

Algoritmos Rápidos para a Implementação de Modelos Teóricos de Filtros Adaptativos

Márcio H. Costa, Guilherme H. Costa, José C. M. Bermudez e Fábio P. Itturriet

Resumo — Este trabalho apresenta um conjunto de algoritmos de baixo custo computacional para a implementação de modelos teóricos dos filtros adaptativos *Normalized Least Mean Square* (NLMS), *Least Mean Fourth* (LMF) e *Projeções Afins* (AP). Como resultado, a complexidade computacional originalmente de $\mathcal{O}(N^3)$ (N - número de coeficientes) decresce para $\mathcal{O}(N)$ sem perda de precisão.

Palavras-Chave — Filtragem adaptativa, modelo teórico, NLMS, LMF, AP, projeções afins, educação.

Abstract — This work presents a set of low computational cost theoretical models for the *Normalized Least Mean Square* (NLMS), *Least Mean Fourth* (LMF) and *Affine Projection* (AP) adaptive algorithms. As a result, the $\mathcal{O}(N^3)$ (N - number of taps) original computational complexity decreases to $\mathcal{O}(N)$ without lost in precision.

Index Terms — Adaptive filtering, theoretical model, NLMS, LMF, AP, Affine projection, education.

I. INTRODUÇÃO

Filtros adaptativos constituem uma importante ferramenta na solução de uma série de problemas na área de processamento de sinais, dentre eles: identificação de sistemas, equalização, controle ativo de ruído e cancelamento de eco [1-3]. Seu uso é especialmente indicado quando o conhecimento *a priori* sobre o sistema é reduzido ou as condições de operação são variantes no tempo.

Uma importante classe de filtros adaptativos é a chamada família LMS. Seu principal membro, o algoritmo *Least Mean Square*, foi desenvolvido por Widrow e Hoff em 1959 [4]. A partir deste uma série de outros algoritmos foram derivados, como por exemplo: LMS normalizado, *Leaky LMS*, *Least Mean Fourth* (LMF) e *Projeções Afins* (AP).

Ao longo dos anos, vários modelos teóricos têm sido desenvolvidos para prever o comportamento de algoritmos

adaptativos, em especial os da família LMS [1, 3, 5-11]. O objetivo principal de tais modelos é o de auxiliar no projeto de sistemas adaptativos para aplicações reais, possibilitando a predição do desempenho do algoritmo adaptativo em diferentes condições de operação.

Esses modelos são normalmente apresentados em formas analíticas que facilitam a compreensão das interações entre os diversos parâmetros do sistema e seus efeitos sobre o comportamento global do algoritmo. Desta forma, permitem o projeto otimizado do algoritmo a partir de estimativas das propriedades estatísticas dos sinais envolvidos.

Entretanto, durante o projeto, frequentemente são necessárias diversas simulações a partir dos modelos analíticos, seja para avaliar o desempenho do algoritmo em diferentes situações, seja para comparar os desempenhos de diferentes algoritmos. Nesses casos, a eficiência computacional do modelo assume grande importância. Modelos analíticos numericamente ineficientes podem requerer um esforço computacional maior do que a simulação estatística do comportamento do algoritmo.

Este trabalho apresenta versões computacionalmente eficientes de modelos teóricos de três algoritmos adaptativos, as quais não estão disponíveis na literatura. São derivados modelos para os algoritmos ε -NLMS (tendo o NLMS como caso particular), LMF e *Projeções Afins*, a partir dos modelos analíticos propostos em [8], [9] e [10], respectivamente. Os modelos apresentados são descrições exatas dos modelos originalmente propostos, mas possuem custo computacional significativamente reduzido.

A metodologia utilizada baseia-se na eliminação de operações redundantes, obtida através da decomposição da matriz de autocorrelação do sinal de entrada em termos de matrizes de autovalores e de autovetores, assim como da conseqüente reorganização das equações [1,3]. O resultado é a determinação de algoritmos cuja complexidade computacional é proporcional ao número de coeficientes do filtro adaptativo ($\mathcal{O}(N)$) em contraposição à implementação direta das equações fornecidas pela literatura que são proporcionais ao cubo do número de coeficientes ($\mathcal{O}(N^3)$).

A seção II discute a importância do uso de modelos teóricos pelo projetista. A seção III descreve os algoritmos adaptativos analisados e a seção IV apresenta os modelos eficientes para os diversos algoritmos. A seção V apresenta comparações entre as complexidades computacionais dos algoritmos originalmente propostos e de suas implementações eficientes. A seção VI apresenta as conclusões finais.

