

Algoritmo IPNLMS com Parâmetro de Proporcionalidade Ótimo

César Augusto Santana Castelo Branco e Francisco das Chagas de Souza

Resumo—Este trabalho propõe um procedimento para otimizar a escolha do parâmetro de proporcionalidade do algoritmo adaptativo IPNLMS (*improved proportionate normalized least-mean-square*). A abordagem proposta utiliza um método de otimização linear denominado razão áurea, o qual realiza a busca do parâmetro de proporcionalidade ótimo em cada iteração do processo de adaptação do IPNLMS. Esta estratégia considera a minimização de uma função objetivo baseada no erro quadrático *a posteriori*. Resultados de simulações mostram que a metodologia proposta melhora o desempenho do algoritmo IPNLMS em termos de velocidade de convergência para diversos cenários em que a resposta ao impulso da planta é esparsa.

Palavras-Chave—Algoritmo IPNLMS, otimização, razão áurea, identificação de plantas esparsas.

Abstract—This work proposes a procedure to optimize the selection of the proportionality parameter of the improved proportionate normalized least-mean-square (IPNLMS) adaptive algorithm. The proposed approach utilizes a linear optimization method called golden section, which searches the optimal proportionality parameter at each iteration of the adaptation process of the IPNLMS. This strategy considers the minimization of an objective function based on the *a posteriori* squared error. Simulation results show that the proposed methodology improves the performance of the IPNLMS algorithm in terms of convergence speed for several scenarios in which the plant impulse response is sparse.

Keywords—IPNLMS algorithm, optimization, golden section, sparse plant identification.

I. INTRODUÇÃO

Filtros adaptativos desempenham um papel fundamental na resolução de vários problemas da área de processamento de sinais, tais como previsão, equalização de canais de comunicação e identificação de sistemas [1]-[4]. A Fig. 1 ilustra a aplicação de um filtro adaptativo em um problema de identificação de sistemas, onde $x(n)$ é o sinal de entrada, $d(n)$ é o sinal desejado, $y(n)$ é a saída do filtro, $e(n)$ é o sinal de erro e $z(n)$ é o ruído de medição. O algoritmo adaptativo, o qual possui como entradas $x(n)$ e $e(n)$, modifica a resposta ao impulso do filtro com o objetivo de minimizar a diferença entre $y(n)$ e $d(n)$. Quando apenas uma pequena parcela da resposta ao impulso da planta é formada por pesos não nulos, diz-se que a mesma é esparsa [5]. Essas plantas podem ser encontradas em diversas aplicações de engenharia, tais como cancelamento de eco em telecomunicações [6], estimação de harmônicas em sistemas elétricos de potência [7], processos de natureza química e sísmica [4], etc.

Dentre as alternativas para lidar com plantas esparsas, tem-se os algoritmos adaptativos proporcionais como o LMS normalizado proporcional (PNLMS - *proportionate normalized*

César Augusto Santana Castelo Branco e Francisco das Chagas de Souza, LSAPS - Laboratório de Sistemas Adaptativos e Processamento de Sinais, Departamento de Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís-MA, Brasil, e-mails: santana_castelo@hotmail.com; francisco.souza@ufma.br.

Este trabalho foi parcialmente financiado pela Fundação de Amparo à Pesquisa e Desenvolvimento Científico do Maranhão (FAPEMA), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

least-mean-square) [8] e o PNLMS melhorado (IPNLMS - *improved PNLMS*) [9]. Estes algoritmos alcançam um desempenho superior ao LMS e NLMS, os quais apresentam desempenho pobre em termos de velocidade de convergência para ambientes esparsos [3]-[4], [8]-[12]. No entanto, os algoritmos proporcionais PNLMS e IPNLMS apresentam baixa velocidade de convergência ao lidarem com plantas dispersivas [4]. Para lidar com plantas com característica dispersiva ou de esparsidade variável, tem-se os algoritmos PNLMS++ [10], SC-PNLMS e SC-IPNLMS [11]. O algoritmo PNLMS++ alterna, a cada iteração, entre as regras de atualização dos algoritmos PNLMS e NLMS. Já os algoritmos SC-PNLMS e SC-IPNLMS empregam estimativas da esparsidade da planta em suas respectivas regras de atualização. Outros trabalhos correlatos podem ser encontrados em [5] e [13].

