

# Algoritmo ZA-PNLMS com Fatores de Ativação Individuais e Ganhos dos Coeficientes Limitados Superiormente\*

Alzeneide Dutra Pessoa\*, Francisco das Chagas de Souza\*

\*Laboratório de Sistemas Adaptativos e Processamento de Sinais – LSAPS  
Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Maranhão – UFMA  
Av. dos Portugueses, 1966, CEP 65080-805, Bacanga, São Luís, MA, Brasil.  
E-mails: alzeneide.pessoa@discente.ufma.br, francisco.souza@ufma.br

---

**Abstract:** In this paper, a zero-attracted proportionate normalized LMS adaptive algorithm (ZA-PNLMS) is proposed, which combines individual activation factors with upper limited gains. The proposal presented here upper limits the gains of the algorithm coefficients for identifying plants with a high degree of sparseness, leading to better distribution of the adaptation energy between the coefficients. Computer simulations, considering sparse systems with disturbance and tracking, attest to the effectiveness of the proposed algorithm.

**Resumo:** Neste artigo, propõe-se um algoritmo adaptativo LMS normalizado proporcional com atrator para zero (ZA-PNLMS – *zero-attracting proportionate normalized LMS algorithm*), o qual combina fatores de ativação individuais com ganhos limitados superiormente. A proposta aqui apresentada limita superiormente os ganhos dos coeficientes do algoritmo para identificação de plantas com alto grau de esparsidade, levando a uma melhor distribuição da energia de adaptação entre os coeficientes do algoritmo. Simulações computacionais, considerando sistemas esparsos com perturbação e rastreamento, atestam a eficácia do algoritmo proposto.

**Keywords:** Activation factors; adaptive algorithm; sparse plants; upper limited gains; zero-attracted.

**Palavras-chaves:** Fatores de ativação; algoritmo adaptativo; plantas esparsas; ganhos limitados superiormente; atrator para zero.

---

## 1. INTRODUÇÃO

Entre as técnicas de processamento digital de sinais, os filtros adaptativos têm se destacado, principalmente porque apresentam bom desempenho na solução de diversos problemas. O termo “filtro” pode ser usado para descrever um dispositivo físico ou computacional que colete uma série de dados aleatórios e processa-os segundo um conjunto de regras pré-definidas (Haykin, 2014). Quando os parâmetros de um filtro são capazes de alterar seus respectivos valores com o tempo, diz-se que o filtro é “adaptativo”. Entre os problemas nos quais se utiliza filtros adaptativos está a identificação de sistemas (Müller et al., 2006). O desafio de identificar um sistema torna-se ainda mais desafiador quando este possui alto grau de esparsidade, ou seja, poucos coeficientes diferentes de zero (Widrow et al., 1985). Para sistemas com alto grau de esparsidade, os algoritmos adaptativos clássicos como o LMS (*least-mean-square*) e LMS normalizado (NLMS – *normalized least-mean-square*) apresentam desempenho pobre em termo de velocidade de

convergência. Por outro lado, o algoritmo LMS normalizado proporcional (PNLMS – *proportionate normalized least-mean-square*) tem se mostrado superior quando a tarefa é a identificação de plantas esparsas (Gay, 1998; Gay, 2002; Huang, 2006; Paleologu et al., 2010).

O algoritmo PNLMS é conhecido por produzir uma taxa de convergência inicial mais rápida, fornecendo um maior ganho para coeficientes ativos em comparação com os inativos (Duttweiler, 2000). Embora o PNLMS apresente uma convergência inicial mais rápida, tem sido observado que sua taxa de convergência reduz-se conforme avançam as iterações, e a sua taxa de convergência pode até ser menor do que a do NLMS na fase posterior do processo de adaptação (Das, 2020).

A fim de melhorar o desempenho do PNLMS, vários algoritmos adaptativos foram propostos, surgindo então uma classe de algoritmos PNLMS, tais como o PNLMS aprimorado (IPNLMS – *improved proportionate normalized least-mean-square*) (Benesty et al., 2002) e o PNLMS com fatores de ativação individuais (IAF-PNLMS – *individual-activation-factor proportionate normalized least-mean-square*) (Souza et al., 2010). No PNLMS, o fator de ativação é o mesmo para todos os coeficientes e depende da norma infinita instantânea do vetor coeficientes do filtro adaptativo.

---

\*O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, a Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Já no IAF-PNLMS, é utilizados um fator de ativação para cada coeficiente do filtro (Souza et al., 2010).