Márcio Holsbach Costa e Fábio Pires Itturriet, Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, Pelotas-RS. Guilherme Holsbach Costa e José Carlos Moreira Bermudez, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis-SC. E-mails: m.costa@ieee.org, holsbach@eel.ufsc.br, j.bermudez@ieee.org, fpit@pop.com.br.

II. ANÁLISE DE FILTROS ADAPTATIVOS

De forma geral, os parâmetros de controle de um filtro adaptativo (passo de convergência, fator de esquecimento, etc) são determinados em função de critérios de desempenho como velocidade de convergência e desajuste em regime permanente, e limitados por considerações de estabilidade. Esse procedimento visa a obtenção das condições ótimas, determinadas pelas especificações de projeto, associadas a uma margem de segurança (em decorrência da variabilidade das condições de operação). Para tanto, dois métodos de auxílio ao projeto são largamente utilizados: simulações Monte Carlo e modelos teóricos [1-3].

A. Simulações Monte Carlo

A análise de desempenho através de simulações é interessante visto que, em sua grande maioria, os membros da família LMS são simples e de baixo custo computacional. Entretanto, a aleatoriedade dos sinais envolvidos muitas vezes acarreta a necessidade de um grande número de realizações do processo para que o comportamento médio do sistema possa ser corretamente avaliado. Estudos demonstram que, para o caso de passos de convergência elevados, é necessário um grande número de promediações para obter-se uma aproximação confiável da curva de aprendizado [12]. Desta forma, é requerido um grande tempo de processamento, ou o uso de sistemas computacionais de alta velocidade (e por consequência de alto custo).

B. Modelos Teóricos

Modelos matemáticos (teóricos) permitem analisar de forma explícita as inter-relações entre os parâmetros de projetos e as diversas medidas de desempenho do algoritmo. Esses modelos, no entanto, nem sempre são computacionalmente eficientes. Assim, sua utilização pode acarretar uma elevada carga computacional, inviabilizando a análise exaustiva das diferentes possibilidades pelo projetista. Isso é especialmente importante em aplicações que requerem um grande número de coeficientes no filtro adaptativo, como é o caso de cancelamento de eco acústico [13].

Em determinadas situações, o tempo necessário para a avaliação do projeto usando o modelo teórico pode exceder o limite do tolerável, obrigando o projetista a recorrer apenas a simulações estatísticas e à interpretação do modelo. Desta forma, as características da aplicação podem limitar a utilidade da modelagem teórica na estimação de desempenho do algoritmo, dificultando o projeto dos parâmetros do filtro. Como consequência, modelos teóricos de baixo custo computacional são importantes para o projetista.

C. Equações Determinísticas Recursivas

Embora existam várias técnicas para a obtenção de modelos teóricos, uma metodologia largamente utilizada é o desenvolvimento de equações determinísticas recursivas para o comportamento do valor médio dos coeficientes do filtro e do erro médio quadrático com o tempo. Essa forma de

representação permite o estudo do comportamento médio do sistema adaptativo a cada instante de tempo.

III. ALGORITMOS ADAPTATIVOS

Considere o sistema descrito pelo diagrama apresentado na Fig. 1. Neste diagrama n é o instante de tempo discreto, $x(n)$ é o sinal de entrada, $d(n)$ é o sinal desejado, $y(n)$ é o sinal de cancelamento, $e(n)$ é o sinal de erro e $z(n)$ é o ruído de medição. $\mathbf{w}(n) = [w_0(n) w_1(n) \dots w_{N-1}(n)]^T$ é o vetor de coeficientes adaptativos e $\mathbf{w}^0 = [w_0^0 w_1^0 \dots w_{N-1}^0]^T$ é o vetor de coeficientes da resposta ao impulso da planta. Os vetores $\mathbf{w}(n)$ e \mathbf{w}^0 são considerados de dimensão N . O bloco *Adapt.* representa o algoritmo adaptativo utilizado.

O sinal de erro é dado pela expressão

$$e(n) = z(n) - \mathbf{v}^T(n) \mathbf{x}(n), \quad (1)$$

em que $\mathbf{x}(n) = [x(n) x(n-1) \dots x(n-N+1)]^T$ é o vetor de dados observados e $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}^0$ é o vetor erro dos coeficientes $\mathbf{v}(n) = [v_0(n) v_1(n) \dots v_{N-1}(n)]^T$.

