

SOBRE O ALGORITMO ACELERADOR PARA FILTRAGEM ADAPTATIVA

Pablo E. Jojoa, Max Gerken, Felipe M. Pait

{pablo, mgk}@lcs.poli.usp.br, pait@lac.usp.br
Universidade de São Paulo - EPUSP - PTC
Av. Prof. Luciano Gualberto, travessa 3, n. 158
05508-900 São Paulo SP - Brasil

SUMÁRIO

O algoritmo Acelerador para filtragem adaptativa apresenta um compromisso mais favorável entre velocidade de convergência e variância da estimativa quando comparado com os algoritmos LMS ou NLMS. Isso é conseguido às custas de um moderado aumento na complexidade computacional. Neste trabalho o algoritmo Acelerador é interpretado como sendo do tipo quasi-Newton, sendo também apresentada uma versão simplificada que é de ajuste mais simples. Para esta versão é apresentada uma análise dos momentos de primeira e segunda ordens que resulta em expressões para o erro quadrático médio e o desajuste. Usando estas expressões, uma comparação simples com o algoritmo LMS mostra as potenciais vantagens do algoritmo Acelerador. São também apresentados resultados de simulações.

1. INTRODUÇÃO

Em [1] foi introduzido um algoritmo rápido de filtragem adaptativa de tempo discreto, derivado de um sintonizador de tempo contínuo que fixa a segunda derivada (aceleração) da estimativa dos parâmetros [2]. Em princípio um tal sintonizador tem aplicações em controle adaptativo. Mediante simulações foi mostrado em [1] que, para sinais de entrada coloridos, o algoritmo proposto pode apresentar em regime um compromisso mais favorável entre rapidez de convergência e erro de estimação que os algoritmos LMS e NLMS, uma propriedade obtida às custas de um moderado aumento na complexidade computacional. Foi também apresentada uma demonstração da convergência determinística.

Como já foi mencionado, o algoritmo Acelerador (AC) é baseado em um sintonizador de tempo contínuo que ajusta a segunda derivada, com respeito ao tempo, dos coeficientes do filtro. Igualmente, o popular

algoritmo LMS pode ser obtido de um sintonizador de tempo contínuo que ajusta a primeira derivada, com respeito ao tempo, dos coeficientes do filtro. Por outro lado, no algoritmo LMS utiliza-se uma estimativa da primeira derivada, com respeito aos coeficientes do filtro, de uma função custo. Assim pode-se suspeitar que no algoritmo AC se faz uso de uma estimativa da segunda derivada com respeito aos coeficientes do filtro. De fato, esta suspeita é correta e neste artigo é mostrado que o algoritmo AC pode ser interpretado como um método quasi-Newton, sendo apresentadas as aproximações do vetor gradiente e da matriz Hessiana utilizadas.

Em sua forma geral o algoritmo AC depende de quatro parâmetros, um escalar e três matrizes positivas definidas, que devem ser selecionados adequadamente. Se por um lado os parâmetros extras adicionam liberdade ao projetista do filtro adaptativo, por outro lado a seleção deles é uma dificuldade no uso do algoritmo AC, o que motiva a obtenção de novas versões que sejam mais fáceis de ajustar. Neste artigo é apresentada uma versão simplificada do algoritmo AC e os resultados de uma análise dos momentos de primeira e segunda ordens dos erros dos coeficientes do filtro. Com isso são obtidas expressões que caracterizam o comportamento transitório e em regime do erro quadrático médio e o desajuste. Usando estas expressões uma simples comparação com o algoritmo LMS é apresentada, mostrando as possíveis vantagens de usar o algoritmo AC. Por exemplo, mostra-se que para velocidades de convergência semelhantes o algoritmo AC apresenta um menor desajuste que o algoritmo LMS. Esta “atenuação” do desajuste pode ser tanto maior quanto mais mal condicionados são os sinais de entrada. Como resultado da comparação de desempenho com o bem conhecido algoritmo LMS, obtém-se algumas indicações de cunho prático para selecionar os parâmetros do algoritmo AC.

No texto a seguir é inicialmente apresentada uma versão simplificada do algoritmo AC. Em seguida é mostrado que o algoritmo AC, tanto na versão completa como na simplificada, pode ser considerado como

Este trabalho foi apoiado pela FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo e pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico.

sendo do tipo quasi-Newton. Posteriormente são apresentados alguns resultados da análise dos momentos de primeira e segunda ordens e a comparação com o algoritmo LMS. Sempre que apropriado são apresentados alguns resultados de simulações. Finaliza-se o artigo com uma seção de conclusões.