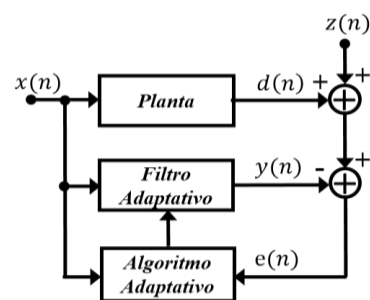


Fig. 1. Identificação de sistemas utilizando filtro adaptativo

O algoritmo IPNLMS foi construído a partir de modificações na função de proporcionalidade do PNLMS. O principal objetivo dessas modificações foi tornar mais “suave” a escolha do ganho individual para cada peso do filtro adaptativo e, conseqüentemente, explorar melhor a filosofia proporcional originária do PNLMS. Embora também possa ser apresentado como uma alternativa eficiente para plantas de esparsidade média, o IPNLMS não é capaz de atingir a mesma velocidade inicial de convergência do PNLMS para plantas com resposta ao impulso altamente esparsas [14]. Além disso, o desempenho do IPNLMS depende da escolha do parâmetro de proporcionalidade, o qual é constante durante todo o processo de adaptação. Resultados experimentais demonstram que 0,0, -0,5 e -0,75 são boas escolhas para tal parâmetro [11].

Sabe-se que a esparsidade de uma planta pode variar devido a fatores como temperatura, pressão, dentre outros [11]. Um valor de parâmetro de proporcionalidade constante pode comprometer o desempenho do IPNLMS ao lidar com plantas de esparsidade variável. Além disso, o valor desse parâmetro deve ser escolhido de acordo com o grau de esparsidade da planta. Dessa forma, um valor inapropriado para tal parâmetro pode degradar o desempenho do IPNLMS.

Neste trabalho propõe-se um procedimento para otimizar a escolha do parâmetro de proporcionalidade do algoritmo IPNLMS, utilizando uma técnica de otimização linear, denominada razão áurea [15], para encontrar o valor ótimo deste parâmetro em cada iteração do processo de adaptação. A função objetivo a ser minimizada é baseada no erro *a posteriori* [16].

O algoritmo IPNLMS proposto é aplicado em um problema de identificação de plantas esparsas. Resultados de simulações

mostram que a metodologia proposta apresenta maior velocidade de convergência bem como resposta mais rápida a perturbações na planta em relação aos algoritmos IPNLMS e SC-IPNLMS.

II. REVISITANDO OS ALGORITMOS PNLMS E IPNLMS

A. Algoritmo PNLMS Padrão

O algoritmo PNLMS padrão pode ser descrito pelo seguinte conjunto de equações [4]:

Atualização dos pesos ($N \times 1$)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T \mathbf{G}(n) \mathbf{x}(n) + \delta}. \quad (1)$$

Erro a priori

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

Matriz de distribuição de ganhos ($N \times N$)

$$\mathbf{G}(n) = \text{diag}[g_1(n) \quad g_2(n) \quad \dots \quad g_N(n)]. \quad (3)$$

Ganho individual

$$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^N \phi_j(n)}. \quad (4)$$

Função de proporcionalidade

$$\phi_i(n) = \max[f(n), |w_i(n)|]. \quad (5)$$

Fator de ativação

$$f(n) = \rho \max[\xi, \|\mathbf{w}(n)\|_\infty] \quad (6)$$

onde $0 < \mu < 2$ é o parâmetro de passo, $\delta > 0$ é um parâmetro de regularização responsável por estabilizar a solução, $\mathbf{x}(n)$ e $\mathbf{w}(n)$ são os vetores de entrada e de pesos do filtro adaptativo, respectivamente, e N é o tamanho do filtro. A matriz $\mathbf{G}(n)$ distribui os ganhos $g_i(n)$ entre os pesos, controlando a energia de adaptação do filtro. O fator de ativação $f(n)$ depende do vetor de pesos do filtro bem como de ξ e ρ , os quais são os parâmetros de inicialização e de proporcionalidade, respectivamente [8]. Conforme discutido em [4], este fator desempenha um importante papel no processo de adaptação do algoritmo PNLMS.

B. Derivação do Algoritmo IPNLMS

O algoritmo IPNLMS pode ser derivado a partir do PNLMS padrão [17]. Para este fim, em (6), substitui-se $\|\mathbf{w}(n)\|_\infty$ pela norma-1 do vetor de pesos, $\|\mathbf{w}(n)\|_1$, e adota-se $\rho = 1/N$. Assim, tem-se um novo fator de ativação dado por

$$f(n) = \begin{cases} \frac{1}{N} \max[\xi, \|\mathbf{w}(n)\|_1], & n = 0 \\ \frac{\|\mathbf{w}(n)\|_1}{N}, & n \geq 1. \end{cases} \quad (7)$$

Desta forma, para $n \geq 1$, (5) pode ser reescrita como

$$\phi_i(n) = \max\left[\frac{\|\mathbf{w}(n)\|_1}{N}, |w_i(n)|\right]. \quad (8)$$

Agora, para a função de proporcionalidade dada por (8), uma média ponderada é usada no lugar do operador $\max[\cdot]$. Logo, a nova função de proporcionalidade é [9]

$$\phi_i(n) = (1 - \alpha) \frac{\|\mathbf{w}(n)\|_1}{N} + (1 + \alpha) |w_i(n)| \quad (9)$$

onde $-1 \leq \alpha < 1$ é um fator de ponderação. Note que a primeira parcela de (9) é comum a todos os pesos do filtro, enquanto que a segunda é proporcional à magnitude do i -ésimo peso.