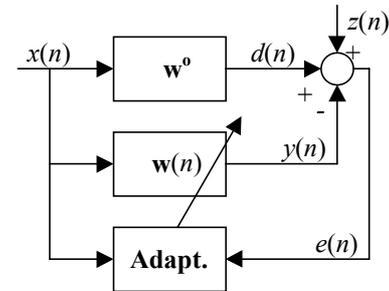


Fig. 1. Diagrama em blocos do sistema adaptativo abordado.

As equações de atualização dos coeficientes dos algoritmos abordados são, respectivamente:

A. ε -NLMS

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta e(n) \mathbf{x}(n)}{\varepsilon + \mathbf{x}^T(n) \mathbf{x}(n)} \quad (2)$$

B. LMF

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e^3(n) \mathbf{x}(n) \quad (3)$$

C. Projeções Afins

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n) \boldsymbol{\phi}(n)}{\boldsymbol{\phi}^T(n) \boldsymbol{\phi}(n)} \quad (4)$$

na qual $\boldsymbol{\phi}(n) = [\mathbf{I} - \mathbf{X}_P(n) [\mathbf{X}_P^T(n) \mathbf{X}_P(n)]^{-1} \mathbf{X}_P^T(n)] \mathbf{x}(n)$ e $\mathbf{X}_P(n) = [\mathbf{x}(n-1) \mathbf{x}(n-2) \dots \mathbf{x}(n-P)]^T$.

Os parâmetros β e μ são chamados de passo de convergência e são responsáveis pela velocidade do processo de adaptação. O parâmetro ε é a chamada constante de regularização e sua função é impedir divisões por zero.

IV. METODOLOGIA

O procedimento de simplificação utilizado neste trabalho baseia-se na decomposição da matriz de autocorrelação do sinal de excitação em termos de suas matrizes de autovalores e autovetores [1, pp. 263] [3, pp. 529], de tal forma que:

$$E\{\mathbf{x}(n)\mathbf{x}^T(n)\} = \mathbf{R}_0 = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \quad (5)$$

em que $E\{\cdot\}$ é a operação de valor esperado, $\mathbf{\Lambda}$ é matriz de autovalores e \mathbf{Q} é a matriz de autovetores associados a \mathbf{R}_0 . Então, em decorrência da ortogonalidade entre as colunas de \mathbf{Q} ,

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}. \quad (6)$$

Utilizando-se as equações (5) e (6) no equacionamento teórico dos algoritmos em questão, é possível substituir as equações matriciais recursivas por equações vetoriais, diminuindo assim a complexidade das operações necessárias para avaliar o comportamento do algoritmo.

Embora essa técnica seja conhecida e utilizada na simplificação do modelo teórico do algoritmo LMS, apenas recentemente, com a publicação de [8-10] é que os algoritmos ε -NLMS, LMF e Projeções Afins foram descritos em termos de equações determinísticas recursivas, de tal forma que esse artifício possa ser utilizado.

A seguir é apresentado o procedimento passo a passo para o caso do algoritmo ε -NLMS e, na seqüência, os resultados para os algoritmos LMF e Projeções Afins.

A. Algoritmo ε -NLMS

O comportamento médio dos coeficientes do algoritmo ε -NLMS para entrada gaussiana é dado por [8]

$$E\{\mathbf{v}(n+1)\} = [\mathbf{I} - \beta/(\varepsilon + Nr_0) \cdot \mathbf{R}_0] E\{\mathbf{v}(n)\}. \quad (7)$$

em que r_0 é a variância do sinal de excitação. Substituindo-se (5) em (7) e pré-multiplicando-se por \mathbf{Q}^T chega-se a

$$E\{\mathbf{v}(n+1)\} = [\mathbf{I} - \beta/(\varepsilon + Nr_0) \cdot \mathbf{\Lambda}] E\{\mathbf{v}(n)\}, \quad (8)$$

em que $E\{\mathbf{v}(n)\} = \mathbf{Q}^T E\{\mathbf{v}(n)\}$. O vetor $\mathbf{v}(n)$ é o vetor de erro nos coeficientes projetado no espaço determinado pelos autovetores de \mathbf{R}_0 . Dessa forma, o conjunto de equações dos coeficientes é desacoplado, permitindo que cada um de seus elementos seja determinado de forma independente dos demais. O erro médio quadrático do algoritmo ε -NLMS é descrito por

$$E\{e^2(n)\} = r_z + tr\{\mathbf{R}_0\mathbf{K}(n)\}. \quad (9)$$

em que r_z é a variância do ruído de medição $z(n)$, $tr\{\mathbf{A}\}$ o traço da matriz \mathbf{A} e $\mathbf{K}(n)$ é a matriz de correlação do vetor erro nos coeficientes. Substituindo-se (5) e (6) em (9) obtém-se

$$E\{e^2(n)\} = r_z + \boldsymbol{\lambda}^T \mathbf{p}(n), \quad (10)$$

em que $\boldsymbol{\lambda} = [\lambda_0 \lambda_1 \dots \lambda_{N-1}]^T$ é um vetor coluna cujos elementos são os autovalores de \mathbf{R}_0 e $\mathbf{p}(n) = [p_0(n) p_1(n) \dots$

