

Filtros Volterra Adaptativos: Interpolado e Parcialmente Interpolado

Eduardo L. O. Batista, Orlando J. Tobias e Rui Seara

Resumo—Este trabalho apresenta dois algoritmos simplificados para a implementação de filtros Volterra adaptativos. A idéia por trás da abordagem proposta é o uso de estruturas adaptativas esparsas para reduzir o número elevado de coeficientes do filtro Volterra convencional. Um filtro interpolador é então associado ao filtro esperso com o objetivo de recriar os coeficientes removidos. Adicionalmente, uma segunda estrutura também interpolada que explora uma decomposição paralela de blocos do filtro Volterra é proposta. Resultados de simulação são apresentados comparando o desempenho do algoritmo do filtro Volterra adaptativo convencional com aqueles dos algoritmos propostos.

Palavras-Chave—Filtragem adaptativa, filtros interpolados, filtro Volterra.

Abstract—This paper presents two simplified algorithms for implementing adaptive Volterra filters. The idea behind the proposed approach is the use of sparse adaptive structures to reduce the number of coefficients, which is the major drawback of adaptive Volterra filters. An interpolator is used together with the sparse filter in order to recreate the removed coefficients. In addition, a second interpolated structure is proposed which exploits a parallel block decomposition of the Volterra filter. Simulation results are presented comparing the ordinary adaptive Volterra implementation with the proposed ones for performance.

Index Terms—Adaptive filtering, interpolated filters, Volterra filter.

I. INTRODUÇÃO

O uso de estruturas adaptativas lineares para lidar com problemas de filtragem não-linear têm sido uma prática muito comum. Isso se deve ao fato de que ferramentas simples e bem estabelecidas para filtragem adaptativa linear são bastante difundidas na literatura. Além disso, uma solução não-linear geralmente apresenta alta complexidade computacional associada a uma maior complexidade matemática. Entretanto, devido ao aumento da capacidade de processamento dos DSPs (*digital signal processor*) atualmente disponíveis, os filtros adaptativos não-lineares têm recebido grande atenção nos últimos anos. Nesse contexto, os filtros Volterra adaptativos têm se tornado uma opção interessante para lidar com diversas aplicações, tais como controle ativo de processos não-lineares [1], cancelamento de

eco acústico [2], identificação e redução de distorções em sistemas de alto-falantes [3], e equalização de canais de comunicação em satélites [4]. Contudo, os filtros Volterra ainda apresentam um grande problema: o número total de coeficientes cresce exponencialmente com o aumento do tamanho da memória do filtro, restringindo severamente sua aplicação. Algumas alternativas têm sido propostas para superar tal problema. Como exemplos de implementação que apresentam complexidade reduzida, tem-se: filtros Volterra no domínio da frequência [5], filtros Volterra com atraso [6], e filtros Volterra simplificados [7]. Contribuindo nesse sentido, o presente trabalho propõe duas estruturas Volterra adaptativas com complexidade reduzida. A idéia é utilizar um filtro Volterra esperso 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 [8], [9], [10]. 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.

Este trabalho está organizado como descrito a seguir. A Seção II apresenta uma breve descrição do filtro Volterra e sua estrutura em blocos. As Seções III e IV descrevem as estruturas do filtro Volterra adaptativo interpolado e do filtro Volterra adaptativo parcialmente interpolado, respectivamente. Na Seção V, são mostrados resultados de simulação comparando o desempenho do filtro FIR linear adaptativo, do filtro Volterra adaptativo convencional e do filtro Volterra adaptativo simplificado com aqueles das estruturas propostas neste trabalho. Finalmente, as conclusões finais do artigo são apresentadas na Seção VI.

II. FILTRO VOLTERRA

A relação de entrada e saída de um filtro Volterra causal é dada por [11]:

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

Eduardo L. O. Batista, Orlando J. Tobias e Rui Seara, LINSE - Laboratório de Circuitos e Processamento de Sinais, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, SC, E-mails: {dudu, orlando, seara}@linse.ufsc.br.

Este trabalho foi parcialmente financiado pela CAPES e pelo CNPq.

onde $x(n)$ e $y(n)$ caracterizam o sinal de entrada e o sinal de saída, respectivamente; $h_p(m_1, \dots, m_p)$ denota os coeficientes de ordem p ; N representa o tamanho da memória e P a ordem do filtro. É importante ressaltar ainda que, sem perda de generalidade, os elementos redundantes foram removidos nesta expressão [1].