Das (2016) propõe outro algoritmo da classe PNLMS denominado PNLMS com atração para zero (ZA-PNLMS – *zero-attracting proportionate normalized least-mean-square*). A proposta do ZA-PNLMS é que uma norma  $l_1$  do vetor de coeficientes do filtro seja introduzida como um termo de penalidade na função custo do algoritmo. Das (2020), além da introdução de um termo de penalidade, propõe que a matriz de ganhos utilizada na adaptação dos coeficientes seja calculada considerando uma função de limite superior adaptativa, concedendo assim o algoritmo denominado ZA-PNLMS com limite superior (UT-ZA-PNLMS – *upper threshold based ZA-PNLMS*).

Diante do exposto, verifica-se que o UT-ZA-PNLMS melhora o desempenho do algoritmo ZA-PNLMS através de um limiar superior para os ganhos dos coeficientes, enquanto que o IAF-PNLMS melhora o PNLMS baseado em fatores de ativação individuais. A proposta deste trabalho é combinar em um único algoritmo as filosofias dos algoritmos UT-ZA-PNLMS e IAF-PNLMS, ou seja, utilizar um limite superior adaptativo para cálculo dos ganhos do filtro, introduzindo o uso dos fatores de ativação determinados de maneira individual para cada coeficiente.

Este artigo está organizado como segue. A Seção II revisita os algoritmos proporcionais. A Seção III apresenta uma visão geral dos algoritmos da família *zero-attracting*. Na Seção IV, um novo algoritmo é proposto, contemplando as características de atrator para zero, proporcionais e de ganhos limitados superiormente. Na Seção V, são mostrados os resultados de simulação atestando a eficácia da proposta. Finalmente, na Seção VI, são apresentadas as conclusões deste trabalho de pesquisa.

## 2. ALGORITMOS PROPORCIONAIS

Nessa seção, são apresentados os principais algoritmos adaptativos proporcionais. Para maior clareza, ilustra-se na Figura 1 o diagrama de blocos de um esquema de filtragem adaptativa, onde  $\mathbf{x}(n) = [x_1(n) \ x_2(n) \ \dots \ x_N(n)]^T$  é o vetor de entrada do filtro,  $\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \dots \ w_N(n)]^T$  é o vetor de coeficientes do filtro adaptativo,  $d(n)$  é o sinal desejado e  $v(n)$ , um ruído de medição. A saída do filtro adaptativo é dada por

$$y(n) = \mathbf{x}^T(n)\mathbf{w}(n) \quad (1)$$

e o sinal de erro instantâneo por

$$e(n) = d(n) - y(n) + v(n). \quad (2)$$

O algoritmo adaptativo mais básico é o LMS (Haykin, 1996). Entretanto, apesar de simples e eficaz, o mesmo possui uma convergência lenta quando a resposta ao impulso é de natureza esparsa (Duttweiler, 2000). A fim de superar essa desvantagem, novos algoritmos foram sendo criados, como o algoritmo PNLMS. Neste algoritmo, a regra de adaptação dos coeficientes  $w(n)$  é dada por (Branco, 2016):

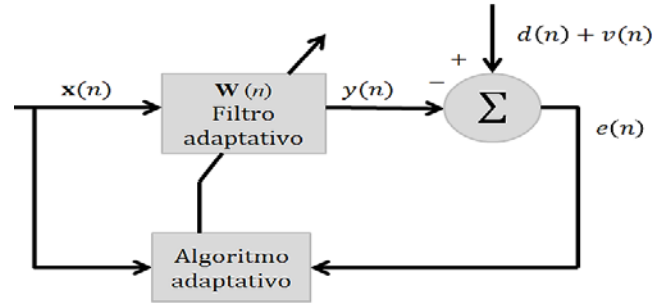


Figura 1. Esquema de filtragem adaptativa.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n)e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n) + \varepsilon}, \quad (3)$$

onde  $\mu$  e o passo de adaptação e  $\varepsilon > 0$  é um parâmetro de regularização numérica para estabilizar a solução. A matriz

$$\mathbf{G}(n) = \text{diag} [g_1(n) \ g_2(n) \ \dots \ g_N(n)], \quad (4)$$

de ordem  $N \times N$  é responsável pela distribuição de ganhos individuais  $g_i(n)$ , controlando o ajuste do  $i$ -ésimo coeficiente do algoritmo. O operador  $\text{diag}$  define uma matriz diagonal cujo os elementos  $g_1(n) \ g_2(n) \ \dots \ g_N(n)$  compõe a diagonal principal.