$p_{N-1}(n)]^T$ é um vetor coluna com os elementos da diagonal principal da matriz $\mathbf{P}(n) = \mathbf{Q}^T \mathbf{K}(n) \mathbf{Q}$. O modelo recursivo para o comportamento de $\mathbf{K}(n)$ é dado por [8, Eq. 12]

$$\mathbf{K}(n+1) = \mathbf{K}(n) - \frac{\beta}{\varepsilon + Nr_0} [\mathbf{K}(n)\mathbf{R}_0 + \mathbf{R}_0\mathbf{K}(n)] + \frac{\beta^2 \{E\{e^2(n)\}\mathbf{I} + 2\mathbf{R}_0\mathbf{K}(n)\}}{\varepsilon^2 + 2\varepsilon Nr_0 + N^2 r_0^2 + 2\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} r_{j-i}^2} \mathbf{R}_0, \quad (11)$$

em que $r_{j-i} = E\{x(n-i)x(n-j)\}$. A equação (11) pode ser simplificada para a forma de uma equação recursiva vetorial. Para tanto, inicialmente, pré e pós-multiplica-se (11) por \mathbf{Q}^T e \mathbf{Q} , respectivamente, levando a

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \frac{\beta}{\varepsilon + Nr_0} [\mathbf{P}(n)\mathbf{\Lambda} + \mathbf{\Lambda}\mathbf{P}(n)] + \frac{\beta^2 \{E\{e^2(n)\}\mathbf{I} + 2\mathbf{\Lambda}\mathbf{P}(n)\}}{\varepsilon^2 + 2\varepsilon Nr_0 + N^2 r_0^2 + 2\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} r_{j-i}^2} \mathbf{\Lambda} \quad (12)$$

Visto que apenas os elementos da diagonal principal de $\mathbf{P}(n)$ são de interesse, a partir de (12), pode-se construir a seguinte equação recursiva vetorial

$$\mathbf{p}(n+1) = [\mathbf{I} - a\mathbf{\Lambda} + 2b\mathbf{\Lambda}\mathbf{\Lambda}] \mathbf{p}(n) + b\boldsymbol{\lambda}^T \mathbf{p}(n) \boldsymbol{\lambda} + b_r \boldsymbol{\lambda} \quad (13)$$

em que $b = \beta^2 / (\varepsilon^2 + 2\varepsilon Nr_0 + N^2 r_0^2 + 2\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} r_{j-i}^2)$ e $a = 2\beta / (\varepsilon + Nr_0)$.

As equações (8), (10) e (13) podem ser implementadas computacionalmente da seguinte forma:

Preparação:

$$\begin{cases} x_0 = 1 - 2 \cdot \beta \cdot (\varepsilon + Nr_0)^{-1} \\ x_1 = \beta^2 \cdot \left(\varepsilon^2 + 2 \cdot \varepsilon \cdot N \cdot r_0 + N^2 \cdot r_0^2 + 2 \cdot \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} r_{j-k}^2 \right)^{-1} \\ \left\{ \begin{array}{l} \text{para } i = 0 : N-1 \\ b_i = x_1 \cdot \lambda_i, \quad a_i = x_0 + 2 \cdot b_i \cdot \lambda_i, \quad c_i = b_i \cdot r_z \\ v_i(0) = -\sum_{k=0}^{N-1} (w_k^0 \cdot q_{k,i}) \quad // \text{ inicializa coeficientes} \\ p_i(0) = v_i^2(0) \quad // \text{ inicializa mom. seg. ordem} \end{array} \right. \end{cases}$$