Podemos observar a partir de (1) que o filtro Volterra pode ser entendido como uma composição de um bloco de primeira ordem (linear), cujos coeficientes são representados por h_1 , seguido por outros blocos não-lineares (h_2, h_3, \dots, h_p). Denotando a saída do bloco de ordem p por $y_p(n)$, podemos reescrever (1) como segue:

$$y(n) = \sum_{p=1}^P y_p(n), \quad (2)$$

com $y_p(n)$ descrito por

$$y_p(n) = \sum_{m_1=0}^{N-1} \sum_{m_2=m_1}^{N-1} \dots \sum_{m_p=m_{p-1}}^{N-1} h_p(m_1, m_2, \dots, m_p) \times \prod_{k=1}^p x(n-m_k). \quad (3)$$

Nota-se então que o filtro Volterra pode ser decomposto em uma estrutura paralela de blocos, como ilustrado na Figura 1, onde cada um dos blocos corresponde a um filtro de ordem p . Essa decomposição é bastante vantajosa do ponto de vista prático. 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.

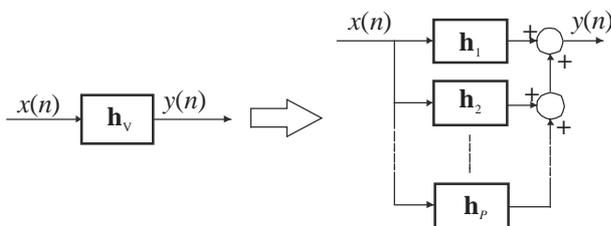


Fig. 1. Decomposição do filtro Volterra em uma estrutura de blocos.

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

$$y_1(n) = \mathbf{h}_1^T \mathbf{x}_1(n), \quad (4)$$

com $\mathbf{x}_1(n)$ representando o vetor de entrada de primeira ordem e \mathbf{h}_1 o vetor de coeficientes. Para os demais blocos, isso é possível considerando-se uma operação vetorial pseudo-linear. Por exemplo, para um bloco de segunda ordem pode-se escrever o vetor de entrada e o de coeficiente como segue:

$$\mathbf{x}_2(n) = [x^2(n) \ x(n)x(n-1) \ \dots \ x(n)x(n-N+1) \ x^2(n-1) \ \dots \ x^2(n-N+1)]^T, \quad (5)$$

$$\mathbf{h}_2 = [h_2(0,0) \ h_2(0,1) \ \dots \ h_2(0,N-1) \ h_2(1,1) \ \dots \ h_2(N-1,N-1)]^T, \quad (6)$$

resultando em uma relação de entrada e saída dada por

$$y_2(n) = \mathbf{h}_2^T \mathbf{x}_2(n). \quad (7)$$

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

$$y_p(n) = \mathbf{h}_p^T \mathbf{x}_p(n). \quad (8)$$

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

$$\mathbf{x}_v(n) = [\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \dots, \mathbf{x}_p^T(n)]^T, \quad (9)$$

e o vetor de coeficientes por

$$\mathbf{h}_v = [\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_p^T]^T, \quad (10)$$

tem-se a relação de entrada e saída do filtro Volterra (1) reescrita da seguinte forma:

$$y(n) = \mathbf{h}_v^T \mathbf{x}_v(n). \quad (11)$$

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 (5) e (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) 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 [1]:

$$D_p(N) = \frac{(N+p-1)!}{(N-1)!p!}. \quad (12)$$

Além disso, considerando $D_v(N, P)$ como o número total de coeficientes de um filtro Volterra de ordem P e tamanho de memória N , o número total de coeficientes é dado pelo somatório dos coeficientes de cada um dos blocos. Assim,

$$D_V(N, P) = \sum_{p=1}^P D_p(N), \quad (13)$$

o que resulta em

$$D_V(N, P) = \frac{(N+P)!}{N!P!} - 1. \quad (14)$$

III. FILTRO VOLTERRA ADAPTATIVO INTERPOLADO

Como mostrado na Seção II, reduzindo-se o tamanho da memória de um filtro Volterra, obtém-se uma redução exponencial do número de coeficientes. Isso pode ser obtido através do uso de um filtro Volterra adaptativo esparso em conjunto com um filtro interpolador de entrada. Assim, uma considerável redução de complexidade é obtida uma vez que o filtro interpolador associado requer apenas algumas operações por amostra $[O(M)]$, com $M < 10$. A Figura 2 mostra a estrutura proposta para o filtro Volterra adaptativo interpolado, usando o algoritmo LMS no processo de adaptação. Nesta figura, \mathbf{h}_{vi} representa o filtro Volterra adaptativo esparso, enquanto o interpolador é representado por um filtro FIR $\mathbf{I} = [i_0 \ i_1 \ \dots \ i_{M-1}]^T$, com M coeficientes. O sinal de entrada e o sinal de entrada interpolado são denotados por $x(n)$ e $x_i(n)$, respectivamente, onde