Agora, considerando (9), desenvolve-se o denominador de (4) para obter

$$\sum_{i=1}^N \phi_i(n) = 2\|\mathbf{w}(n)\|_1. \quad (10)$$

Então, substituindo (9) e (10) em (4), obtém-se o ganho individual do algoritmo IPNLMS como sendo

$$g_i(n) = (1 - \alpha) \frac{1}{2N} + (1 + \alpha) \frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varepsilon} \quad (11)$$

onde $\varepsilon > 0$ é um parâmetro de regularização usado para evitar divisão por zero. O fator α é denominado parâmetro de proporcionalidade do algoritmo IPNLMS.

A regra de atualização de pesos do algoritmo IPNLMS possui a mesma estrutura de (1), sendo que os elementos da matriz $\mathbf{G}(n)$ são calculados por (11). A seguir é apresentada a metodologia proposta para otimizar a escolha do parâmetro de proporcionalidade do IPNLMS.

III. ABORDAGEM PROPOSTA

Esta seção descreve a metodologia proposta para otimizar a escolha do parâmetro α do algoritmo IPNLMS. Primeiramente, é feito um estudo sobre o impacto de α no comportamento da função objetivo considerada, a qual é baseada no erro *a posteriori*. Logo após, o método da razão áurea é apresentado. Por fim, o algoritmo IPNLMS com parâmetro α otimizado é obtido.

A. Impacto do Parâmetro de Proporcionalidade sobre o Erro Quadrático a Posteriori do Algoritmo IPNLMS

Nesta seção é avaliado o impacto do parâmetro α na função objetivo adotada, a qual é baseada no erro *a posteriori* dado por [16]

$$e_p(n) = d(n) - \mathbf{x}^T(n) \mathbf{w}(n+1) + z(n). \quad (12)$$

Note que (12) é obtida substituindo, em (2), a estimativa $\mathbf{w}(n)$ pela estimativa atualizada dos pesos do filtro $\mathbf{w}(n+1)$, a qual é obtida a partir de (1), (2) e (3). Então, considerando (12), a função objetivo adotada é dada por

$$\psi(n) = 10 \log_{10} \left\{ [e_p(n)]^2 \right\} \quad (13)$$

onde (13) representa o erro quadrático *a posteriori* em dB. Assim, em cada iteração do processo de adaptação do algoritmo IPNLMS, o problema de otimização a ser solucionado pelo método da razão áurea é

$$\alpha_{\text{opt}}(n) = \arg \min_{-1 \leq \alpha(n) < 1} \psi(n) \quad (14)$$

sendo $\alpha_{\text{opt}}(n)$ o parâmetro de proporcionalidade ótimo na n -ésima iteração do processo de adaptação. Note, de (1) e (11), que α influencia diretamente no valor de $\mathbf{w}(n+1)$ e este, por sua vez, impacta no valor de (13).

Para verificar o efeito de α no comportamento de (13), são realizadas simulações de Monte Carlo (média de 100 realizações independentes) para um problema de identificação de sistemas. Os valores das variáveis avaliadas são obtidos por

$$E\{V(n)\} = \frac{1}{R} \sum_{r=1}^R V_r(n) \quad (15)$$

onde $E\{\cdot\}$ é o operador valor esperado, $V_r(n)$ é o valor da variável avaliada V na n -ésima iteração da r -ésima realização e R é o número de realizações consideradas. O cenário de todas as simulações consiste de uma planta esparsa com $N = 100$ pesos, apresentada em [4]. Nesta planta, os pesos ativos estão localizados nas posições $\{1, 30, 35, 85\}$, com seus respectivos valores iguais a $\{0, 1, 1, 0, -0, 5, 0, 1\}$. A medida de esparsidade desta planta é igual a $S(\mathbf{p}) = 0,9435$, calculada a partir da definição [14], [18]

$$S(\mathbf{p}) = \frac{N}{\sqrt{N} - N} \left(1 - \frac{\|\mathbf{p}\|_1}{\sqrt{N} \|\mathbf{p}\|_2} \right) \quad (16)$$

sendo $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_N]^T$ o vetor da resposta ao impulso da planta. Como entrada é utilizado um sinal correlacionado auto regressivo de ordem 2, AR(2), de variância unitária dado por

$$x(n) = 0,4x(n-1) - 0,4x(n-2) + v(n) \quad (17)$$

onde $v(n)$ é um ruído branco gaussiano de variância $\sigma_v^2 = 0,77$. A dispersão da matriz de autocorrelação do vetor de entrada é $\chi = 10$ e o ruído de medição $z(n)$ é gaussiano branco com variância $\sigma_z^2 = 10^{-3}$ (SNR = 30dB).