Laço Principal:

$$\begin{cases} \text{para } n = 0 : T-2 \\ r(n) = \sum_{i=0}^{N-1} (\lambda_i \cdot p_i(n)) \\ \left\{ \begin{array}{l} \text{para } i = 0 : N-1 \\ p_i(n+1) = a_i \cdot p_i(n) + b_i \cdot r(n) + c_i \\ v_i(n+1) = v_i \cdot t_i(n) \end{array} \right. \\ r(n) = \sum_{i=0}^{N-1} (\lambda_i \cdot p_i(n)) \end{cases}$$

Finalização:

$$\left\{ \begin{array}{l} \text{para } n = 0 : T - 1 \\ E(n) = r_z + r(n) \\ \left\{ \begin{array}{l} \text{para } i = 0 : N - 1 \\ v_i(n) = \sum_{j=0}^{N-1} (q_{i,j} \cdot v_j(n)) \end{array} \right. \end{array} \right.$$

em que T é o número de iterações, $q_{k,i}$ é o elemento da k -ésima linha e i -ésima coluna da matriz de autovetores \mathbf{Q} . $r(n)$ é o erro médio quadrático em excesso e $E(n)$ é o erro médio quadrático na iteração n . Supõe-se nessa implementação que a inicialização é feita na origem, ou seja, $\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$. Caso a informação de erro em excesso seja suficiente e o conhecimento do valor real dos coeficientes não seja necessário, importando apenas a sua evolução, a etapa de finalização pode ser desconsiderada.

Para o caso em que $\varepsilon = 0$, essa seqüência de operações modela de forma eficiente o comportamento do algoritmo NLMS.

B. Algoritmo LMF

No caso do algoritmo LMF, as equações de coeficientes, erro médio quadrático e momentos de segunda ordem estão interligadas da seguinte forma [9, Eqs. 13, 22, 23]:

$$E\{\mathbf{v}(n+1)\} = [\mathbf{I} - 3\mu E\{e^2(n)\} \mathbf{R}_0] E\{\mathbf{v}(n)\} \quad (14)$$

$$E\{e^2(n)\} = r_z + tr\{\mathbf{R}_0 \mathbf{K}(n)\} \quad (15)$$

$$\begin{aligned} \mathbf{K}(n+1) &= \mathbf{K}(n) - 3\mu E\{e^2(n)\} [\mathbf{K}(n) \mathbf{R}_0 + \mathbf{R}_0 \mathbf{K}(n)] \\ &+ 15\mu^2 E\{z^4(n)\} \{tr\{\mathbf{R}_0 \mathbf{K}(n)\} \mathbf{I} + 2\mathbf{R}_0 \mathbf{K}(n)\} \mathbf{R}_0 \\ &+ \mu^2 E\{z^6(n)\} \mathbf{R}_0 \end{aligned} \quad (16)$$

Substituindo-se as equações (5) e (6), pré-multiplicando-se a equação dos coeficientes por \mathbf{Q}^T e pré e pós multiplicando-se (16) por \mathbf{Q}^T e \mathbf{Q} , respectivamente, chega-se em

$$E\{\mathbf{v}(n+1)\} = [\mathbf{I} - 3\mu \boldsymbol{\lambda}^T \mathbf{p}(n) \boldsymbol{\Lambda} - 3\mu r_z \boldsymbol{\Lambda}] E\{\mathbf{v}(n)\} \quad (17)$$

$$E\{e^2(n)\} = r_z + \boldsymbol{\lambda}^T \mathbf{p}(n) \quad (18)$$

$$\mathbf{p}(n+1) = \mathbf{B} \cdot \mathbf{p}(n) + c \cdot \boldsymbol{\lambda}^T \mathbf{p}(n) \boldsymbol{\lambda} + d \cdot \boldsymbol{\lambda} \quad (19)$$

em que $\mathbf{B} = \mathbf{I} - 6\mu E\{e^2(n)\} \boldsymbol{\Lambda} + 30\mu^2 E\{z^4(n)\} \boldsymbol{\Lambda} \boldsymbol{\Lambda}$, $c = 15\mu^2 E\{z^4(n)\}$, $d = \mu E\{z^6(n)\}$, $E\{\mathbf{v}(n)\} = \mathbf{Q}^T \{\mathbf{v}(n)\}$ e $\mathbf{p}(n)$ é um vetor coluna com os elementos da diagonal principal da matriz $\mathbf{P}(n) = \mathbf{Q}^T \mathbf{K}(n) \mathbf{Q}$, definido da mesma forma que para o algoritmo ε -NLMS.