$$x_i(n) = \sum_{j=0}^M i_j x(n-j). \quad (15)$$

O sinal desejado $d(n)$ é somado a um ruído de medição $z(n)$ de média zero e decorrelacionado dos demais sinais do sistema. Os sinais $y(n)$ e $e(n)$ representam o sinal de saída do filtro Volterra adaptativo esparso e o sinal de erro, respectivamente. Essa estrutura é similar à dos filtros IFIR adaptativos discutidos em [8] e [9].

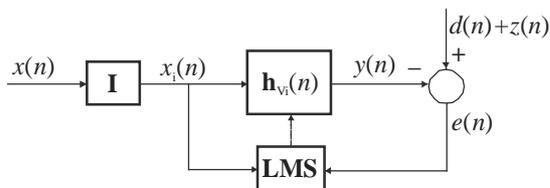


Fig. 2. Diagrama de blocos do filtro Volterra adaptativo interpolado.

A escolha do algoritmo LMS para o processo de adaptação é baseada na sua simplicidade e robustez [12]. Assim, usando esse algoritmo, os coeficientes do filtro Volterra são atualizados da seguinte forma:

$$\mathbf{h}_{vi}(n+1) = \mathbf{h}_{vi}(n) + 2\mu e(n)\mathbf{x}_{vi}(n), \quad (16)$$

onde $\mathbf{x}_{vi}(n)$ é o vetor de entrada do filtro Volterra, dado por (9), para o sinal de entrada interpolado. Na expressão de atualização, é possível ainda usar valores diferentes de μ para os diversos blocos do filtro Volterra, com o objetivo de melhorar o desempenho geral do sistema.

Como em [8], L é usado para definir o fator de interpolação. Ele determina o grau esparsidade do filtro Volterra. A entrada do filtro esparso é obtida retirando $(L-1)$ amostras de cada L amostras consecutivas do sinal de

entrada original. Por exemplo, o vetor de entrada de primeira ordem possui a seguinte estrutura:

$$\mathbf{x}_{i1}(n) = [x_1(n) \ x_1(n-L) \ x_1(n-2L) \ \dots \ x_1(n-N+1)]^T. \quad (17)$$

Com isso, o tamanho de memória do filtro esparso, representado por N_i , fica reduzido a

$$N_i = \left\lfloor \frac{N-1}{L} \right\rfloor + 1, \quad (18)$$

onde $\lfloor \cdot \rfloor$ denota a operação de truncamento.

A Tabela 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 essas estruturas.

TABELA 1
COMPLEXIDADE: FILTRO VOLTERRA ADAPTATIVO
EM COMPARAÇÃO COM SUA VERSÃO INTERPOLADA

N	P	L	N_i	$D_V(N, P)$	$D_V(N_i, P)$	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, esta redução da quantidade de coeficientes pode ser insuficiente, considerando que estamos lidando com uma solução subótima para problemas com plantas Volterra. Além disso, é bem conhecido que filtros interpolados não apresentam bom desempenho para a modelagem de plantas com coeficientes poucos correlacionados [8], [9]. Para esse caso, uma solução mais apropriada pode ser o uso de um filtro Volterra adaptativo parcialmente interpolado.

IV. 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 assim, 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 3 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 3 são equivalentes aos da Figura 2.

Para a estrutura da Figura 3, os coeficientes são atualizados da seguinte forma:

$$\begin{aligned} \mathbf{h}_1(n+1) &= \mathbf{h}_1(n) + 2\mu_1 e(n)\mathbf{x}_1(n) \\ \mathbf{h}_{2i}(n+1) &= \mathbf{h}_{2i}(n) + 2\mu_2 e(n)\mathbf{x}_{2i}(n), \end{aligned} \quad (19)$$

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 $\mathbf{x}_i(n)$ (17). Os passos de atualização do algoritmo adaptativo para o bloco de primeira ordem e para o bloco esparsa de segunda ordem são dados por μ_1 e μ_2 , respectivamente.

