

Análise Estatística do Algoritmo MDLMS

Javier E. Kolodziej, Orlando J. Tobias e Rui Seara

Resumo—Este trabalho apresenta uma análise estatística do algoritmo adaptativo LMS com atraso modificado (MDLMS) para a condição de adaptação lenta. Esse algoritmo é uma alternativa ao algoritmo LMS com atraso convencional, objetivando melhorar a velocidade de convergência e eliminar a influência do atraso no desempenho do algoritmo. Modelos analíticos para os momentos de primeira e segunda ordens do vetor de coeficientes do filtro adaptativo são obtidos. Na análise proposta, são considerados a natureza variante no tempo do passo de adaptação e um parâmetro de regularização, adicionado à estimativa da potência, incluído para evitar a divisão por zero na operação de normalização. A partir do momento de segunda ordem, é determinado o valor máximo do passo de adaptação que garante a convergência do algoritmo. Finalmente, uma expressão para o desajuste também é derivada. Resultados de simulação numérica permitem avaliar a precisão dos modelos aqui propostos.

Palavras-Chave—Algoritmo LMS com atraso, análise estatística, desajuste, estabilidade.

Abstract—This paper proposes a stochastic analysis for the modified delayed least-mean-square (MDLMS) algorithm under slow adaptation condition. This algorithm is used as an alternative to the standard delayed LMS (DLMS) one for both improving the algorithm convergence speed and making its behavior independent on the delay value. Analytical models for the first and second moments of the adaptive filter weight vector are obtained. Furthermore, the time-varying nature of normalized step size and a regularization parameter (which prevents division by zero during the normalization operation) are included in the models. The adaptive filter stability is analyzed and an upper bound for the step-size parameter and expressions for the algorithm misadjustment are also derived. Numerical simulation results permit to assess the accuracy of the proposed models.

Keywords—Delayed LMS algorithm, stochastic analysis, misadjustment, stability.

I. INTRODUÇÃO

O algoritmo LMS [1] tem sido muito utilizado para implementação de sistemas adaptativos considerando filtros com resposta ao impulso finita (FIR). Isso se deve, principalmente, à sua simplicidade e robustez. Entretanto, em algumas aplicações, a adaptação dos coeficientes do filtro não pode mais ser efetuada usando o algoritmo LMS convencional. Um desses casos é quando existe algum atraso

(inerente ao sistema) no caminho do sinal de erro. Por exemplo, em algumas aplicações de cancelamento de eco [2], a resposta desejada do filtro adaptativo estará disponível para o algoritmo com um certo retardo em função do atraso introduzido pelo canal de comunicação. O mesmo problema acontece em equalização adaptativa de canais [3], onde se tem um atraso inerente introduzido pelo decodificador. O efeito final desses atrasos é a perda de alinhamento temporal entre as entradas do filtro e o sinal de erro utilizado para a atualização dos coeficientes. Para resolver esse problema, deve-se também usar o mesmo valor de atraso tanto no sinal de entrada quanto no erro residual. Tais sinais são utilizados pelo algoritmo de adaptação dos coeficientes. Esse procedimento resulta no algoritmo adaptativo denominado LMS com atraso (DLMS) [4], [5]. A maior desvantagem desse algoritmo é a baixa velocidade de convergência e a deterioração de desempenho, na medida em que aumenta o valor do atraso. Isso é decorrente do reduzido passo de adaptação, requerido para a manutenção da estabilidade do algoritmo, em função do atraso do sistema. Para contornar tal problema, tem sido proposto, em [6], uma modificação no algoritmo DLMS. Tal algoritmo é denominado LMS com atraso modificado (MDLMS). As vantagens desse algoritmo são:

- i) seu comportamento não é mais influenciado pelo atraso no sinal de erro introduzido pelo sistema;
- ii) sua melhor velocidade de convergência, uma vez que a modificação resulta em um algoritmo normalizado.

Uma análise detalhada desse algoritmo ainda não se encontra disponível na literatura técnica da área. Portanto, o objetivo deste trabalho é obter um modelo estatístico apropriado que descreva seu comportamento. Em particular, são apresentadas expressões analíticas para o momento de primeira e segunda ordens do vetor de coeficientes do filtro adaptativo, bem como para a curva de aprendizagem. Tal análise é realizada assumindo-se sinais de entrada Gaussianos. Também, é determinada a faixa de valores do passo de adaptação que garante a estabilidade do algoritmo adaptativo. Uma expressão para o desajuste do algoritmo em questão é também apresentada. Resultados de simulação numérica permitem verificar a qualidade do modelo proposto quando comparado com aqueles obtidos pelo método Monte Carlo (MC).

II. ALGORITMO LMS COM ATRASO

A. Algoritmo DLMS Convencional e MDLMS

A equação de atualização dos coeficientes do algoritmo DLMS convencional [4], [5], para um filtro FIR com N coeficientes, é dada por

Javier E. Kolodziej, 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: {javier, orlando, seara}@linse.ufsc.br.