2. UMA VERSÃO SIMPLIFICADA DO ALGORITMO ACELERADOR

Da mesma forma que outros algoritmos de filtragem adaptativa, o algoritmo AC ajusta recursivamente uma estimativa $\mathbf{p}[n]$ de um vetor de coeficientes \mathbf{p}_o usando um sinal desejado $d[n] = \mathbf{x}^T[n]\mathbf{p}_o - \eta[n]$, com $\mathbf{x}[n]$ sendo o vetor de sinal de entrada (regressor) e $\eta[n]$ o ruído de medida. O objetivo é manter o erro de estimação *a posteriori* $e[n] = \mathbf{x}^T[n]\mathbf{p}[n] - d[n]$, o erro de estimação *a priori* $e_a[n] = \mathbf{x}^T[n]\mathbf{p}[n-1] - d[n]$, e o erro dos coeficientes $\delta\mathbf{p}[n] = \mathbf{p}[n] - \mathbf{p}_o$ tão pequenos quanto possível.

A forma original do algoritmo AC é dado por [1]:

$$e_a[n] = \mathbf{x}^T[n]\mathbf{p}[n-1] - d[n] \quad (1)$$

$$\mathbf{G}[n] = \mathbf{I} + \alpha^2\mathbf{M}_1\mathbf{x}[n]\mathbf{x}^T[n] + 2\alpha\mathbf{M}_1(\mathbf{M}_2 + \mathbf{x}[n]\mathbf{x}^T[n]\mathbf{M}_1\mathbf{M}_3) \quad (2)$$

$$\mathbf{q}[n] = (\mathbf{G}[n])^{-1}(\mathbf{q}[n-1] - \alpha\mathbf{M}_1\mathbf{x}[n]e_a[n]) \quad (3)$$

$$\mathbf{p}[n] = \mathbf{p}[n-1] + \alpha\mathbf{q}[n]. \quad (4)$$

Nas equações anteriores α é um escalar positivo, \mathbf{I} é a matriz identidade e \mathbf{M}_i , $i = 1, 2, 3$, são matrizes simétricas positivas definidas que devem satisfazer as seguintes desigualdades para garantir a estabilidade do algoritmo:

$$4\mathbf{M}_1\mathbf{M}_3\mathbf{M}_1 \geq \mathbf{M}_2^{-1} \quad (5)$$

$$\mathbf{M}_2\mathbf{M}_1\mathbf{M}_3 + \mathbf{M}_3\mathbf{M}_1\mathbf{M}_2 \geq \mathbf{M}_1^{-1}/2. \quad (6)$$

Em [1] foi mostrado que a atualização (3) de $\mathbf{q}[n]$ pode ser feita da seguinte forma:

$$\mathbf{C}[n] = \frac{\alpha e_a[n] + \mathbf{x}^T[n]\mathbf{B}\mathbf{M}_1^{-1}\mathbf{q}[n-1]}{1 + \mathbf{x}^T[n]\mathbf{B}\mathbf{x}[n]}\mathbf{M}_1$$

$$\mathbf{q}[n] = \mathbf{A}(\mathbf{q}[n-1] - \mathbf{C}[n]\mathbf{x}[n]),$$

com

$$\mathbf{A} = (\mathbf{I} + 2\alpha\mathbf{M}_1\mathbf{M}_2)^{-1},$$

$$\mathbf{B} = \alpha(\alpha\mathbf{I} + 2\mathbf{M}_1\mathbf{M}_3)\mathbf{A}\mathbf{M}_1.$$

Motivado pelo fato de que a maior rapidez de convergência é obtida no limite superior de (5) e (6), são introduzidas as relações $\mathbf{M}_1\mathbf{M}_2 = \mathbf{I}/(2\gamma)$ e $\mathbf{M}_1\mathbf{M}_3 = \gamma\mathbf{I}/2$. Assim, para um escalar positivo γ e uma matriz definida positiva \mathbf{M}_1 os limites superiores de (5) e (6)

são sempre atingidos. Com estas considerações obtém-se uma versão simplificada do algoritmo AC que é dada por:

$$\begin{aligned} e_a[n] &= \mathbf{x}^T[n]\mathbf{p}[n-1] - d[n] \\ g[n] &= \frac{e_a[n] + \gamma\mathbf{x}^T[n]\mathbf{q}[n-1]}{1 + \alpha\gamma\mathbf{x}^T[n]\mathbf{M}_1\mathbf{x}[n]} \\ \mathbf{q}[n] &= \frac{\gamma}{\alpha + \gamma}(\mathbf{q}[n-1] - \alpha g[n]\mathbf{M}_1\mathbf{x}[n]) \\ \mathbf{p}[n] &= \mathbf{p}[n-1] + \alpha\mathbf{q}[n]. \end{aligned} \quad (7)$$

Uma simplificação adicional pode ser introduzida impondo $\mathbf{M}_1 = m_1\mathbf{I}$, sendo \mathbf{I} a matriz identidade. Deve-se lembrar que, em conformidade com os resultados já mencionados, o algoritmo acima será estável para parâmetros α , γ e m_1 positivos (ou \mathbf{M}_1 positiva definida).