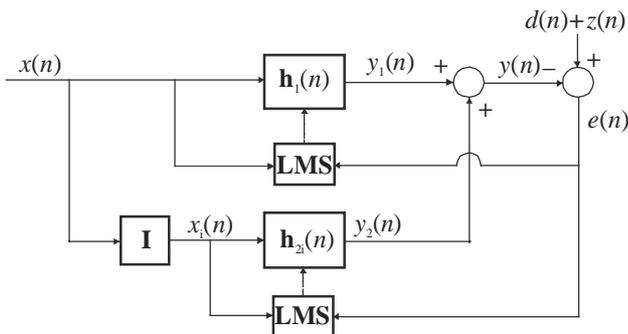


Fig. 3. Diagrama de blocos do filtro Volterra adaptativo parcialmente interpolado.

O número total de coeficientes para este filtro é determinado somando o número de coeficientes para cada um dos blocos, obtido através de (12). É importante lembrar que o tamanho de memória dos blocos interpolados é reduzido de acordo com (18). Na Tabela 2, as diferentes estruturas são comparadas em número de coeficientes requerido. 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 2

NÚMERO DE COEFICIENTES PARA AS DIFERENTES ESTRUTURAS COM $L = 2$				
memória	ordem	número de coeficientes		
		Volterra	interpolado	parc. interp.
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 coeficientes pouco correlacionados do que a versão totalmente interpolada. Isso ocorre pois a parte linear é modelada integralmente, sendo que a esparsidade é apenas considerada na parte não-linear. Assim, a estrutura da Figura 3, 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.

V. 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 com um filtro Volterra adaptativo convencional e um filtro FIR linear adaptativo, em relação ao comportamento do erro quadrático médio (EQM). Nos

exemplos que apresentam filtros de segunda ordem, também são realizadas comparações com implementações do filtro Volterra adaptativo simplificado [7]. Esse último consiste de uma implementação esparsa do filtro Volterra adaptativo de segunda ordem que despreza os coeficientes mais distantes da diagonal principal, uma vez que esses coeficientes, em muitas aplicações práticas, são de menor importância [7]. Isso pode ser melhor observado considerando-se a relação de entrada e saída do bloco de segunda ordem, dada por

$$y_2(n) = \sum_{k=0}^K \sum_{m=0}^{N-k-1} h_2(m, m+k) x(n-m) x(n-m-k), \quad (20)$$

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 (20), tem-se ainda o fator K que determina a quantidade de elementos que é desprezada. Escolhendo-se $K = N - 1$ tem-se a implementação de um filtro Volterra completo. Reduzindo-se esse valor estaremos desprezando os elementos mais afastados da diagonal principal [7]. 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 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, com variância unitária. A escolha do passo de adaptação é feita levando-se em consideração um limite superior, denotado aqui por μ_{\max} , dado por [12]

$$\mu_{\max} = \frac{1}{3\text{tr}[\mathbf{R}]}, \quad (21)$$

onde \mathbf{R} é a matriz de auto-correlação do sinal de entrada e $\text{tr}[\cdot]$ corresponde ao operador traço da matriz. Apesar de (21) ser obtida de forma aproximada [12], tal limite é bastante próximo do que é observado 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 de EQM são obtidas através de simulação Monte Carlo (média de 200 realizações). A Figura 4 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 apresentam um desempenho comparável ao do filtro Volterra convencional, especialmente o do filtro Volterra adaptativo parcialmente interpolado. A Figura 5 mostra uma comparação entre as implementações do filtro Volterra adaptativo simplificado [7] 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 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 coeficientes pouco correlacionados, afetando sobremaneira o desempenho de uma abordagem interpolada. A não-linearidade sem memória e o sinal de entrada são os mesmos do Exemplo 1. O número de coeficientes e os valores do passo de adaptação são também idênticos aos do Exemplo 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 6 ilustra os resultados obtidos através de simulação Monte Carlo (média de 200 realizações). Como esperado, o filtro Volterra adaptativo interpolado apresenta um 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 IV. Através da comparação com o filtro Volterra simplificado da Figura 7, 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 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 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 8 apresenta as curvas do EQM obtidas. 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.

VI. CONCLUSÕES

Neste trabalho, são apresentadas estruturas simplificadas para a implementação de filtros Volterra adaptativos. O uso de uma abordagem interpolada permite 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.