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n-D)e(n-D), \quad (1)$$

com o sinal de erro expresso como

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n). \quad (2)$$

Em (2), o sobrescrito T denota a operação de transposição, D é o atraso no caminho do erro e $\mathbf{w}(n) = [w_0(n) \ w_1(n) \ \dots \ w_{N-1}(n)]^T$ representa o vetor de coeficientes do filtro adaptativo. O vetor $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$ é o vetor de entrada, $d(n)$ caracteriza o sinal desejado, $e(n)$ representa o erro de estimação e μ é o passo de adaptação.

B. Algoritmo MDLMS

Para melhorar a velocidade de convergência do algoritmo DLMS convencional, como apresentado em [6], o algoritmo MDLMS emprega um coeficiente de adaptação variável no tempo, denotado por $\mu(n)$.

Para se determinar o melhor valor de $\mu(n)$, são considerados o erro de estimação *a posteriori* $\xi(n)$ e a função custo $J(n)$, definidos, respectivamente, como segue:

$$\xi(n) = d(n-D) - \mathbf{x}^T(n-D)\mathbf{w}(n+1) \quad (3)$$

e

$$J(n) = \xi^2(n). \quad (4)$$

Assim, o valor de $\mu(n)$ procurado é aquele que minimiza a função custo $J(n)$, sendo expresso por [6]

$$\mu(n) = \frac{d(n-D) - \mathbf{x}^T(n-D)\mathbf{w}(n)}{\mathbf{x}^T(n-D)\mathbf{x}(n-D)e(n-D)}. \quad (5)$$

Finalmente, substituindo-se (5) em (1), obtém-se a equação de atualização dos coeficientes do algoritmo MDLMS, dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{\mathbf{x}^T(n-D)\mathbf{x}(n-D)} \mathbf{x}(n-D)\alpha(n-D). \quad (6)$$

Note-se que em (6) é utilizado um pseudo-sinal de erro, dado por

$$\alpha(n-D) = d(n-D) - \mathbf{x}^T(n-D)\mathbf{w}(n). \quad (7)$$

C. Considerações Práticas

Na implementação do algoritmo MDLMS, o termo $\mathbf{x}^T(n-D)\mathbf{x}(n-D)$, que fornece a potência do sinal de entrada, pode ser determinado de forma recursiva ao invés do cálculo direto. Para tal, consideramos a seguinte aproximação da potência normalizada. Assim,

$$\mathbf{x}^T(n-D)\mathbf{x}(n-D) \cong N\hat{\sigma}_x^2(n-D), \quad (8)$$

onde

$$\hat{\sigma}_x^2(n) = \frac{1}{M} \sum_{l=0}^{M-1} x^2(n-l), \quad (9)$$

com M sendo o comprimento da janela de observação. Também, para evitar a divisão por zero, é adicionada uma constante positiva ε (parâmetro de regularização) à estimação da potência, resultando na seguinte expressão:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\tau}{N[\hat{\sigma}_x^2(n-D) + \varepsilon]} \mathbf{x}(n-D)\alpha(n-D). \quad (10)$$

Em (6), assim como para qualquer tipo de implementação do algoritmo LMS normalizado (NLMS), é incluída uma constante positiva fixa τ , necessária para fornecer um controle sobre a convergência do algoritmo.

Assim, a partir de (10), determinaremos os momentos de primeira e segunda ordens do vetor de coeficientes do filtro adaptativo. Note que, fazendo $D=0$ e $\tau = N\mu$, o algoritmo MDLMS se transforma no algoritmo NLMS [7], podendo-se estender os resultados aqui obtidos também a este último. Contudo, o foco de nosso trabalho é o algoritmo MDLMS.

III. ANÁLISE

A. Momento de Primeira Ordem do Vetor de Coeficientes

Nesta seção, determinamos o momento de primeira ordem do vetor de coeficientes do algoritmo MDLMS. Para tal, substituímos (7) em (10), resultando em

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \tau \delta^{-1}(n) \mathbf{x}(n-D)d(n-D) - \tau \delta^{-1}(n) \mathbf{x}(n-D)\mathbf{x}^T(n-D)\mathbf{w}(n), \quad (11)$$

onde

$$\delta^{-1}(n) = \frac{1}{N[\hat{\sigma}_x^2(n-D) + \varepsilon]}. \quad (12)$$

Calculando-se o valor esperado em ambos os lados de (11), tem-se

$$E[\mathbf{w}(n+1)] = E[\mathbf{w}(n)] + \tau E[\delta^{-1}(n)\mathbf{x}(n-D)d(n-D)] - \tau E[\delta^{-1}(n)\mathbf{x}(n-D)\mathbf{x}^T(n-D)\mathbf{w}(n)]. \quad (13)$$

Para determinar os valores esperados de (13), utilizaremos as seguintes considerações:

(i) $\mathbf{w}(n)$ é estatisticamente independente de $\mathbf{x}(n)$ [8].