Com o objetivo de ilustrar o impacto de α em (13), a Fig. 2 ilustra o comportamento do erro quadrático *a posteriori* em dB na iteração 25 do processo de adaptação do algoritmo IPNLMS para valores de α na faixa $[-1, 0,99]$.

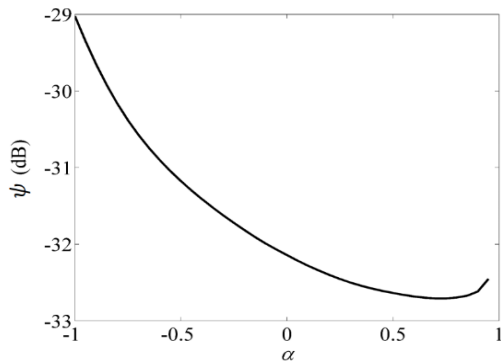


Fig. 2. Erro quadrático *a posteriori* em dB para valores de α na faixa $[-1, 0,99]$, na iteração 25 do processo de adaptação do IPNLMS para identificação da planta esparsa apresentada em [4].

Note da Fig. 2 que a escolha de α influencia diretamente no valor de (13). Então, a velocidade de convergência do algoritmo IPNLMS pode ser melhorada com a aplicação de um método para determinar $\alpha_{\text{opt}}(n)$ tal que solucione o problema de otimização dado por (14). A seguir é apresentado um método de otimização simples e eficaz, o método da razão áurea, o qual será utilizado na resolução de tal problema.

B. Método da Razão Áurea

A relação de seção áurea, dada pela constante $\tau = (\sqrt{5} + 1)/2 \cong 0,618$, foi considerada pelos gregos o valor mais esteticamente correto para a razão entre dois lados adjacentes de um mesmo retângulo [19]. O método da razão áurea, o qual é baseado nesta constante, é uma técnica de otimização linear livre de derivadas. O objetivo de tal método é minimizar uma função objetivo em torno de um dado intervalo de incerteza [15]. Este intervalo consiste de um conjunto de soluções factíveis, onde é sabido que o mesmo contém o minimizador da função. Apesar da fácil implementação e rápida convergência, para que a razão áurea tenha desempenho satisfatório é necessário que a função objetivo seja unimodal dentro do intervalo de incerteza [19]. A seguir é apresentado um algoritmo IPNLMS que utiliza o método da razão áurea para determinar o parâmetro de proporcionalidade ótimo em cada iteração do processo de adaptação.

C. Algoritmo IPNLMS proposto

O algoritmo IPNLMS com α otimizado é descrito a seguir em detalhes:

1. Inicialize o vetor de pesos do filtro ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}.$$

2. Obtenha os sinais desejado e de saída

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

3. Calcule o erro *a priori*

$$e(n) = d(n) - y(n) + z(n).$$

4. Atualize os pesos do filtro adaptativo

Nesta etapa, obtenha

$$\alpha_{\text{opt}}(n) = \arg \min_{-1 \leq \alpha(n) < 1} \psi(n)$$

onde

$$\psi(n) = 10 \log_{10} \{ [d(n) - \mathbf{x}^T(n)\mathbf{w}(n+1) + z(n)]^2 \}.$$

Para tal utilize o seguinte procedimento baseado na razão áurea:

- (P1) Faça $k = 1$, $a = -1$, $b = +1$, defina a tolerância *tol* para convergência da razão áurea e calcule

$$\alpha_1^k(n) = a(1 - \tau) + b\tau$$

$$\alpha_2^k(n) = a\tau + b(1 - \tau)$$

- (P2) A partir de (1), (2), (3) e (11), obtenha $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$ usando $\alpha_1^k(n)$ e $\alpha_2^k(n)$, respectivamente.

- (P3) A partir de (13), obtenha $\psi[\alpha_1^k(n)]$ e $\psi[\alpha_2^k(n)]$ usando $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$, respectivamente.