3. O ALGORITMO ACELERADOR E O MÉTODO DE NEWTON

Dada uma função a ser minimizada $f(\mathbf{p})$, no método de Newton aproxima-se esta função localmente por uma função quadrática e determina-se o mínimo da aproximação e não da função original [3]. Perto de $\mathbf{p}_{n-1} = \mathbf{p}[n-1]$ podemos aproximar $f(\mathbf{p})$ pela série truncada de Taylor

$$\begin{aligned} f(\mathbf{p}) &\simeq f(\mathbf{p}_{n-1}) + \nabla f^T(n, \mathbf{p}_{n-1})(\mathbf{p} - \mathbf{p}_{n-1}) \\ &\quad + \frac{1}{2}(\mathbf{p} - \mathbf{p}_{n-1})^T \mathbf{F}(n, \mathbf{p}_{n-1})(\mathbf{p} - \mathbf{p}_{n-1}), \end{aligned} \quad (8)$$

sendo $\nabla f(\cdot, \cdot)$ o vetor gradiente definido como um vetor coluna M -dimensional e $\mathbf{F}(\cdot, \cdot)$ a matriz Hessiana de $f(\mathbf{p})$. No algoritmo de Newton $\mathbf{p}[n]$ é ajustado para minimizar o lado direito de (8) da seguinte forma:

$$\mathbf{p}_n = \mathbf{p}_{n-1} - \mathbf{F}^{-1}(n, \mathbf{p}_{n-1})\nabla f(n, \mathbf{p}_{n-1}). \quad (9)$$

É ilustrativo considerar a minimização da função custo quadrática $f(\mathbf{p}) = E\{\mathbf{e}^2[n]\}/2$, sendo $E\{\cdot\}$ o operador esperança e $\mathbf{e}[n] = \mathbf{x}^T[n]\mathbf{p} - d[n]$. Considerando uma aproximação como em (8) obtém-se:

$$\mathbf{F}(n, \mathbf{p}_{n-1}) = \mathbf{R}_x = E\{\mathbf{x}[n]\mathbf{x}^T[n]\} \quad (10)$$

$$\nabla f(n, \mathbf{p}_{n-1}) = \mathbf{R}_x\mathbf{p}[n-1] - E\{d[n]\mathbf{x}[n]\} \quad (11)$$

Substituindo estas equações em (9) resulta

$$\mathbf{p}[n] = \mathbf{R}_x^{-1}E\{d[n]\mathbf{x}[n]\},$$

observando-se deste modo que neste caso o algoritmo de Newton converge em uma única iteração.

Embora as propriedades de convergência do método de Newton sejam muito atrativas, o cálculo da inversa da matriz Hessiana não é adequado para uma implementação prática. A idéia básica dos métodos

quasi-Newton é usar aproximações da inversa da matriz Hessiana. A primeira modificação na direção de um método quasi-Newton é usualmente a introdução de um passo de adaptação ρ_n :

$$\mathbf{p}_n = \mathbf{p}_{n-1} - \rho_n \mathbf{F}^{-1}(n, \mathbf{p}_{n-1}) \nabla f(n, \mathbf{p}_{n-1}). \quad (12)$$

Para verificar que o algoritmo AC pode ser colocado na forma (12) utilizando aproximações para a inversa da matriz Hessiana e para o vetor gradiente basta combinar as expressões (3) e (4). Assim obtém-se

$$\begin{aligned} \mathbf{p}[n] &= \mathbf{p}[n-1] + \alpha (\mathbf{G}[n])^{-1} \{ \mathbf{q}[n-1] - \alpha \mathbf{M}_1 \mathbf{x}[n] e_a[n] \} \\ &= \mathbf{p}[n-1] - (\mathbf{G}[n])^{-1} \{ \alpha^2 \mathbf{M}_1 \mathbf{x}[n] e_a[n] + \\ &\quad - (\mathbf{p}[n-1] - \mathbf{p}[n-2]) \}. \end{aligned} \quad (13)$$