(ii) $\delta^{-1}(n)$ e $\mathbf{x}(n-D)\mathbf{x}^T(n-D)$ são processos conjuntamente estacionários, onde $\delta^{-1}(n)$ tem uma variação lenta quando comparado com $\mathbf{x}(n-D)\mathbf{x}^T(n-D)$. Assim, em tal situação, o Princípio da Média [9] pode ser invocado.

Deve ser ressaltado que tais considerações simplificativas limitam a abrangência do modelo para a condição de adaptação lenta. Dessa forma, a hipótese (i) é válida apenas para valores de τ pequeno [10] e (ii), se o comprimento da janela de observação M é suficientemente grande [9]. Assim, a partir dessas considerações, obtém-se as seguintes expressões:

$$E[\delta^{-1}(n)\mathbf{x}(n-D)d(n-D)] \approx E[\delta^{-1}(n)]E[\mathbf{x}(n-D)d(n-D)] \quad (14)$$

e

$$E[\delta^{-1}(n)\mathbf{x}(n-D)\mathbf{x}^T(n-D)\mathbf{w}(n)] \approx E[\delta^{-1}(n)]E[\mathbf{x}(n-D)\mathbf{x}^T(n-D)]E[\mathbf{w}(n)]. \quad (15)$$

Considerando-se tanto o sinal de entrada quanto a planta estacionários, tem-se

$$E[\mathbf{x}(n-D)d(n-D)] = \mathbf{p}_{xd} . \quad (16)$$

A expressão (16) representa o vetor de correlação cruzada entre os sinais de entrada e desejado; e

$$E[\mathbf{x}(n-D)\mathbf{x}^T(n-D)] = \mathbf{R}_{xx} , \quad (17)$$

denota a matriz de autocorrelação do sinal de entrada.

Substituindo-se (16) e (17) em (13), obtém-se a expressão do momento de primeira ordem do vetor de coeficientes. Assim,

$$E[\mathbf{w}(n+1)] = \{\mathbf{I} - \tau E[\delta^{-1}(n)]\mathbf{R}_{xx}\}E[\mathbf{w}(n)] + \tau E[\delta^{-1}(n)]\mathbf{p}_{xd} . \quad (18)$$

Para concluir, devemos determinar o valor esperado $E[\delta^{-1}(n)]$. Assumindo-se que o processo $\{x^2(n)\}$ tenha uma distribuição *qui*-quadrado com M graus de liberdade [11], o valor esperado procurado é, então, dado por [12] (ver Apêndice):

$$E[\delta^{-1}(n)] = \frac{M}{N(M-2)\sigma_x^2} - \varepsilon \frac{M^2}{N(M-2)(M-4)\sigma_x^4} . \quad (19)$$

B. Determinação do Vetor de Coeficientes Ótimo e do Valor de Regime Permanente

O vetor de coeficientes ótimo \mathbf{w}_o , que minimiza a função custo $J(n)$, é obtido substituindo-se (3) em (4) e fazendo-se $\mathbf{w} = \mathbf{w}_o$. Dessa forma,

$$J(n) = d^2(n-D) - d(n-D)\mathbf{x}^T(n-D)\mathbf{w}_o + \mathbf{x}^T(n-D)\mathbf{w}_o\mathbf{x}^T(n-D)\mathbf{w}_o . \quad (20)$$

Agora, determinando-se o valor esperado em ambos os lados de (20) e utilizando-se as considerações (i) e (ii), obtém-se

$$E[J(n)] = E[d^2(n-D)] - \mathbf{p}_{xd}\mathbf{w}_o + \mathbf{w}_o^T\mathbf{R}_{xx}\mathbf{w}_o . \quad (21)$$

Fazendo-se $\nabla E[J(n)] = 0$ [10], a expressão do vetor de coeficientes ótimo é

$$\mathbf{w}_o = \mathbf{R}_{xx}^{-1}\mathbf{p}_{xd} . \quad (22)$$

Note que a equação (22) coincide com a expressão que determina os coeficientes ótimos para o algoritmo LMS convencional [10].

Assumindo-se que o algoritmo converge, o valor de regime permanente do vetor de coeficientes do filtro é obtido fazendo-se $\lim_{n \rightarrow \infty} E[\mathbf{w}(n+1)] = \lim_{n \rightarrow \infty} E[\mathbf{w}(n)] = \mathbf{w}_\infty$. Assim, de (18), tem-se

$$\mathbf{w}_\infty = \{\mathbf{I} - \tau E[\delta^{-1}(n)]\mathbf{R}_{xx}\}\mathbf{w}_\infty + \tau E[\delta^{-1}(n)]\mathbf{p}_{xd} . \quad (23)$$

Então, após algumas manipulações algébricas, obtém-se

$$\mathbf{w}_\infty = \mathbf{R}_{xx}^{-1}\mathbf{p}_{xd} . \quad (24)$$

Novamente, a equação (24) coincide com o valor de regime permanente do algoritmo LMS convencional como também com a expressão (22).