REFERÊNCIAS

- [1] L. Tan and J. Jiang, "Adaptive Volterra Filters for Active Control of Nonlinear Noise Processes," *IEEE Trans. on Signal Processing*, vol. 49, no. 8, pp. 1667-1676, Aug. 2001.
- [2] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear Acoustic Echo Cancellation with Second Order Adaptive Volterra Filters," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 15-19, Mar. 1999.
- [3] M. Tsujikawa, T. Shiozaki, Y. Kajikawa, and Y. Nomura, "Identification and Elimination of Second-order Nonlinear Distortion of Loudspeaker Systems Using Volterra Filters," *IEEE Int. Symp. on Circ. and Systems*, vol. 5, pp. 28-31, May 2000.
- [4] A. Gutierrez and W. E. Ryan, "Performance of Adaptive Volterra Equalizers on Nonlinear Satellite Channels," *IEEE Int. Conf. on Communications*, vol. 1, pp. 19-22, Jun. 1995.
- [5] M. J. Reed and M. O. J. Hawksford, "Efficient Implementation of the Volterra Filter," *IEE Proc.-Vis. Image Signal Processing*, vol. 147, no. 2, pp. 109-114, Apr. 2000.
- [6] L. Tan and J. Jiang, "System Modeling Using a Second-order Volterra Delay Filter," *IEEE 39th Midwest Symposium on Circuits and Systems*, vol. 3, pp. 18-21, Aug. 1996.
- [7] A. Fermo, A. Carini, G. L. Sicuranza, "Simplified Volterra Filters for Acoustic Echo Cancellation in GSM Receivers," *Proc. EUSIPCO 2000*, Tampere, Finland, Sept. 2000.
- [8] R. Seara, J.C.M. Bermudez, E. Beck, "A new technique for the implementation of adaptive IFIR filters," *Proc. ISSSE 1992*, vol. 2, pp. 644-647, Set. 1992.
- [9] O. J. Tobias, R. Seara, "Analytical Model for the Mean Weight Behavior of Adaptive Interpolated-FIR Filters using the Constrained Filtered LMS Algorithm," *The IEEE 2000 AS-SPCC*, pp. 272-277, Oct. 2000.
- [10] A. Abousaada, T. Aboulnasr, and W. Steenaert, "An Echo Tail Canceller Based on Adaptive Interpolated FIR Filtering," *IEEE Trans. Circuits Syst.-II*, vol. 39, pp. 409-416, Jul. 1992.
- [11] V. J. Mathews, "Adaptive Polynomial Filters," *IEEE Signal Processing Magazine*, vol. 8, pp. 10-26, Jul. 1991.
- [12] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*, John Wiley & Sons Ltd., 1999.

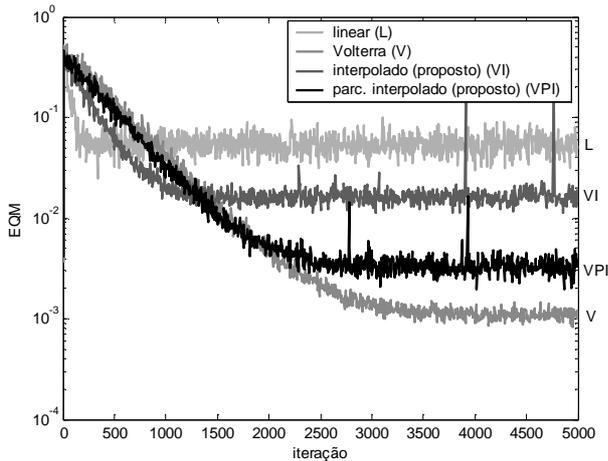


Fig. 4. Exemplo 1. Evolução do EQM (média de 200 realizações).

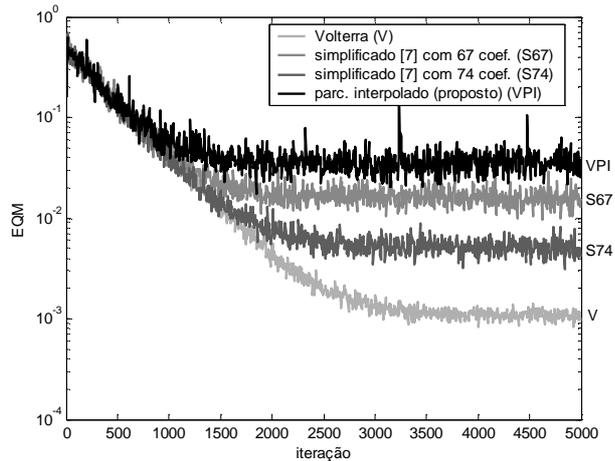


Fig. 7. Exemplo 2. Evolução do EQM (média de 200 realizações).