Comparando (13) com (12) pode-se concluir imediatamente que o algoritmo AC faz uso das seguintes estimativas:

$$\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = \beta (\alpha^2 \mathbf{M}_1 e_a[n] \mathbf{x}[n] + \mathbf{p}[n-1] - \mathbf{p}[n-2]) \quad (14)$$

$$\widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \beta \mathbf{G}[n], \quad (15)$$

sendo β uma constante positiva adequada. Considerando a versão simplificada com $\mathbf{M}_1 = m_1 \mathbf{I}$ e $\beta = \gamma / (\alpha + \gamma)$ resultam expressões mais familiares:

$$\rho_n = \frac{\alpha}{\alpha + \gamma} \alpha \gamma m_1 \quad (16)$$

$$\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = e_a[n] \mathbf{x}[n] + \frac{1}{\alpha^2 m_1} (\mathbf{p}[n-1] - \mathbf{p}[n-2]) \quad (17)$$

$$\widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \mathbf{I} + \alpha \gamma m_1 \mathbf{x}[n] \mathbf{x}[n]^T. \quad (18)$$

A expressão (17) é uma estimativa instantânea do vetor gradiente $E\{e_a[n] \mathbf{x}[n]\}$ “corrigida” por um termo proporcional ao incremento do coeficiente anterior, e (18) é uma estimativa instantânea regularizada da matriz Hessiana $\mathbf{R}_x = E\{\mathbf{x}[n] \mathbf{x}[n]^T\}$. É ilustrativo observar que para o algoritmo LMS vale $\rho_n = \mu$ (o passo de adaptação), $\widehat{\nabla} f(n, \mathbf{p}_{n-1}) = e_a[n] \mathbf{x}[n]$ e $\widehat{\mathbf{F}}(n, \mathbf{p}_{n-1}) = \mathbf{I}$.

Para ilustrar a natureza Newton do algoritmo AC será considerado um exemplo simples de um filtro preditor de dois coeficientes excitado por um sinal senoidal $\sin((5\pi/128)n + 0,4\pi)$ e sujeito às seguintes condições: coeficientes iniciais $-1,5$ e $-1,7$ e ruído de medida branco, Gaussiano, com variância 10^{-4} . Os coeficientes ótimos são 2 e -1 . São considerados três algoritmos: LMS ($\mu = 0,03$), Newton ($\rho_n = 0,05$; ver Eq. (10), (11) e (12)) e AC ($\alpha = 0,125$, $m_1 = 0,09$, $\gamma = 15$ e $\gamma = 320$). Os passos de adaptação e os parâmetros de todos os algoritmos, exceto o de Newton, foram ajustados para produzir o mesmo desvio padrão dos

coeficientes (0,002) depois da convergência. A Fig. 1 apresenta as trajetórias no plano dos coeficientes onde também estão representadas as curvas de contorno da função custo. Dois comportamentos diferentes do algoritmo AC são mostrados, um deles similar ao do algoritmo LMS (AC1 com $\gamma = 15$) e outro similar ao algoritmo de Newton (AC2 com $\gamma = 320$). Neste último caso, considerando o espaço dos coeficientes, a convergência ocorre aproximadamente sobre uma reta ligando os coeficientes iniciais aos coeficientes ótimos. Fica assim evidenciado que, dependendo do valor dos parâmetros α , m_1 e γ , o algoritmo acelerador pode apresentar comportamentos bastante diversos. Pode causar uma certa surpresa que quando o algoritmo AC apresenta um comportamento como o algoritmo de Newton, a velocidade de convergência seja menor do que quando o comportamento é semelhante ao algoritmo LMS. Uma possível explicação para este comportamento é que os parâmetros do algoritmo AC devem estar ajustados de tal forma que um maior número de iterações ocorra para que boas estimativas do gradiente e da inversa da matriz Hessiana sejam obtidas.

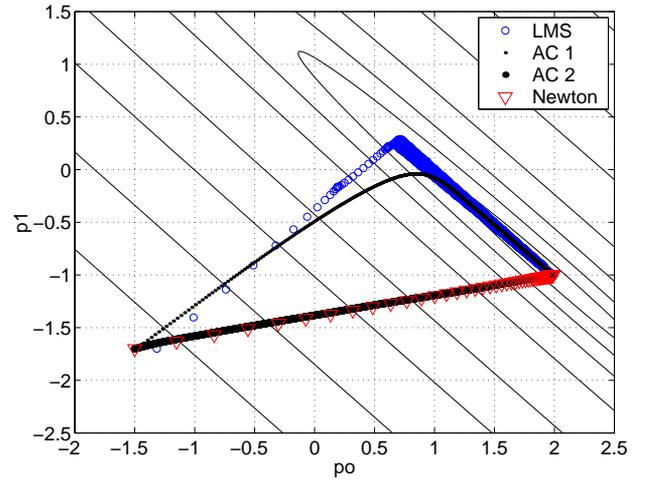


Fig. 1. Trajetórias no plano dos coeficientes para um preditor com dois coeficientes. AC1: $\gamma = 15$, AC2: $\gamma = 320$.