O algoritmo otimizado para a implementação do conjunto de equações (17)-(19), com inicialização na origem, é descrito pela seguinte seqüência de operações:

Preparação:

$$\left\{ \begin{array}{l} x_0 = -3 \cdot \mu, \quad x_1 = 2 \cdot x_0, \quad x_2 = \mu^2, \quad x_3 = 15 \cdot x_2 \cdot E\{z^4(n)\} \\ x_4 = 2 \cdot x_3, \quad x_5 = x_1 \cdot r_z, \quad x_6 = x_2 \cdot E\{z^6(n)\}, \quad x_7 = x_0 \cdot r_z \\ \left\{ \begin{array}{l} \text{para } i = 0 : N - 1 \\ a_i = x_1 \cdot \lambda_i, \quad b_i = 1 + \lambda_i \cdot (x_4 \cdot \lambda_i + x_5) \\ c_i = x_3 \cdot \lambda_i, \quad d_i = x_6 \cdot \lambda_i, \quad e_i = x_0 \cdot \lambda_i, \quad f_i = 1 + x_7 \cdot \lambda_i \\ v_i(0) = -\sum_{k=0}^{N-1} (w_k^0 \cdot q_{k,i}) \quad // \text{ inicializa coeficientes} \\ p_i(0) = v_i^2(n) \quad // \text{ inicializa mom. seg. ordem} \end{array} \right. \end{array} \right.$$

Lço Principal:

$$\left\{ \begin{array}{l} \text{para } n = 0 : T - 2 \\ r(n) = \sum_{i=0}^{N-1} (\lambda_i \cdot p_i(n)) \\ \left\{ \begin{array}{l} \text{para } i = 0 : N - 1 \\ p_i(n+1) = (a_i \cdot r(n) + b_i) \cdot p_i(n) + c_i \cdot r(n) + d_i \\ v_i(n+1) = (e_i \cdot r(n) + f_i) \cdot v_i(n) \end{array} \right. \\ r(n) = \sum_{i=0}^{N-1} (\lambda_i \cdot p_i(n)) \end{array} \right.$$

Finalização:

$$\left\{ \begin{array}{l} \text{para } n = 0 : T - 1 \\ E(n) = r_z + r(n) \\ \left\{ \begin{array}{l} \text{para } i = 0 : N - 1 \\ v_i(n) = \sum_{j=0}^{N-1} (q_{i,j} \cdot v_j(n)) \end{array} \right. \end{array} \right.$$

C. Algoritmo Projeções Afins

As equações que descrevem o comportamento do algoritmo de Projeções afins [10, Eqs. 11,19,20,23] são definidas como

$$\begin{aligned} \mathbf{K}(n+1) &= \mathbf{K}(n) - 2/(G-2) \mathbf{K}(n) \\ &+ \frac{[G \cdot tr\{\mathbf{K}(n)\} + P \cdot E\{\mathbf{v}^T(n)\} E\{\mathbf{v}(n)\}]}{N(G^2 + 2G)} \mathbf{I} \end{aligned} \quad (20)$$

$$+ \frac{Nr_z \left(1 + \mathbf{a}^T \mathbf{a} + r_u \cdot tr\left\{ E\left\{ \left[\mathbf{X}_p^T \mathbf{X}_p \right]^{-1} \right\} \right\} \right)}{G(G-2)(G-4)r_u} \mathbf{I}$$

$$E\{\mathbf{v}(n+1)\} = (G-3)/(G-2) \cdot E\{\mathbf{v}(n)\} \quad (21)$$

$$\begin{aligned} E\{e^2(n)\} &= \left(1 + \mathbf{a}^T \mathbf{a} + r_u \cdot tr\left\{ E\left\{ \left[\mathbf{X}_p^T \mathbf{X}_p \right]^{-1} \right\} \right\} \right) r_z \\ &+ tr\{\mathbf{K}(n)\} \cdot Gr_u / N \end{aligned} \quad (22)$$

em que $G = N - P$, r_u é a variância do sinal Gaussiano branco sujeito a um filtro autoregressivo de ordem P e coeficientes $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_p]$ que gera o sinal de excitação $x(n)$. As

equações (20) a (22) são apresentadas de forma ligeiramente diferente de [10], de forma a tornar evidente que $\mathbf{R}_{\phi\phi}$ é diagonal. Assim, $\mathbf{Q} = \mathbf{I}$ e a etapa de desacoplamento das equações (multiplicação por \mathbf{Q} e \mathbf{Q}^T) não é necessária. Portanto, procede-se diretamente à determinação da equação recursiva vetorial dos elementos da diagonal principal de $\mathbf{K}(n)$, resultando em:

$$\mathbf{k}(n+1) = (G-4)/(G-2) \cdot \mathbf{k}(n) + \frac{\left[G \cdot \sum_{i=0}^{N-1} \mathbf{k}(n) + PE \{ \mathbf{v}^T(n) \} E \{ \mathbf{v}(n) \} \right] \mathbf{i}}{N(G^2 + 2G)} + \frac{Nr_z \left(1 + \mathbf{a}^T \mathbf{a} + r_u \text{tr} \left\{ E \left\{ \left[\mathbf{X}_p^T \mathbf{X}_p \right]^{-1} \right\} \right\} \right) \mathbf{i}}{G(G-2)(G-4)r_u} \quad (23)$$

$$E \{ \mathbf{v}(n+1) \} = (G-3)/(G-2) \cdot E \{ \mathbf{v}(n) \} \quad (24)$$

$$E \{ e^2(n) \} = \frac{Gr_u}{N} \sum_{i=0}^{N-1} \mathbf{k}(n) + \left(1 + \mathbf{a}^T \mathbf{a} + r_u \text{tr} \left\{ E \left\{ \left[\mathbf{X}_p^T \mathbf{X}_p \right]^{-1} \right\} \right\} \right) r_z \quad (25)$$

em que $\mathbf{i} = [1 \ 1 \ \dots \ 1]^T$. A implementação otimizada do conjunto de equações (23)-(25) é dada por:

Preparação:

$$\left\{ \begin{array}{l} G = N - P, \ a = (G-4)/(G-2), \ b = [N \cdot (G+2)]^{-1} \\ c = P \cdot [N \cdot G \cdot (G+2)]^{-1}, \ h = G \cdot r_u / N \\ g = \left(1 + \mathbf{a}^T \mathbf{a} + r_u \cdot \text{tr} \left\{ E \left\{ \left[\mathbf{X}_p^T(n) \mathbf{X}_p(n) \right]^{-1} \right\} \right\} \right) \cdot r_z \\ e = g \cdot N [G \cdot (G-2) \cdot (G-4) \cdot r_u]^{-1}, \ f = (G-3)/(G-2) \\ \left\{ \begin{array}{l} \text{para } i = 0 : N-1 \\ \mathbf{v}_i(0) = -w_i^o \quad // \text{ inicializa coeficientes na origem} \\ \mathbf{k}_i(0) = \mathbf{v}_i^2(0) \quad // \text{ inicializa mom. seg. ordem} \end{array} \right. \end{array} \right.$$

Laço Principal:

$$\left\{ \begin{array}{l} \text{para } n = 0 : T-2 \\ r(n) = \sum_{i=0}^{N-1} k_i(n), \ d = \sum_{i=0}^{N-1} v_i^2(n) \\ l = b \cdot r(n) + c \cdot d + e \\ \left\{ \begin{array}{l} \text{para } i = 0 : N-1 \\ k_i(n+1) = a \cdot p_i(n) + l \\ v_i(n+1) = f \cdot v_i(n) \end{array} \right. \end{array} \right.$$

Finalização:

$$\left\{ \begin{array}{l} \text{para } n = 0 : T-1 \\ E(n) = g + h \cdot r(n) \end{array} \right.$$

As variáveis são descritas da mesma forma que para o caso do algoritmo ε -NLMS. Caso apenas o erro em excesso seja de interesse, a parte referente à finalização pode ser desconsiderada.

V. RESULTADOS

As tabelas I a III apresentam uma confrontação entre a complexidade computacional de implementações dos modelos dos algoritmos ε -NLMS, LMF e Projeções Afins para as versões originais e simplificadas. Como resultado pode-se observar a eficiência dos algoritmos apresentados neste trabalho.

TABELA I
COMPLEXIDADE PARA AVALIAÇÃO DO ERRO MÉDIO QUADRÁTICO EM EXCESSO POR ITERAÇÃO DO ALGORITMO ε -NLMS

	Multiplicações	Adições
versão original	$2N^3+2N^2+1$	$2N^3+2N^2+N-1$
nova versão	$3N$	$3N-1$

TABELA II
COMPLEXIDADE PARA AVALIAÇÃO DO ERRO MÉDIO QUADRÁTICO EM EXCESSO POR ITERAÇÃO DO ALGORITMO LMF

	Multiplicações	Adições
versão original	$2N^3+2N^2+2N+2$	$2N^3+2N^2+2N+1$
nova versão	$4N$	$4N-1$