Se

$$|\psi[\alpha_1^k(n)] - \psi[\alpha_2^k(n)]| < \text{tol}$$

Vá para o passo (P5). Caso contrário, vá para (P4).

- (P4) Se

$$\psi[\alpha_1^k(n)] > \psi[\alpha_2^k(n)]$$

faça

$$k = k + 1$$

$$a = \alpha_1^{k-1}(n)$$

$$\alpha_1^k(n) = \alpha_2^{k-1}(n)$$

$$\alpha_2^k(n) = a\tau + b(1 - \tau)$$

e retorne para (P2).

Caso contrário, faça

$$k = k + 1$$

$$b = \alpha_2^{k-1}(n)$$

$$\alpha_2^k(n) = \alpha_1^{k-1}(n)$$

$$\alpha_1^k(n) = b\tau + a(1 - \tau)$$

e retorne para (P2).

- (P5) Razão áurea convergiu. Faça $\alpha_{\text{opt}}(n) = \alpha_1^k(n)$ [ou $\alpha_{\text{opt}}(n) = \alpha_2^k(n)$]. Em seguida, calcule $\mathbf{w}(n+1)$ usando

$$g_i(n) = [1 - \alpha_{\text{opt}}(n)] \frac{1}{2N} + [1 + \alpha_{\text{opt}}(n)] \frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varepsilon}.$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os itens 2, 3 e 4 até o fim do processo de adaptação.

Sendo $\alpha_1^k(n)$ e $\alpha_2^k(n)$ candidatos a $\alpha_{\text{opt}}(n)$ do ciclo atual k da razão áurea.

D. Complexidade Computacional

Na Tabela I, são avaliadas as complexidades computacionais dos algoritmos NLMS, IPNLMS e proposto em termos do número total de adições (A), multiplicações (M), divisões (D) e comparações (C). Aqui, para propósito de avaliação da complexidade computacional do algoritmo proposto, considera-se que K é número de ciclos necessário para o método da razão áurea convergir na n -ésima iteração do processo de adaptação.

TABELA I. COMPARAÇÃO DE COMPLEXIDADE COMPUTACIONAL

Algoritmo	A	M	D	C
NLMS	$3N$	$3N + 1$	1	0
IPNLMS	$5N + 2$	$6N + 2$	$N + 1$	0
Proposto	$(5K + 4)N + 4K + 2$	$(6K + 5)N + 5K + 2$	$(K + 2)N + 3$	$2K$

IV. SIMULAÇÃO E RESULTADOS

Para avaliar o algoritmo IPNLMS proposto, referido deste ponto em diante como GS-IPNLMS (*golden section IPNLMS*), são realizadas simulações de Monte Carlo (média de 100 realizações independentes) para um problema de identificação de sistemas. A figura de mérito utilizada para avaliar o desempenho dos algoritmos considerados é o desalinhamento normalizado em dB dado por

$$\eta(n) = 10 \log_{10} \left(\frac{\|\mathbf{p} - \mathbf{w}(n)\|_2^2}{\|\mathbf{p}\|_2^2} \right). \quad (18)$$

Três exemplos comparativos são considerados a seguir. Os dois primeiros consideram a planta esparsa descrita na seção III. Já o terceiro exemplo utiliza uma planta de esparsidade variável descrita por um processo markoviano de primeira ordem, semelhante ao apresentado em [20].

A. Exemplo 1

Este exemplo tem como objetivo comparar o desempenho dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS em termos de velocidade de convergência e resposta a uma perturbação ocorrida no instante $n = 2000$, quando o vetor de pesos da resposta ao impulso da planta muda de \mathbf{p} para $-\mathbf{p}$. Esta inversão não altera a esparsidade da planta.

A Fig. 3 mostra o desalinhamento normalizado em dB dos algoritmos considerados e o comportamento do α ótimo para o GS-IPNLMS. O parâmetro de passo adotado neste exemplo para os algoritmos IPNLMS e GS-IPNLMS é $\mu = 0,5$; já para o SC-IPNLMS, considera-se $\mu = 0,9$, de forma que todos os algoritmos considerados apresentem o mesmo desalinhamento em regime permanente. Para os algoritmos IPNLMS e SC-IPNLMS, adota-se $\alpha = 0,0$, de acordo com [9] e [11]. Para a convergência da razão áurea, considera-se $tol = 10^{-3}$. Note das curvas da Fig. 3 que o GS-IPNLMS alcança a mais rápida velocidade de convergência, mesmo após a ocorrência da perturbação na planta. Neste caso, α ótimo inicia em $-0,230$ na primeira iteração, muda para $0,984$ logo na segunda iteração e, em menos de 100 iterações, estabiliza-se em torno de $0,965$, não sendo afetado pela inversão da resposta ao impulso da planta.