C. Curva de Aprendizagem e Momento de Segunda Ordem do Vetor de Coeficientes

As expressões matemáticas que representam o comportamento do filtro podem ser expressas de forma mais compacta utilizando-se o vetor de erro nos coeficientes, definido como

$$\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_o . \quad (25)$$

Agora, pode-se expressar o sinal de erro (2) da seguinte forma:

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{v}(n) - \mathbf{x}^T(n)\mathbf{w}_o + \eta(n) , \quad (26)$$

onde $\eta(n)$ representa o erro de medição, sendo uma variável i.i.d, com média zero e variância σ_η^2 , não correlacionado com qualquer outro sinal do sistema.

Definindo-se o erro de estimação como

$$e_o(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}_o + \eta(n) , \quad (27)$$

substituindo-se (27) em (26), elevando-se ao quadrado ambos os lados da expressão resultante e determinando-se o valor esperado, obtém-se

$$E[e^2(n)] = E[e_o^2(n)] - 2E[e_o(n)\mathbf{x}^T(n)\mathbf{v}(n)] + E[\mathbf{x}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}(n)] . \quad (28)$$

Então, a partir de (i) e pelo Princípio da Ortogonalidade, temos que $E[e_o(n)\mathbf{x}(n)] = 0$ [10]. Considerando-se também que

$$E[\mathbf{x}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{x}(n)] = \text{tr}\{\mathbf{R}_{xx}E[\mathbf{v}(n)\mathbf{v}^T(n)]\} , \quad (29)$$

a curva de aprendizagem pode agora ser escrita como

$$E[e^2(n)] = e_{\min} + \text{tr}\{\mathbf{R}_{xx}E[\mathbf{v}(n)\mathbf{v}^T(n)]\} , \quad (30)$$

onde $e_{\min} = E[e_o^2(n)]$ é o mínimo erro atingível em regime permanente. Note que (30) é completamente definida a partir do conhecimento da matriz de covariância do vetor de erro nos coeficientes, denotado por $\mathbf{K}(n) = E[\mathbf{v}(n)\mathbf{v}^T(n)]$. Para obter $\mathbf{K}(n)$ subtrai-se \mathbf{w}_o em ambos os lados de (18) e determina-se o produto externo $\mathbf{v}(n)\mathbf{v}^T(n)$; logo, tomando-se o valor esperado em ambos os lados da expressão resultante, obtém-se

$$\begin{aligned} \mathbf{K}(n+1) &= \{\mathbf{I} - \tau E[\delta^{-1}(n)]\mathbf{R}_{xx}\}\mathbf{K}(n) - \tau E[\delta^{-1}(n)]\mathbf{K}(n)\mathbf{R}_{xx} \\ &\quad + \tau^2 E[\delta^{-2}(n)]\{2\mathbf{R}_{xx}\mathbf{K}(n)\mathbf{R}_{xx} + \mathbf{R}_{xx}\text{tr}\{\mathbf{R}_{xx}\mathbf{K}(n)\}\} \\ &\quad + \tau^2 E[\delta^{-2}(n)]e_{\min}\mathbf{R}_{xx} , \end{aligned} \quad (31)$$

onde (ver Apêndice)

$$E[\delta^{-2}(n)] = \frac{1}{N^2} \left[\frac{M^2}{(M-2)(M-4)\sigma_x^4} - \varepsilon \frac{M^3}{(M-2)(M-4)(M-6)\sigma_x^6} \right] . \quad (32)$$

D. Estabilidade do Algoritmo

Determinaremos agora a faixa de valores da constante τ , que garante a estabilidade do algoritmo. Para tal, define-se as matrizes $\mathbf{C}(n) = \mathbf{Q}^T \mathbf{K}(n) \mathbf{Q}$ e $\mathbf{\Lambda} = \mathbf{Q}^T \mathbf{R}_{\mathbf{xx}} \mathbf{Q}$, onde $\mathbf{\Lambda}$ é uma matriz diagonal contendo os autovalores λ_i de $\mathbf{R}_{\mathbf{xx}}$; e \mathbf{Q} é uma matriz ortogonal, tendo como colunas os autovetores de $\mathbf{R}_{\mathbf{xx}}$. Então, pré-multiplicando e pós-multiplicando ambos os lados de (31) por \mathbf{Q}^T e \mathbf{Q} , respectivamente, obtém-se

$$\begin{aligned} \mathbf{C}(n+1) &= \mathbf{C}(n) - \tau E[\delta^{-1}(n)] \mathbf{C}(n) \mathbf{\Lambda} - \tau E[\delta^{-1}(n)] \mathbf{\Lambda} \mathbf{C}(n) \\ &+ \tau^2 E[\delta^{-2}(n)] \{2 \mathbf{\Lambda} \mathbf{C}(n) \mathbf{\Lambda} + \mathbf{\Lambda} \text{tr}[\mathbf{\Lambda} \mathbf{C}(n)]\} + \tau^2 E[\delta^{-2}(n)] e_{\min} \mathbf{\Lambda}. \end{aligned} \quad (33)$$

