução no número de coeficientes, são obtidas. A primeira estrutura apresentada é fundamentada nos filtros adaptativos interpolados lineares [28]. Tais filtros envolvem um filtro adaptativo esparso em conjunto com um filtro interpolador de entrada. Já a segunda estrutura explora a decomposição em blocos do filtro Volterra. Assim, é possível obter uma estrutura com melhor desempenho e maior tolerância a plantas com baixa similaridade entre os coeficientes, além de possibilitar uma redução de complexidade bastante próxima à obtida com a primeira estrutura proposta. Resultados de simulação são apresentados, comparando o desempenho das estruturas propostas com a implementação convencional do filtro Volterra adaptativo. O desempenho das novas estruturas é também comparado com o filtro Volterra adaptativo simplificado [15]. Os resultados obtidos são bastante promissores do ponto de vista prático, principalmente quando limitações de complexidade são consideradas.

No Capítulo 5, uma análise do filtro Volterra adaptado pelo algoritmo LMS com restrição nos pesos é apresentada. O foco dessa análise é lidar com filtros de topologia esparsa, como o caso das estruturas baseadas em filtros Volterra interpolados e parcialmente interpolados, filtros Volterra com atraso [14] e, ainda, filtros Volterra simplificados [15]. Expressões para os momentos de primeira e segunda ordens são apresentadas e sua utilização é descrita. De maneira similar ao Capítulo 3, a validade do modelo obtido é verificada através de simulação.

## 6.2 Contribuições

O trabalho desenvolvido inclui, como primeira contribuição, uma análise estatística do filtro Volterra adaptado pelo algoritmo LMS. O modelo desenvolvido descreve o comportamento dos momentos de primeira e segunda ordens, considerando o uso de algumas aproximações matemáticas. Também são apresentadas duas estruturas para implementação do filtro Volterra adaptativo com redução no número de coeficientes. Esforços nesse sentido são relevantes, visto que a alta complexidade computacional

requerida é um dos principais problemas enfrentados na implementação de filtros Volterra adaptativos. Apesar de existirem outras iniciativas nesse sentido, nenhuma delas utiliza a abordagem aqui adotada, nem apresenta resultados tão promissores. Além dessas contribuições, uma análise estatística do filtro Volterra adaptado pelo algoritmo LMS com restrição nos pesos é desenvolvida, com o objetivo de proporcionar mais uma ferramenta para lidar com tais estruturas de filtragem.

### 6.3 Propostas para Trabalhos Futuros

Muitos trabalhos de pesquisa ainda podem ser desenvolvidos sobre este tema com o objetivo de aperfeiçoar a base teórica e proporcionar novos algoritmos para implementação de filtros adaptativos não-lineares, como é o caso do filtro Volterra. Como exemplos, podemos citar:

- Aprimoramento dos modelos que descrevem o comportamento médio dos pesos e o erro quadrático médio, com o objetivo de lidar com filtros que empreguem passos de adaptação maiores;
- Estudo de outras estruturas que usam interpolação e esparsidade para redução de complexidade computacional, como por exemplo, o caso de um arranjo semelhante ao filtro Volterra adaptativo X-filtrado [4];
- Busca de novas estratégias para redução da complexidade computacional do filtro Volterra adaptativo;
- Estudo do emprego de outros algoritmos adaptativos associados ao filtro Volterra;
- Estudo de outras estruturas adaptativas não-lineares.

## Princípio da Ortogonalidade

Neste apêndice, uma breve demonstração do princípio da ortogonalidade é apresentada. Tal princípio estabelece que o vetor de entrada e o sinal de erro, quando os coeficientes atingem seu valor ótimo, são descorrelacionados. Assim,

$$E[\mathbf{x}_v(n)e_o(n)] = 0. \quad (\text{A.1})$$

Da mesma forma como foi mostrado na Seção 3.5, definimos o sinal de erro ótimo de um filtro Volterra adaptativo, aplicado à estimação de um sinal (ver Figura 3.1), por

$$e_o(n) = d(n) + z(n) - \mathbf{h}_{v_o}^T \mathbf{x}_v(n), \quad (\text{A.2})$$

onde  $d(n)$  é o sinal a ser estimado,  $\mathbf{x}_v(n)$  o vetor de entrada do filtro Volterra,  $\mathbf{h}_{v_o}$  o vetor de coeficientes ótimo e  $z(n)$  um ruído aditivo descorrelacionado dos demais sinais do sistema. Multiplicando-se por  $\mathbf{x}_v(n)$  os dois lados de (A.2) e aplicando-se o operador esperança, obtemos

$$E[\mathbf{x}_v(n)e_o(n)] = E[\mathbf{x}_v(n)d(n)] + E[\mathbf{x}_v(n)z(n)] - E[\mathbf{x}_v(n)\mathbf{h}_{v_o}^T \mathbf{x}_v(n)]. \quad (\text{A.3})$$