B. Exemplo 2

Neste exemplo, os algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS são novamente comparados em termos de velocidade de convergência e resposta a uma perturbação que não altera a esparsidade da planta. Desta vez, o vetor \mathbf{p} da resposta ao impulso da planta é deslocado de 12 amostras para a direita no instante $n = 2000$. Assim, os pesos ativos de \mathbf{p} , cujos valores são iguais a $\{0,1, 1,0, -0,5, 0,1\}$, são movidos das posições $\{1, 30, 35, 85\}$ para as posições $\{13, 42, 47, 97\}$, respectivamente.

A Fig. 4 mostra o desalinhamento normalizado em dB dos algoritmos considerados e o comportamento de α ótimo para o GS-IPNLMS. O parâmetro de passo adotado para os algoritmos IPNLMS e GS-IPNLMS é $\mu = 0,5$. Já para o SC-IPNLMS, considera-se $\mu = 0,9$, de forma que todos os algoritmos apresentem o mesmo desalinhamento em regime permanente. Para os algoritmos IPNLMS e SC-IPNLMS adota-se $\alpha = 0,0$, de acordo com [9] e [11]. Para a convergência da razão áurea, considera-se $tol = 10^{-3}$. Note das curvas da Fig. 4 que o GS-IPNLMS alcança novamente a mais rápida velocidade de convergência mesmo após ocorrer a perturbação. Neste caso, α ótimo inicia em $-0,225$, muda para $0,986$ na iteração seguinte e, em menos de 100 iterações, estabiliza-se em torno de $0,968$.

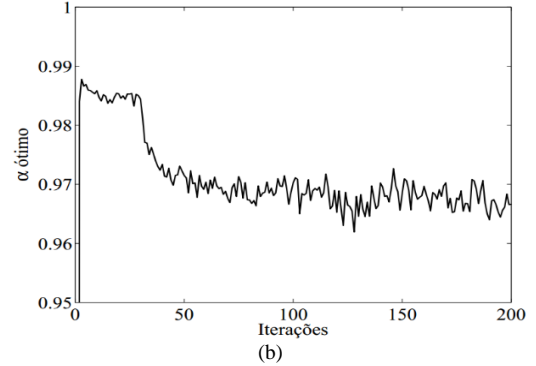
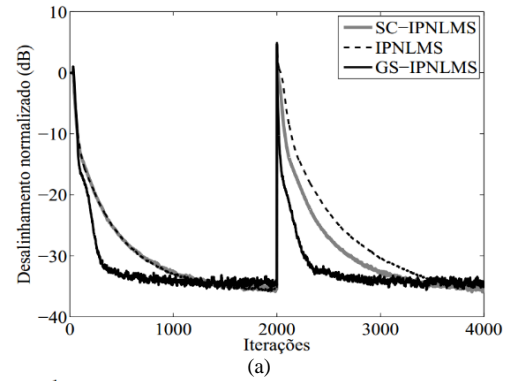


Fig. 3. Comportamento dos algoritmos IPNLMS ($\alpha = 0,0$ e $\mu = 0,5$), SC-IPNLMS ($\alpha = 0,0$ e $\mu = 0,9$) e GS-IPNLMS ($tol = 10^{-3}$ e $\mu = 0,5$), considerando uma inversão da planta em $n = 2000$. (a) Desalinhamento normalizado em dB. (b) Parâmetro de proporcionalidade ótimo do GS-IPNLMS.

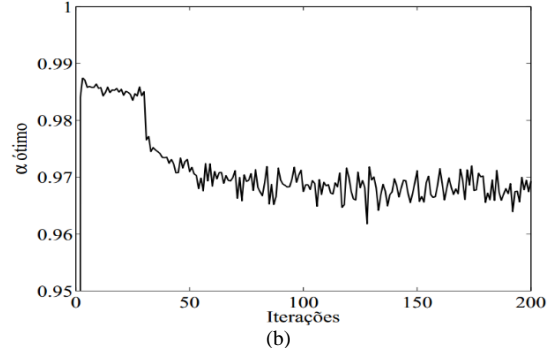
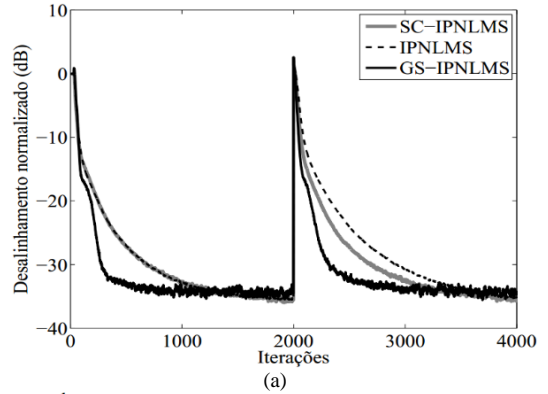