A partir de [13], a relação $c_{ij}^2(n) \leq c_{ii}(n) c_{jj}(n)$ é válida, sendo c_{ij} um elemento da matriz $\mathbf{C}(n)$. Assim, a convergência de (33) é determinada analisando-se apenas os elementos da diagonal de $\mathbf{C}(n)$. Então, definindo-se o vetor contendo os elementos da diagonal de $\mathbf{C}(n)$ por $\mathbf{c}(n) = [c_{11}(n) \ c_{22}(n) \ \dots \ c_{NN}(n)]^T$, e o vetor de autovalores da matriz $\mathbf{R}_{\mathbf{xx}}$ por $\boldsymbol{\lambda} = [\lambda_0 \ \lambda_1 \ \dots \ \lambda_{N-1}]^T$, pode-se escrever

$$\mathbf{\Lambda} \text{tr}[\mathbf{\Lambda} \mathbf{C}(n)] = \boldsymbol{\lambda} \boldsymbol{\lambda}^T \mathbf{c}(n). \quad (34)$$

Rescrevendo-se (33) em termos de $\mathbf{c}(n)$, obtém-se

$$\mathbf{c}(n+1) = \mathbf{F} \mathbf{c}(n) + E[\delta^{-2}(n)] e_{\min} \boldsymbol{\lambda}, \quad (35)$$

onde

$$\mathbf{F} = \mathbf{I} - 2\tau E[\delta^{-1}(n)] \mathbf{\Lambda} + 2\tau^2 E[\delta^{-2}(n)] \mathbf{\Lambda}^2 + \tau^2 E[\delta^{-2}(n)] \boldsymbol{\lambda} \boldsymbol{\lambda}^T. \quad (36)$$

Uma condição necessária e suficiente para se obter a convergência do algoritmo é que o módulo dos autovalores de \mathbf{F} sejam menores do que 1. Então, utilizando-se $\lambda' = \lambda E[\delta^{-1}(n)]$ e a aproximação $E[\delta^{-2}(n)] \approx E[\delta^{-1}(n)] \times E[\delta^{-1}(n)]$, pode-se seguir o mesmo procedimento usado em [13]. Assim, a expressão para o limite superior de τ é

$$0 < \tau \leq \frac{1}{3 \sum_{i=0}^{N-1} \lambda'_i} = \frac{1}{3 \text{tr}\{E[\delta^{-1}(n)] \mathbf{R}_{\mathbf{xx}}\}}. \quad (37)$$

Observa-se que (37) fornece uma estimativa para o valor máximo de τ , valor esse superior ao correspondente obtido através de simulações numéricas. Tal fato advém das hipóteses simplificativas usadas na derivação da expressão $\mathbf{K}(n)$.

E. Desajuste do Algoritmo

A curva de aprendizagem (30) depende apenas dos elementos da diagonal da matriz $\mathbf{R}_{\mathbf{xx}} \mathbf{K}(n)$, dados por $\text{tr}[\mathbf{R}_{\mathbf{xx}} \mathbf{K}(n)]$. Dessa forma, rescrevendo-se (30), tem-se

$$E[e^2(n)] = e_{\min} + \boldsymbol{\lambda}^T \mathbf{c}(n). \quad (38)$$

Assumindo-se que o algoritmo convergiu, o valor do erro quadrático médio (EQM), em regime permanente, é dado por

$$E[e^2(\infty)] = e_{\min} + \boldsymbol{\lambda}^T \mathbf{c}_{\infty}. \quad (39)$$

O vetor \mathbf{c}_{∞} é obtido pelo $\lim_{n \rightarrow \infty} E[\mathbf{c}(n+1)] = \lim_{n \rightarrow \infty} E[\mathbf{c}(n)] = \mathbf{c}_{\infty}$ em (35). Assim, tem-se

$$\mathbf{c}_{\infty} = E[\delta^{-2}(\infty)] e_{\min} (\mathbf{I} - \mathbf{F}) \boldsymbol{\lambda}. \quad (40)$$

O termo $\boldsymbol{\lambda}^T \mathbf{c}_{\infty}$ é denominado erro em excesso [10]. Uma figura de mérito usualmente considerada em filtragem adaptativa é o desajuste do algoritmo, definido como

$$\psi = \frac{E[e^2(\infty)] - e_{\min}}{e_{\min}} = E[\delta^{-2}(n)] \boldsymbol{\lambda}^T (\mathbf{I} - \mathbf{F})^{-1} \boldsymbol{\lambda}. \quad (41)$$