Fechando esta seção segue um resumo das principais características do algoritmo AC tratadas até agora:

- (i) O algoritmo AC minimiza a função $E\{e^2[n]\}$.
- (ii) Em cada passo considera-se uma aproximação local da função custo como a dada em (8) usando estimativas não triviais para o vetor gradiente e para a matriz Hessiana, dadas respectivamente por (14) e (15), ou por (17) e (18) para a versão simplificada.

- (iii) No sentido de (i) e (ii) o algoritmo AC pode ser classificado como um método quasi-Newton.
- (iv) Conforme mostrado em [1] o algoritmo AC é estável, o que significa que com as estimativas utilizadas do vetor gradiente e da matriz Hessiana a convergência fica assegurada.

4. ANÁLISE DOS MOMENTOS DE PRIMEIRA E SEGUNDA ORDENS

Observando que $d[n] = \mathbf{x}^T[n] \mathbf{p}_o - \eta[n]$ e $\delta \mathbf{p}[n] = \mathbf{p}[n] - \mathbf{p}_o$ pode-se escrever

$$\begin{aligned} e_a[n] &= \mathbf{x}^T[n] \mathbf{p}[n-1] - d[n] \\ &= \mathbf{x}^T \delta \mathbf{p}[n-1] + \eta[n], \end{aligned}$$

com $\eta[n]$ sendo o ruído de medida. Introduzindo a variável $\mathbf{r}[n] = \delta \mathbf{p}[n] + \gamma \mathbf{q}[n]$ nas Equações (7) resulta

$$\mathbf{r}[n] = \mathbf{\Gamma}[n] \mathbf{r}[n-1] + \alpha \gamma m_1 \mathbf{\Gamma}[n] \mathbf{x}[n] \eta[n] \quad (19)$$

$$\delta \mathbf{p}[n] = a \delta \mathbf{p}[n-1] + b \mathbf{r}[n], \quad (20)$$

com

$$\mathbf{\Gamma}[n] = \mathbf{I} - \frac{\alpha \gamma m_1 \mathbf{x}[n] \mathbf{x}^T[n]}{1 + \alpha \gamma m_1 \mathbf{x}^T[n] \mathbf{x}[n]}$$

e

$$a = 1 - b = \gamma / (\alpha + \gamma).$$

Com estas equações pode-se realizar uma análise dos momentos de primeira e segunda ordens de $\mathbf{r}[n]$ e $\delta \mathbf{p}[n]$, semelhante à análise usual dos momentos do algoritmo LMS [4, 6]. Para isso faz-se uso das seguintes suposições simplificadoras:

- $\mathbf{x}[n]$ é um vetor de ruído branco Gaussiano de média nula com matriz de covariância

$$\mathbf{R}_x = E\{\mathbf{x}[n] \mathbf{x}^T[n]\}.$$

- $\eta[n]$ é um ruído branco Gaussiano de média nula e variância σ_η^2 e estatisticamente independente de $\mathbf{x}[n]$.
- A ordem M do filtro adaptativo é grande o suficiente para fazer a variável $\mathbf{x}^T[n] \mathbf{x}[n]$ se comportar de forma “lenta” em comparação com $\mathbf{x}[n]$.

Esta última suposição possibilita a aplicação da propriedade das médias (“averaging property”) [5] para obter

$$\begin{aligned} E\{\mathbf{\Gamma}[n]\} &\approx \mathbf{I} - \frac{\alpha \gamma m_1 E\{\mathbf{x}[n] \mathbf{x}^T[n]\}}{1 + \alpha \gamma m_1 E\{\mathbf{x}^T[n] \mathbf{x}[n]\}} \\ &= \mathbf{I} - \frac{\alpha \gamma m_1}{1 + \alpha \gamma m_1 \text{tr}\{\mathbf{R}_x\}} \mathbf{R}_x = \mathbf{I} - \mu_a \mathbf{R}_x, \end{aligned} \quad (21)$$

com $\text{tr}\{\cdot\}$ representando o traço da matriz e

$$\mu_a = \frac{\alpha \gamma m_1}{1 + \alpha \gamma m_1 \text{tr}\{\mathbf{R}_x\}}. \quad (22)$$

Nestas condições é possível mostrar que:

- Os momentos de primeiro ordem de $\mathbf{r}[n]$ e $\delta \mathbf{p}[n]$ satisfazem

$$E\{\mathbf{r}[n]\} = (\mathbf{I} - \mu_a \mathbf{R}_x) E\{\mathbf{r}[n-1]\} \quad (23)$$

$$E\{\delta \mathbf{p}[n]\} = a E\{\delta \mathbf{p}[n-1]\} + b E\{\mathbf{r}[n]\} \quad (24)$$

- As matrizes de covariância $\mathcal{R}[n] = E\{\mathbf{r}[n] \mathbf{r}^T[n]\}$ e $\mathcal{P}[n] = E\{\delta \mathbf{p}[n] \delta \mathbf{p}^T[n]\}$ satisfazem

$$\begin{aligned} \mathcal{R}[n] &= (\mathbf{I} - \mu_a \mathbf{R}_x) \mathcal{R}[n-1] (\mathbf{I} - \mu_a \mathbf{R}_x) \\ &\quad + \mu_a^2 \mathbf{R}_x \sigma_\eta^2 \end{aligned} \quad (25)$$

$$\begin{aligned} \mathcal{P}[n] &= a^2 \mathcal{P}[n-1] + b^2 \mathcal{R}[n] \\ &\quad + ab E\{\delta \mathbf{p}[n-1] \mathbf{r}^T[n]\} \\ &\quad + ab E\{\mathbf{r}[n] \delta \mathbf{p}^T[n-1]\} \\ &\quad + \frac{2\alpha}{\gamma} a^2 E\{\delta \mathbf{p}[n-1]\} E\{\mathbf{r}^T[n-1]\} \end{aligned} \quad (26)$$

- O erro quadrático médio $E\{e_a^2[n]\}$ satisfaz

$$E\{e_a^2[n]\} = \text{tr}\{\mathbf{R}_x \mathcal{P}[n-1]\} + \sigma_\eta^2. \quad (27)$$

Com $n \rightarrow \infty$ obtém-se das equações (23) a (27) o traço da matriz de covariância (em regime) dos erros dos coeficientes

$$\text{tr}\{\mathcal{P}_\infty\} = \sigma_\eta^2 \frac{\alpha}{\alpha + 2\gamma} \sum_{k=1}^M \frac{\mu_a}{2 - \mu_a \lambda_k} \quad (28)$$

e o erro quadrático médio de estimação após a convergência

$$E\{e_a^2\} = \sigma_\eta^2 + \underbrace{\sigma_\eta^2 \frac{\alpha}{\alpha + 2\gamma} \sum_{k=1}^M \frac{\mu_a \lambda_k}{2 - \mu_a \lambda_k}}_{\mathcal{M}_a}, \quad (29)$$

com λ_k sendo os autovalores de \mathbf{R}_x . Observando-se as Equações (23) e (25) conclui-se que no caso do algoritmo AC a constante μ_a faz o mesmo papel do passo de adaptação μ do algoritmo LMS (ver Cap. 9 de [6]). Esse fato leva à denominação de “passo de adaptação equivalente” para a constante μ_a . Da Equação (29) observa-se que, para um passo de adaptação equivalente μ_a igual ao passo de adaptação do algoritmo LMS ($\mu = \mu_a$), o desajuste do algoritmo AC é igual ao desajuste do algoritmo LMS (ver Cap. 9 de [6]) atenuado por um fator

$$\alpha / (\alpha + 2\gamma) < 1. \quad (30)$$

Por outra lado, a matriz de covariância do erro dos coeficientes do algoritmo LMS satisfaz a mesma equação de diferenças que $\mathcal{R}[n]$. Portanto, desde que as desigualdades

$$\gamma / (\alpha + \gamma) < 1 - \mu_a \lambda_{\min}$$

ou

$$\alpha/(\alpha + \gamma) > \mu_a \lambda_{\min} \quad (31)$$

estejam satisfeitas, $\mathcal{P}[n]$ em (26) tem essencialmente o mesmo comportamento de convergência que $\mathcal{R}[n]$ em (25). Deste modo, com esta condição satisfeita, pode-se esperar que o algoritmo AC tenha aproximadamente a mesma rapidez de convergência que o LMS com passo de adaptação $\mu = \mu_a$. Lembrando que para o algoritmo LMS $\mu \lambda_{\min} < 2\lambda_{\min}/\lambda_{\max}$, conclui-se que a atenuação do desajuste do AC sobre o algoritmo LMS pode ser aumentado proporcionalmente ao mal condicionamento da matriz \mathbf{R}_x mantendo-se a rapidez de convergência constante.