O primeiro elemento do lado direito de (A.3) corresponde ao vetor de correlação cruzada entre o vetor de entrada e o sinal desejado, definido por

$$\mathbf{p}_v = E[\mathbf{x}_v(n)d(n)]. \quad (\text{A.4})$$

O segundo elemento é resolvido assumindo-se que o sinal de ruído possua média zero e é descorrelacionado dos demais sinais do sistema, resultando em

$$E[\mathbf{x}_v(n)z(n)] = 0. \quad (\text{A.5})$$

Manipulando-se o terceiro elemento do lado direito de (A.3) e aplicando-se a definição da matriz de autocorrelação de entrada  $\mathbf{R}_{vv} = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]$ , obtemos

$$E[\mathbf{x}_v(n)\mathbf{h}_{v_o}^T \mathbf{x}_v(n)] = E[\mathbf{x}_v(n)\mathbf{x}_v^T(n)]\mathbf{h}_{v_o} = \mathbf{R}_{vv}\mathbf{h}_{v_o}. \quad (\text{A.6})$$

Com os resultados de (A.4), (A.5) e (A.6), podemos reescrever (A.3) como segue:

$$E[\mathbf{x}_v(n)e_o(n)] = \mathbf{p}_v - \mathbf{R}_{vv}\mathbf{h}_{v_o}. \quad (\text{A.7})$$

De (3.5), equação de Wiener-Hopf para o filtro Volterra, obtém-se

$$\mathbf{p}_v = \mathbf{R}_{vv} \mathbf{h}_{v_0}. \quad (\text{A.8})$$

Através da substituição de (A.8) em (A.7) tem-se, então, a equação que define o princípio da ortogonalidade, dada por

$$E[\mathbf{x}_v(n)e_o(n)] = 0. \quad (\text{A.9})$$

Podemos encontrar ainda em [5], uma outra maneira de representar o princípio da ortogonalidade.

## Referências Bibliográficas

- [1] IEEE History Center. **Legacies: Bernard Widrow**. Disponível em <[http://www.ieee.org/organizations/history\\_center/legacies/widrow.html](http://www.ieee.org/organizations/history_center/legacies/widrow.html)>. Acesso em 24 mar. 2004.
- [2] Widrow B.; Stearns, S. D. **Adaptive Signal Processing**. Prentice-Hall, 1985.
- [3] Brening, C. et al. Acoustic Echo Control – An Application of Very-High-Order Adaptive Filters. **IEEE Signal Processing Magazine**, v. 16, n. 4, p. 42-69, Jul. 1999.
- [4] Kuo, S. M.; Morgan, D. R. **Active Noise Control Systems: Algorithms and DSP Implementations**. John Wiley & Sons, 1996.
- [5] Farhang-Boroujeny, B. **Adaptive Filters Theory and Applications**. John Wiley & Sons Ltd., 1999.
- [6] Haykin, S. **Adaptive Filter Theory**. Prentice Hall, 2001.
- [7] Oppenheim, A. V.; Willsky, A. S. **Signals & Systems**. Prentice Hall, 1997.
- [8] Mathews, V. J. Adaptive polynomial filters. **IEEE Signal Processing Magazine**, v. 8, p. 10-26, Jul. 1991.
- [9] Goh, S. L.; Mandic, D. P.; Bozic M. A neural nonlinear adaptive filter with a trainable activation function. In: 6TH SEMINAR ON NEURAL NETWORK APPLICATIONS IN ELECTRICAL ENGINEERING. Sep. 2002. **Proceedings...** p. 7-10.