Seguindo-se novamente o procedimento usado em [13] (tomando-se $\lambda' = \lambda E[\delta^{-1}(n)]$ e a aproximação $E[\delta^{-2}(n)] \approx E[\delta^{-1}(n)] E[\delta^{-1}(n)]$), obtém-se a expressão do desajuste do algoritmo. Assim,

$$\psi = \frac{\sum_{i=0}^{N-1} \frac{\tau E[\delta^{-1}(n)] \lambda_i}{1 - \tau E[\delta^{-1}(n)] \lambda_i}}{1 - \sum_{i=0}^{N-1} \frac{\tau E[\delta^{-1}(n)] \lambda_i}{1 - \tau E[\delta^{-1}(n)] \lambda_i}}. \quad (42)$$

IV. RESULTADOS DE SIMULAÇÃO

As simulações numéricas descritas nesta seção são realizadas tomando-se um problema de identificação de sistema. Nesse contexto, são apresentados três exemplos.

Exemplo 1

Este exemplo mostra uma comparação entre o desempenho dos algoritmos DLMS convencional e MDLMS. O passo de adaptação do primeiro foi determinado segundo o critério da mais rápida velocidade de convergência, apresentado em [4], [5]. Para se fazer uma comparação justa, a constante τ do algoritmo MDLMS foi selecionada de forma que o valor do EQM em regime permanente seja idêntico ao do DLMS convencional. O sinal de entrada usado é um sinal correlacionado, obtido através de um sistema AR(2), dado por $x(n) = a_1 x(n-1) + a_2 x(n-2) + v(n)$, onde $v(n)$ é um ruído branco com variância σ_v^2 de tal forma que $\sigma_x^2 = 1$. O valor dos coeficientes do sistema auto-regressivo são: $a_1 = 0,08$ e $a_2 = -0,85$. A dispersão dos autovalores da matriz de autocorrelação do sinal de entrada é igual a $\lambda_{\max} / \lambda_{\min} = 56$. O vetor que representa a planta é dado por $[-1,00; 0,80; 0,70; 0,80; -0,15; 0,20; -0,50; 0,90; 0,40; 1,00; 0,50]^T$. Os outros parâmetros utilizados são: $\tau = 0,3$; $\varepsilon = 0,001$; $\sigma_x^2 = 1$; $\sigma_{\eta}^2 = 10^{-4}$ e $D = 20$. Para as simulações Monte Carlo, têm sido consideradas 300 realizações independentes. A Fig. 1(a) ilustra o comportamento dos coeficientes do filtro obtido pelo método MC e através do modelo proposto. Nessa figura, verifica-se uma boa concordância entre os resultados obtidos pelo modelo e por simulação MC. Na Fig. 1(b), são mostradas as curvas de

aprendizagem tanto para o algoritmo MDLMS quanto para o algoritmo DLMS convencional. Além da concordância entre o modelo e a simulação, verifica-se também um melhor desempenho do algoritmo MDLMS, quando comparado com o DLMS convencional. Nessa figura, nota-se também que o modelo apresenta um certo descasamento com a simulação na fase de transiente. Isso é decorrente das hipóteses simplificativas consideradas. Em particular, o modelo apresentado é mais apropriado para uma condição de adaptação lenta. Assim, quando não satisfeita tal condição, algumas suposições de análise são menos justificáveis. Por exemplo, a hipótese de independência entre $\mathbf{w}(n)$ e $\mathbf{x}(n-D)\mathbf{x}^T(n-D)$. É importante notar que, para os atrasos utilizados, a curva de aprendizagem do algoritmo MDLMS é sempre a mesma. Isso representa uma vantagem muito importante em relação ao algoritmo DLMS convencional, o qual piora o seu comportamento à medida que se aumenta o valor do atraso.