Para verificar até onde o raciocínio anterior é válido foram realizadas algumas simulações. Foi considerada a identificação de um sistema FIR com $M = 6$ coeficientes iguais a 1. Como sinal de entrada foi tomado um sinal Gaussiano colorido de média nula e variância σ_x^2 . Como ruído de medida foi considerado um ruído Gaussiano branco de média nula e variância σ_η^2 . Os valores utilizados estão relacionados na Tabela 1.

Na Figura 2 é mostrada a influência de $\gamma/(\alpha + \gamma)$ sobre a rapidez de convergência do algoritmo AC. Os correspondentes valores dos parâmetros são apresentados na Tabela 1. A comparação dos valores de $\gamma/(\alpha + \gamma)$ e $1 - \mu \lambda_{\min}$ e as curvas do erro quadrático médio correspondentes confirmam o comportamento esperado.

Na Tabela 2 são comparados os desajustes medi-

m_1	γ	$\frac{\gamma}{\alpha + \gamma}$	$1 - \mu \lambda_{\min}$	Fig.2
60	0,01	0,242	0,987	AC 1
6	0,1	0,762	0,987	AC 2
0,6	1	0,970	0,987	AC 3
$\sigma_x^2 = 4, \sigma_\eta^2 = 0,5, \alpha = 0,0313, \mu_a = \mu = 0,0129$				

Tabela 1. Parâmetros da Figura 2.

dos dos algoritmos AC e LMS e a atenuação de desajuste teórica $\alpha/(\alpha + 2\gamma)$ dada pela Equação (29). Observamos que aumentando α a atenuação do de-

α	$\alpha/(\alpha + 2\gamma)$	\mathcal{M}_a	\mathcal{M}_{lms}	$\mu = \mu_a$
α_o	0,135	0,075	0,102	0,013
$2\alpha_o$	0,238	0,130	0,202	0,020
$4\alpha_o$	0,347	0,190	0,536	0,025
$\sigma_x^2 = 4, \sigma_\eta^2 = 0,5, \gamma = 0,01, m_1 = 6, \alpha_o = 0,0313$				

Tabela 2. Desajuste dos algoritmos AC (\mathcal{M}_a) e LMS (\mathcal{M}_{lms}). Média sobre 200 experimentos.

sajuste alcançado é bastante próxima do valor teórico, enquanto que para valores baixos de α a atenuação de

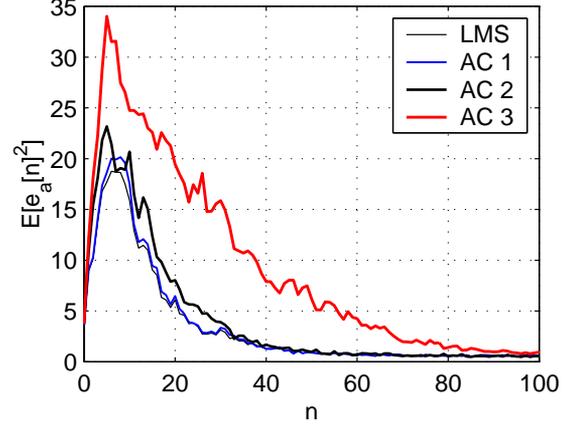


Fig. 2. Curvas de aprendizagem dos algoritmos AC e LMS com parâmetros na Tabela 1. Média de 200 experimentos.

desajuste alcançada não é tão boa como seria de se esperar. A Tabela 2 é apresentada como um exemplo deste efeito que foi observado também em outras simulações. Na Tabela 2 observa-se que as reduções de desajuste alcançadas não são tão grandes como as previstas pela expressão (29). Embora os resultados obtidos sobre o desajuste indiquem um comportamento qualitativo correto há um desvio entre os resultados do modelo obtido e os resultados das simulações. Este desvio se deve provavelmente às hipóteses simplificadas utilizadas. Por exemplo, a suposição de que $\mathbf{x}(n)$ seja branco não corresponde a realidade. Aqui uma análise mais sofisticada dos momentos de primeira e segunda ordens se faz necessária.