- 
- [10] Hegde, V.; et. Al. Series-cascade nonlinear adaptive filters. In: 45TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS. Aug 2002. **Proceedings...** v. 3, p. 219-222.
- [11] Guérin, A.; Faucon, G.; Bouquin-Jeannès, R. L. Nonlinear acoustic echo cancellation based on Volterra filters. **IEEE Transactions on Speech and Audio Processing**, v. 11, n. 6, Nov. 2003.
- [12] Tan, L.; Jiang, J. Adaptive Volterra filters for active control of nonlinear noise processes. **IEEE Transactions on Signal Processing**, v. 49, n. 8, p. 1667-1676, Aug. 2001.
- [13] Neuvo, Y.; Dong, C. Y.; Mitra, S. K. Interpolated finite impulse response digital filters. **IEEE Transactions on Acoustics, Speech, Signal Processing**, v. ASSP-32, p. 563-570, Jun. 1984.
- [14] Tan, L.; Jiang, J. System modeling using a second-order Volterra delay filter. In: IEEE 39TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS. Aug 1996. **Proceedings...** v. 3, p. 18-21.
- [15] Fermo, A.; Carini, A.; Sicuranza, G. L. Simplified Volterra Filters for Acoustic Echo Cancellation in GSM Receivers. In: EUROPEAN SIGNAL PROCESSING CONFERENCE. Sep. 2000. **Proceedings...**
- [16] Volterra, V. **Theory of Functionals**. Dover Publications Inc., 1959.
- [17] Wiener, N. **Response of a Non-Linear Device to Noise**. Report no. 129, Radiation Laboratory, M.I.T., Cambridge - MA, Apr. 1942.
- [18] Stenger, A.; Trautmann, L.; Rabenstein, R. Nonlinear acoustic echo cancellation with second order adaptive Volterra filters. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING. Mar. 1999. **Proceedings...** v. 2, p. 15-19.

- 
- [19] Gutierrez, A.; Ryan, W. E. Performance of adaptive Volterra equalizers on nonlinear satellite channels. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS. Jun. 1995. **Proceedings...** v. 1, p. 19-22.
- [20] Clarkson, P. M.; Dokic, M. V. Stability and convergence behavior of second order LMS Volterra filter. **Electronic Letters**, v. 27, n. 5, p. 441-443, Feb. 1991.
- [21] Chao, J.; Inomata, A. A convergence analysis of Volterra adaptive filters. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. Jun. 1997. **Proceedings...** v. 4, p. 9-12.
- [22] Ogunfunmi, T.; Chang, S. L. Second order adaptive Volterra system identification based on discrete nonlinear Wiener model. **IEE Proceedings - Vision, Image and Signal Processing**, v. 148, n. 1, p. 21-29, Feb. 2001.
- [23] Batista, E.L.O.; Tobias, O.J.; Seara, R. Análise estatística de filtros Volterra adaptados pelo algoritmo LMS. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES. Set. 2004. Aceito para publicação.
- [24] Reed, M. J.; Hawksford, M. O. J. Efficient implementation of the Volterra filter. **IEE Proceedings - Vision, Image and Signal Processing**, v. 147, n. 2, p. 109-114, Apr. 2000.
- [25] Beck, E.; Seara, R. Estudo sobre a viabilidade de utilização de estruturas IFIR em filtros adaptáveis. In: 8º CONGRESSO BRASILEIRO DE AUTOMÁTICA. Set. 1990. **Anais...** v. 2, p. 1130-1135.
- [26] Seara, R.; Bermudez, J.C.M.; Beck, E. A new technique for the implementation of adaptive IFIR filters. In: INTERNATIONAL SYMPOSIUM ON SIGNALS, SYSTEMS AND ELETRONICS. Set. 1992. **Proceedings...** v. 2, p. 644-647.
- [27] Abousaada, A.; Aboulnasr, T.; Steenaart, W. An echo tail canceller based on adaptive interpolated FIR filtering. **IEEE Transactions on Circuits and Systems - II**, v. 39, p. 409-416, Jul. 1992.

- 
- [28] Tobias, O. J.; Seara, R. Analytical Model for the Mean Weight Behavior of Adaptive Interpolated-FIR Filters using the Constrained Filtered LMS Algorithm. **The IEEE 2000 AS-SPCC**, p. 272-277, Oct. 2000.
- [29] Tobias, O. J.; Seara, R. Analytical Model for the First and Second Moments of an Adaptive Interpolated FIR Filter using the Constrained Filtered-X LMS Algorithm. **IEE Proceedings – Vision, Image and Signal Processing**, v. 148, n. 5, p. 337-347, Oct. 2001.
- [30] Batista, E.L.O.; Tobias, O.J.; Seara, R. Filtros Volterra adaptativos: interpolado e parcialmente interpolado. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES. Set. 2004. Aceito para publicação.
- [31] Frost III, O.L. An Algorithm for Linearly Constrained Adaptive Array Processing. **Proceedings of the IEEE**, v. 60, n. 8, p. 926-935, Aug. 1972.
- [32] Wah, B. W.; Wu, Z. **The Theory of Discrete Lagrange Multipliers for Nonlinear Discrete Optimization.** Disponível em <http://www.manip.crhc.uiuc.edu/Wah/papers/C124/C124.ps.gz>. Acesso em 24 Mar. 2004.