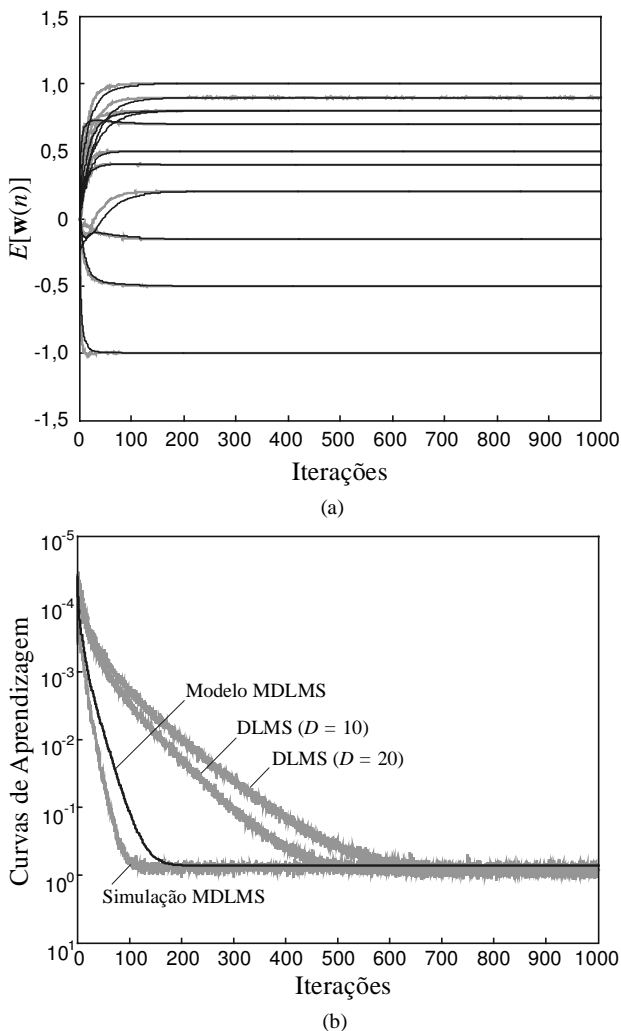


Fig. 1. Exemplo 1. Algoritmo MDLMS. (a) Comportamento médio dos coeficientes. Simulação MC (linha cinza). Modelo proposto (linha preta). (b) Curvas de aprendizagem. Simulação MC (linha irregular cinza). Modelo proposto (linha preta). Curvas de aprendizagem para o algoritmo DLMS convencional com atrasos de $D=10$ e $D=20$ (linha irregular cinza) (apenas para efeito de comparação)

Exemplo 2

Neste exemplo, a planta utilizada é descrita pelos coeficientes de uma janela de Hanning com 32 coeficientes de norma unitária. Os demais parâmetros considerados são: $\tau=0,01$; $\epsilon=0,001$; $\lambda_{\max}/\lambda_{\min}=110$; $D=10$; $\sigma_x^2=1$; e $\sigma_{\eta}^2=10^{-4}$. As simulações MC são obtidas através de 50 realizações independentes. As curvas de aprendizagem, obtidas por simulação e pelo modelo proposto, são mostradas na Fig. 2. Nesse exemplo, observa-se um melhor casamento entre o modelo com a simulação MC. Isso se deve à condição de adaptação lenta considerada nesse exemplo, a qual faz com que as hipóteses (i) e (ii) sejam melhor justificadas.

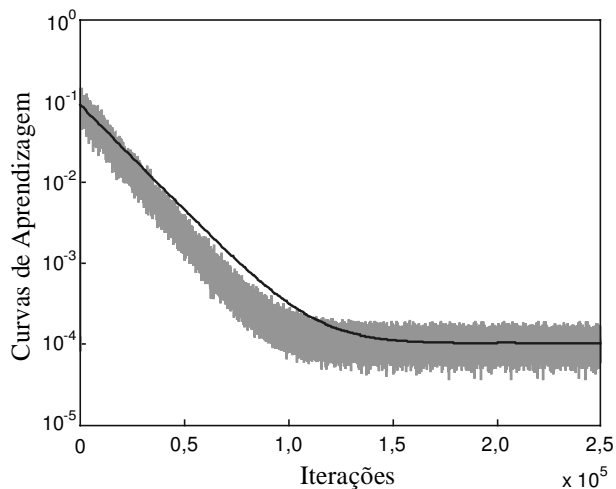


Fig. 2. Exemplo 2. Curvas de aprendizagem para uma planta com 32 coeficientes. Simulação MC (linha cinza). Modelo proposto (linha preta).

Exemplo 3

Na Fig. 3, são apresentados os resultados obtidos para uma planta também modelada por uma janela de Hanning de norma unitária, agora, com 50 coeficientes. O sinal de entrada é um ruído branco. Novamente, é considerada uma condição de adaptação lenta e, como esperado, o modelo proposto apresenta um bom casamento com os resultados de simulação numérica.

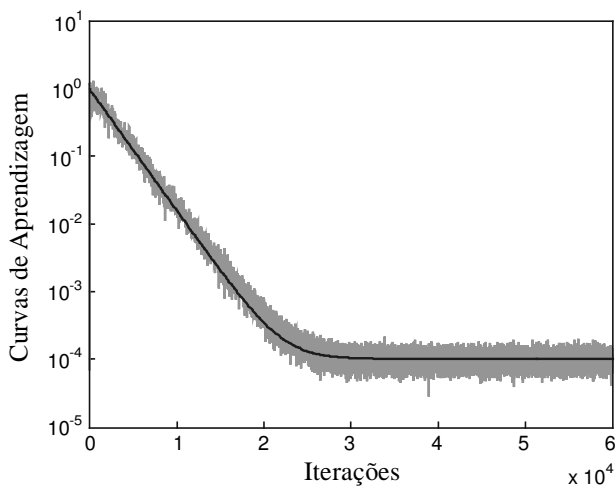


Fig. 3. Exemplo 3. Curvas de aprendizagem para uma planta com 50 coeficientes. Simulação MC (linha cinza). Modelo proposto (linha preta).