Por outro lado, deve-se também observar que para passos de adaptação μ para o quais o comportamento do algoritmo LMS é ruim, o algoritmo AC é estável e converge. Um exemplo é apresentado na Figura 3. As

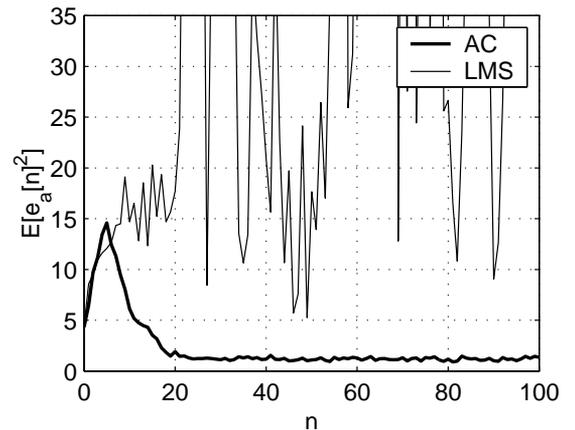


Fig. 3. Curvas de aprendizagem dos algoritmos AC e LMS com $\mu = \mu_a$. Média sobre 200 experimentos.

curvas de aprendizagem mostradas foram obtidas nas seguintes condições: $m_1 = 460$, $\alpha = 0,3125$, $\gamma = 0,1$, $\mu = 0,0416$, $\sigma_x^2 = 4$ e $\sigma_\eta^2 = 0,5$. O condicionamento da matriz de correlação \mathbf{R}_x é dado por $\lambda_{\max}/\lambda_{\min} \approx 6,5$. O desajuste obtido para o algoritmo AC foi $\mathcal{M}_a = 0,64$, enquanto que para o algoritmo LMS obteve-se $\mathcal{M}_a = 3,7 \cdot 10^6$.

Dados estes resultados pode-se agora estabelecer um procedimento para o ajuste dos parâmetros do algoritmo AC. Supõe-se que é desejada uma velocidade de convergência correspondente a um passo de adaptação μ do algoritmo LMS e que se conheça a matriz de covariância \mathbf{R}_x do sinal de entrada ou pelo menos sua variância σ_x^2 e uma estimativa de λ_{\min} , o autovalor mínimo de \mathbf{R}_x . Nesta situação um procedimento simples para a obtenção dos parâmetros do algoritmo AC é o seguinte:

- Toma-se $\mu_a = \mu$ e determina-se o produto $\alpha\gamma m_1$ utilizando-se por exemplo que $\text{tr}\{\mathbf{R}_x\} = M\sigma_x^2$.
- Determina-se α , γ e m_1 satisfazendo a condição (31) com uma certa folga para garantir que a velocidade de convergência será a desejada e procurando obter um valor baixo para a atenuação $\alpha/(\alpha + \gamma)$ do desajuste.

Como mostra o exemplo da Figura 3 o valor de μ pode ser tal que o algoritmo LMS seja em princípio rápido mas não apresente um bom comportamento para o sinal de entrada em questão. Nesta situação é que o algoritmo AC apresentará as maiores vantagens sobre o algoritmo LMS.

5. CONCLUSÃO

Foi mostrado que o algoritmo acelerador é um algoritmo do tipo quasi-Newton apresentando-se as estimativas de gradiente e matriz Hessiana que utiliza. Além disso, foi introduzida uma versão simplificada do algoritmo AC que é mais simples de ajustar do que a sua forma geral. Por meio de uma análise de momentos de primeira e segunda ordens foram obtidas expressões para o erro quadrático médio e para o desajuste. Estes resultados foram confirmados qualitativamente pelos resultados de simulações havendo um certo desvio quantitativo. Esta discrepância indica que algumas hipóteses simplificadoras feitas na análise dos momentos devem ser abandonadas em estudos futuros. De qualquer forma, os resultados obtidos fornecem indicações esclarecedoras que facilitam o ajuste dos parâmetros do algoritmo AC. Além disso, o comportamento robusto de convergência do algoritmo proposto faz dele uma boa alternativa para aplicações onde sinais de entrada são mal condicionados e o algoritmo LMS apresenta problemas de convergência.

6. BIBLIOGRAFIA

- [1] M. Gerken, F.M. Pait, P.E. Jojoa Gomez: An Adaptive Algorithm with Parameter Acceleration, *Proceedings of ICASSP'2000*, Vol I, April 2000, pp. 17-20.
- [2] F.M. Pait: A Tuner that Accelerates Parameters, *Systems & Control Letters* 35 (1998) 65-68.
- [3] D.G. Luenberger: *Linear and Nonlinear Programming*, 2. Ed., Addison-Wesley, 1989.
- [4] V. Solo, X. Kong: *Adaptive Signal Processing Algorithms, Stability and Performance*, Prentice Hall, 1995.
- [5] C. Samson, V.U. Reddy: Fixed point error analysis of the normalized ladder algorithms, *IEEE Trans. on ASSP*, vol. 31, n. 5, October 1983, pp. 1177-1191.
- [6] S. Haykin: *Adaptive Filter Theory*, 3. Ed., Prentice Hall, 1996.