Fig. 4. Comportamento dos algoritmos IPNLMS ($\alpha = 0,0$ e $\mu = 0,5$), SC-IPNLMS ($\alpha = 0,0$ e $\mu = 0,9$) e GS-IPNLMS ($tol = 10^{-3}$ e $\mu = 0,5$), considerando em $n = 2000$ um deslocamento em \mathbf{p} de 12 amostras para a direita. (a) Desalinhamento normalizado em dB. (b) Parâmetro de proporcionalidade ótimo do GS-IPNLMS.

C. Exemplo 3

Neste exemplo a velocidade de convergência dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS é comparada, considerando a identificação de uma planta com esparsidade variável no tempo. Para tal, é utilizado o processo markoviano de primeira ordem dado por

$$\mathbf{p}(n+1) = \epsilon \mathbf{p}(n) + \sqrt{1 - \epsilon^2} \mathbf{s}(n) \quad (19)$$

onde $\mathbf{s}(n)$ é uma sequência aleatória de tamanho N , formada a partir de uma distribuição gaussiana de média zero e variância σ_s^2 . O fator $0 < \epsilon < 1$ controla as contribuições da “memória” e das “inovações” referentes à resposta ao impulso da planta $\mathbf{p}(n)$ [20]. Note que, para o $\epsilon = 1$, tem-se uma planta invariante no tempo. Neste exemplo, a planta variante é inicializada (instante $n = 0$) com os valores da planta esparsa apresentada na seção III, considerando $\sigma_s^2 = 1$ e $\epsilon = 1 - 10^{-7}$, assim como em [20].

A Fig. 5 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento de α ótimo do GS-IPNLMS e a variação de esparsidade da planta. Considera-se $\mu = 1,5$ para o IPNLMS, $\mu = 0,9$ para o SC-IPNLMS e $\mu = 0,5$ para o GS-IPNLMS, de forma que todos os algoritmos apresentem o mesmo desalinhamento em regime permanente. Para os algoritmos IPNLMS e SC-IPNLMS adota-se $\alpha = 0,0$, de acordo com [9] e [11]. Para a convergência da razão áurea considera-se $tol = 10^{-1}$. Note das curvas da Fig. 5 que o GS-IPNLMS alcança mais uma vez a maior velocidade de convergência, mesmo ocorrendo uma redução gradual na esparsidade da planta de 0,9435 em $n = 0$ para 0,8389 em $n = 1000$. Neste caso, o valor de α ótimo na primeira iteração é $-0,215$ e, logo na iteração seguinte, muda para 0,846, estabilizando em 0,520 em menos de 50 iterações.

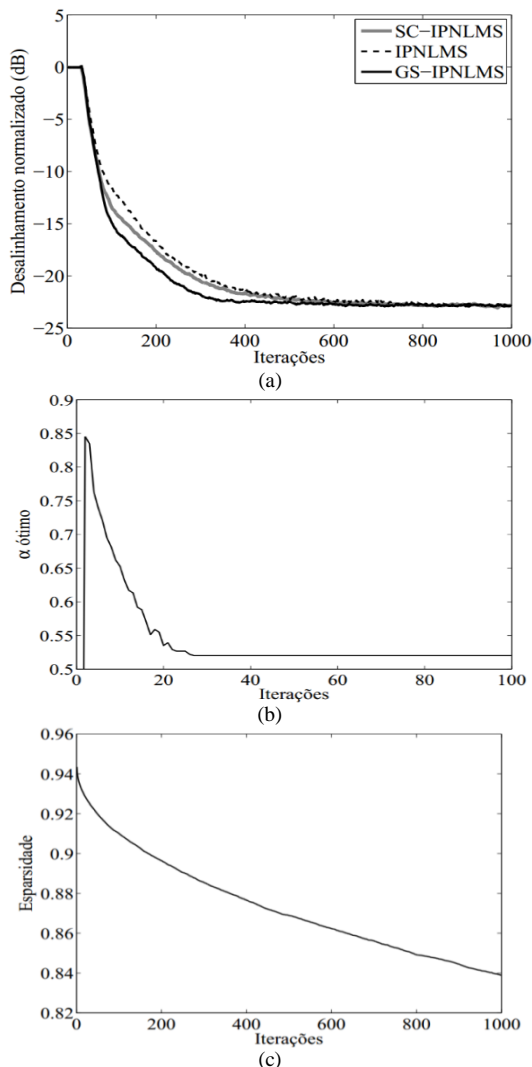


Fig. 5. Comportamento dos algoritmos IPNLMS ($\alpha = 0,0$ e $\mu = 1,5$), SC-IPNLMS ($\alpha = 0,0$ e $\mu = 0,9$) e GS-IPNLMS ($tol = 10^{-1}$ e $\mu = 0,5$) para a identificação de uma planta de esparsidade variável. (a) Desalinhamento normalizado em dB. (b) Parâmetro de proporcionalidade ótimo do GS-IPNLMS. (c) Variação da esparsidade da planta.