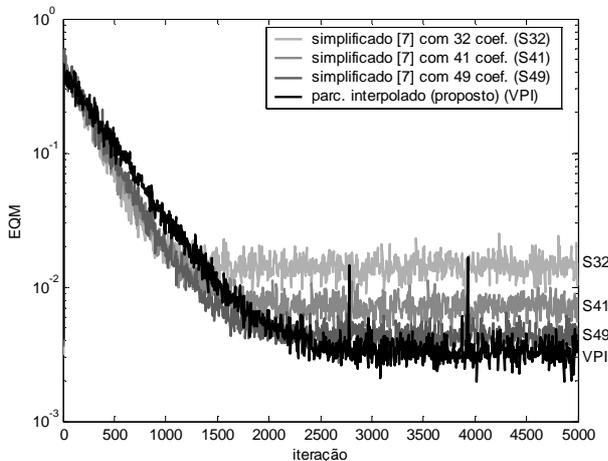
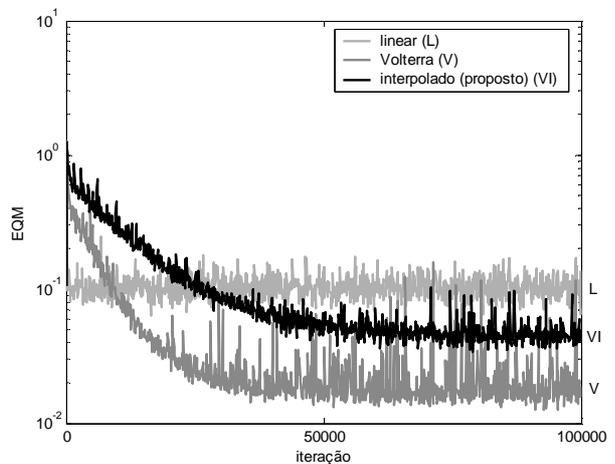


Fig. 5. Exemplo 1. Evolução do EQM (média de 200 realizações).



(a)

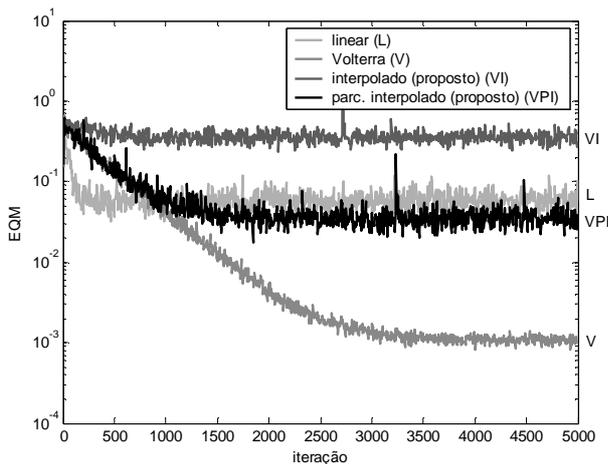
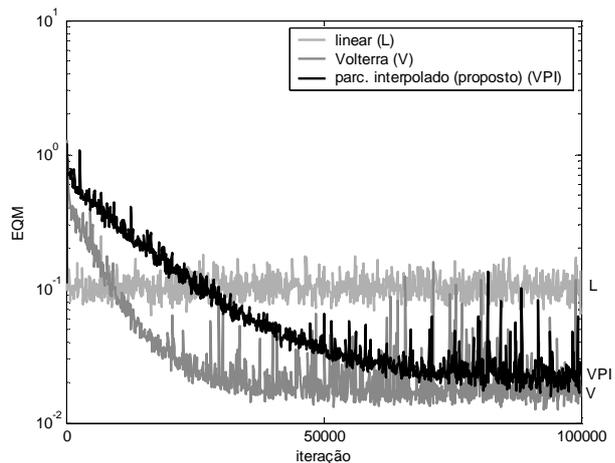


Fig. 6. Exemplo 2. Evolução do EQM (média de 200 realizações).



(b)

Fig. 8. Exemplo 3. Evolução do EQM (média de 200 realizações). Comparações: (a) Filtro Volterra interpolado. (b) Filtro Volterra parcialmente interpolado.