V. CONCLUSÕES

Neste trabalho, foram apresentadas as expressões analíticas que descrevem os momentos de primeira e segunda ordens do vetor de coeficientes, como também a curva de aprendizagem do algoritmo MDLMS. Aproximações simplificativas para o desenvolvimento dos modelos são consideradas, baseadas em algumas hipóteses preestabelecidas, o que limita o modelo à condição de adaptação lenta. Para tais condições, o comportamento dos modelos traduzem satisfatoriamente os resultados esperados.

APÊNDICE

Determinação de $E[\delta^{-1}(n)]$ e $E[\delta^{-2}(n)]$

O valor esperado $E[\delta^{-1}(n)]$ é determinado a partir da seguinte expressão:

$$E\{[\hat{\sigma}_x^2(n) + \varepsilon]^{-1}\} \approx E\left\{\left[\frac{1}{M} \sum_{l=0}^{M-1} x^2(n-l)\right]^{-1}\right\} - \varepsilon E\left\{\left[\frac{1}{M} \sum_{l=0}^{M-1} x^2(n-l)\right]^{-2}\right\}. \quad (43)$$

O que, segundo o desenvolvimento apresentado em [12], resulta em

$$E[\delta^{-1}(n)] = \frac{1}{N} \left[\frac{M}{(M-2)\sigma_x^2} - \varepsilon \frac{M^2}{(M-2)(M-4)\sigma_x^4} \right]. \quad (44)$$

Para o caso do valor esperado $E[\delta^{-2}(n)]$, considera-se a aproximação abaixo:

$$E\{[\hat{\sigma}_x^2(n) + \varepsilon]^{-2}\} \approx E\left\{\left[\frac{1}{M} \sum_{l=0}^{M-1} x^2(n-l)\right]^{-2}\right\} - \varepsilon E\left\{\left[\frac{1}{M} \sum_{l=0}^{M-1} x^2(n-l)\right]^{-3}\right\}. \quad (45)$$

Note-se que o valor esperado do primeiro termo do lado direito de (45) é dado em (44). Entretanto, o valor esperado do segundo termo é determinado considerando-se

$$E[1/z^3] = \frac{M^3}{2^{M/2} \sigma_x^M \Gamma(M/2)} \int_0^\infty z^{(M/2-4)} e^{-z/2\sigma_x^2} dz. \quad (46)$$

Então, utilizando-se o mesmo procedimento que em [12], tem-se

$$E[1/z^3] = \frac{M^3}{(M-2)(M-4)(M-6)\sigma_x^6}. \quad (47)$$

Portanto, o valor esperado do termo procurado é expresso por

$$E[\delta^{-2}(n)] = \frac{1}{N^2} \left[\frac{M^2}{(M-2)(M-4)\sigma_x^4} - \varepsilon \frac{M^3}{(M-2)(M-4)(M-6)\sigma_x^6} \right]. \quad (48)$$

REFERÊNCIAS

- [1] B. Widrow, J. McCool, M. Larimore, and C. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151-1162, Aug. 1976.
- [2] D. D. Falconer, "Adaptive reference echo cancellation," *IEEE Trans. Communications*, vol. 30, no. 9, pp. 2083-2094, Sep. 1982.
- [3] J. G. Proakis, *Digital Communication*, 2nd ed., New York: McGraw-Hill, 1989.
- [4] G. Long, F. Long, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 9, pp. 1397-1405, Sep. 1989.
- [5] G. Long, F. Long, and J. G. Proakis, "Corrections to the LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Signal Process.*, vol. 40, no. 1, pp. 230-232, Jan. 1992.
- [6] T. Wang and C. L. Wang, "Delayed least-mean-square algorithm," *IEE Electronics Letters*, vol. 31, no. 7, pp. 524-526, Mar. 1995.
- [7] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 8, pp. 793-806, Aug. 1986.
- [8] S. Haykin, *Adaptive Filter Theory*, 4th ed., Upper Saddle River, NJ: Prentice-Hall, 2002.
- [9] C. G. Samson and U. Reddy, "Fixed point error analysis of the normalized ladder algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 5, pp. 1177-1191, May 1983.
- [10] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Application*, John Wiley & Sons, 1998.
- [11] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3ed., McGraw-Hill, 1991.
- [12] E. M. Lobato, O. J. Tobias e R. Seara, "Modelo analítico do algoritmo eLMS no domínio transformado para sinais de entrada Gaussianos," *Anais do XXI Simpósio Brasileiro de Telecomunicações*, Belém, PA, CD-ROM, artigo nº 282, Set. 2004, pp. 1-6.
- [13] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 1, pp. 222-230, Jan. 1985.