V. CONSIDERAÇÕES FINAIS

Neste trabalho propõe-se uma metodologia para otimizar a escolha do parâmetro de proporcionalidade do algoritmo IPNLMS. Para tal, utiliza-se um método de otimização linear livre de derivadas chamado razão áurea. Os resultados de simulação obtidos demonstram que a abordagem proposta apresenta desempenho superior ao dos algoritmos IPNLMS original e SC-IPNLMS na identificação de plantas esparsas, mesmo face à ocorrência de perturbações. Os resultados obtidos também revelam que o algoritmo GS-IPNLMS proposto supera o IPNLMS e o SC-IPNLMS na identificação de plantas de esparsidade variável.

REFERÊNCIAS

- [1] V. H. Nascimento and M. T. M. Silva, “Chapter 12 – Adaptive Filters,” in *Proc. Academic Press Library in Signal Processing*, Vol. 1, pp. 619-761, February 2014.
- [2] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*. Wiley & Sons, 2nd ed., 2013.
- [3] B. Jelfs, D. P. Mandic, and J. Benesty, “A unifying approach to the derivation of the class of PNLS algorithms,” in *Proc. 15th Int. Conf. Digital Signal Process.*, Cardiff, UK, pp. 19-22, Jul. 2007.
- [4] F. C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, “A PNLS Algorithm with Individual Activation Factors,” in *Proc. IEEE Trans. on Signal Processing*, Vol. 58, No. 4, April 2010.
- [5] J. Jin, Q. Qu and Y. Gu, “Robust zero-point attraction least mean square algorithm on near sparse system identification,” in *Proc. IET Signal Processing*, Vol. 7, Iss. 3 pp. 210–218.
- [6] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice Hall, 4th ed., 2002.
- [7] A. Abdollahi, P. Zhang, H. Xue, and S. Li, “Enhanced Subspace-Least Mean Square for Fast and Accurate Power System Measurement” in *Proc. IEEE Trans. on Power Delivery*, vol. 28, No. 1, pp. 383-393, January 2013.
- [8] D. Duttweiler, “Proportionate normalized least-meansquares adaptation in echo cancelers,” *IEEE Trans. Speech, Audio Proc.*, vol. 8, no. 5, pp. 508–518, Sep. 2000.
- [9] J. Benesty and S. L. Gay, “An improved PNLS algorithm,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal (ICASSP)*, vol. 2, pp. 1881–1884, 2002.
- [10] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in *Conf. Record of the XXXII Asilomar Conf. on Signals, Systems and Computers*, Monterey, USA, vol. 1, pp. 394–398, Nov. 1998.
- [11] P. Loganathan, A. W. H. Khong, and P. A. Naylor, “A Class of Sparseness-Controlled Algorithms for Echo Cancellation,” in *Proc. IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, no. 8, Nov. 2009.
- [12] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, New Jersey: Prentice Hall, 1985.
- [13] Y. Li, Y. Gu, and K. Tang, “Parallel NLMS Filters with Stochastic Active Taps and Step-Sizes for Sparse System Identification” In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, Toulouse, France, May 2006, Vol. 3, pp. 109-112.
- [14] Y. Huang, J. Benesty and J. Chen, *Acoustic MIMO Signal Processing*, Springer, 2006.
- [15] M. S. Bazarraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 3rd ed., 2006.
- [16] H. A. Sayed, *Fundamentals of Adaptive Filtering*. Wiley & Sons, 2003.
- [17] F. C. de Souza. Algoritmos Adaptativos LMS Normalizados Proporcionalis: Proposta de um Novo Algoritmo e sua Modelagem Estocástica. Tese: Doutorado. Universidade Federal de Santa Catarina (UFSC), 2012.
- [18] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, Nov. 2004.
- [19] D. G. Luenberger, and Y. Ye, *Linear and Nonlinear Programming*. Springer, 3rd ed., 2008.
- [20] P. Loganathan, E. A. P. Habets, and P. A. Naylor, “Performance Analysis of IPNLMS for Identification of Time-Varying Systems”, In *Proc. IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, Texas, USA, March 2010, pp. 317-320.