TABELA III
COMPLEXIDADE PARA AVALIAÇÃO DO ERRO MÉDIO QUADRÁTICO EM EXCESSO POR ITERAÇÃO DO ALGORITMO DE PROJEÇÕES AFINS

	multiplicações	Adições
versão original	N^3+2N^2+N+2	N^3+2N^2+3N-1
nova versão	$3N+2$	$3N$

A partir das tabelas I a III é possível verificar que os algoritmos desenvolvidos neste trabalho permitem que o custo computacional para a avaliação do erro médio quadrático teórico em excesso seja da ordem do número de coeficientes do filtro adaptativo. Isso representa uma significativa diminuição do tempo de processamento visto que a implementação das equações nas formas originalmente propostas possui uma complexidade computacional da ordem do cubo do número de coeficientes.

Os algoritmos propostos e exemplos de aplicações podem ser encontrados em [14] na forma de programas para MatLab e Scilab.

VI. CONCLUSÕES

Este trabalho apresentou um conjunto de algoritmos de baixa complexidade computacional para a implementação de modelos teóricos para o comportamento dos algoritmos ε -NLMS, LMF e Projeções Afins. A metodologia utilizada baseia-se no desacoplamento do conjunto de equações que descrevem o comportamento do valor médio dos coeficientes e do erro médio quadrático, sendo descrições exatas dos modelos originalmente desenvolvidos.

Os resultados aqui apresentados somam-se ao trabalho desenvolvido em [8-10], no sentido de popularizar ferramentas de projeto e análise de filtros adaptativos. Os algoritmos descritos encontram-se disponíveis em [14] na forma de programas para MatLab e SciLab.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pela FAPERGS e pelo CNPq (300787/87-1).

REFERÊNCIAS

- [1] S. Haykin, *Adaptive Filter Theory*, 4 ed., Prentice-Hall, 2002
- [2] B. Widrow e S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.
- [3] D.G. Manolakis, V.K. Ingle e S.M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*, McGraw-Hill, 2000.
- [4] B. Widrow e M. Hoff Jr., "Adaptive Switching Circuits", *IRE WesCon*, pp. 96-104, 1960.
- [5] K. Mayyas, T. Abounasr, "Leaky LMS Algorithm: MSE Analysis for Gaussian Data", *IEEE Transactions on Signal Processing*, v. 45, no. 4, pp. 927-934, 1997.
- [6] J.C.M. Bermudez e M.H. Costa, "Optimum Leakage Factor for the MOV-LMS Algorithm in Nonlinear Modeling and Control Systems", *International Conference on Acoustics, Speech and Signal Processing*, Orlando, v. 2, pp. 1393-1396, 2002.
- [7] M.H. Costa e J.C.M. Bermudez "Análise do Algoritmo LMS Sujeito a uma Não-Linearidade do Tipo Zona-Morta no Caminho Secundário", *XX Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, pp. 705-710, 2003.
- [8] J.C.M. Bermudez e M.H. Costa, "A Statistical Analysis of the ε -NLMS and NLMS Algorithms for Correlated Gaussian Signals", *XX Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, pp. 582-586, 2002.
- [9] P.I. Hubscher e J.C.M. Bermudez, "An Improved Statistical Analysis of the Least Mean Fourth (LMF) Adaptive Algorithm", *IEEE Transactions on Signal Processing*, vol. 51(3), pp. 664-671, 2003.
- [10] S.J.M. Almeida, J.C.M. Bermudez, N.J. Bershad e M.H. Costa, "A Stochastic Affine Projection Algorithm for Gaussian Inputs", *IEEE Conference on Acoustics, Speech and Signal Processing*, Hong-Kong, v. 6, pp. 313-316, 2003.
- [11] M.H. Costa e J.C.M. Bermudez "A Fully Analytical Recursive Stochastic Model to the Normalized Signed Regressor LMS Algorithm", *Seventh International Symposium on Signal Processing and its Applications*, Paris, v. 2, pp. 587-590, 2003.
- [12] V.H. Nascimento e A.H. Sayed, "Are Ensemble-Average Learning Curves Reliable in Evaluating the Performance of Adaptive Filters?", *Thirty-Second Asilomar Conference on Signals, Systems & Computers*, v. 2, pp.1171-1175, 1998.
- [13] C. Breining, P. Dreiscitel, E. Hansler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt e J. Tilp, "Acoustic Echo Control. An Application of Very-High-Order Adaptive Filters", *IEEE Signal Processing Magazine*, v. 16(4), pp. 42-69, 1999.
- [14] <http://atlas.ucpel.tche.br/~fpires>