### 2.1 Algoritmo PNLMS

As Equações (3) e (4) são a base para os algoritmos da família proporcional, sendo que as diferentes versões de algoritmos desta família são obtidas modificando-se a maneira na qual o ganho individual  $g_i(n)$  é calculado. No caso do algoritmo PNLMS, o ganho  $g_i(n)$  associado ao  $i$ -ésimo coeficiente  $w_i(n)$  é obtido a partir de

$$g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}, \quad i = 1, 2, \dots, N, \quad (5)$$

em que a função de proporcionalidade  $\phi_i(n)$  é definida como

$$\phi_i(n) = \max[f(n), |w_i(n)|], \quad (6)$$

sendo

$$f(n) = \rho \max[\xi, \|\mathbf{w}(n)\|_\infty] \quad (7)$$

o fator de ativação, o qual influencia diretamente na adaptação dos coeficientes considerados inativos. Tal fator depende da norma infinita do vetor de coeficientes do filtro,  $\|\mathbf{w}(n)\|_\infty$ , e dos parâmetros de inicialização,  $\xi$ , e de proporcionalidade,  $\rho$  (Souza et al., 2009).

### 2.2 Algoritmo IAF-PNLMS

O algoritmo IAF-PNLMS (Souza et al. 2010) é uma variante do PNLMS em que o fator de ativação é uma função individual, a qual é definida para cada coeficiente do filtro adaptativo, i.e.,

$$f_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN, \ m = 1, 2, \dots \\ f_i(n-1) & \text{caso contrário} \end{cases} \quad (8)$$

De acordo com Souza et al. (2009), o algoritmo IAF-PNLMS fornece uma melhor distribuição de ganho de adaptação do que o PNLMS, levando a um aumento na velocidade de convergência do algoritmo em aplicações com plantas de alta esparsidade.

Ressalta-se, a partir de (8), que no algoritmo IAF-PNLMS, cada fator de ativação é calculado em termos da magnitude de seu correspondente coeficiente, e não depende dos parâmetros de proporcionalidade e inicialização presentes no algoritmo PNLMS.

### 3. ALGORITMOS DA FAMÍLIA ZERO-ATTRACTING

Nessa seção, serão apresentados os principais algoritmos que incorporam o princípio do atrator para zero, ou seja, os algoritmos da família ZA (*zero-attracting*). Tais algoritmos aceleram a convergência quando aplicados na identificação de plantas esparsas (Chen, 2009).

#### 3.1 Algoritmo ZA-LMS

Para melhor compreensão do algoritmo LMS com atrator para zero (ZA-LMS – *zero-attracting LMS*), os conceitos relativos ao LMS são abordados nessa seção de forma introdutória e breve.

O algoritmo LMS consiste em uma técnica de atualização do vetor de coeficientes de um filtro adaptativo utilizando, em sua regra de adaptação, uma estimativa do erro quadrático médio tendo como base o erro instantâneo (Widrow et al., 1985). Assim, considerando o erro quadrático instantâneo  $e^2(n)$  como a função custo utilizada para obtenção da regra de atualização dos coeficientes  $\mathbf{w}(n)$  do filtro, obtém-se pelo método do gradiente descendente que

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} e^2(n) \quad (9)$$

onde  $\mu$  é o parâmetro de passo e  $\nabla_{\mathbf{w}(n)}$  é o operador gradiente em relação a  $\mathbf{w}(n)$  definido como

$$\nabla_{\mathbf{w}(n)} = \left[ \frac{\partial}{\partial w_1(n)} \quad \frac{\partial}{\partial w_2(n)} \quad \dots \quad \frac{\partial}{\partial w_N(n)} \right]^T. \quad (10)$$

Dessa forma, substituindo-se (1) em (2) e usando (10), tem-se

$$\nabla_{\mathbf{w}(n)} e^2(n) = 2e(n) \frac{\partial [d(n) - \mathbf{x}^T(n)\mathbf{w}(n) + v(n)]}{\partial \mathbf{w}(n)} \quad (11)$$

e, conseqüentemente, o gradiente da função custo do LMS resulta em

$$\nabla_{\mathbf{w}(n)} e^2(n) = -2e(n)\mathbf{x}(n). \quad (12)$$

Agora, substituindo (12) em (9), resulta

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n) \quad (13)$$

que define a regra de atualização dos coeficientes  $\mathbf{w}(n)$  do algoritmo LMS.

A fim de apresentar uma alternativa para identificar sistemas esparsos, Chen et al. (2009) introduzem, na função custo do LMS, uma penalidade que favorece a esparsidade da resposta ao impulso, incorporando assim um termo de norma  $l_1$  na função custo do algoritmo. Essa modificação resulta em uma versão modificada do LMS, ou seja, um atrator para zero para todos os coeficientes. A esse algoritmo, Chen et al. (2009) denominam de ZA-LMS. Com essa modificação, a função custo do ZA-LMS fica na forma

$$L_1(n) = \frac{1}{2} e^2(n) + \gamma \|\mathbf{w}(n)\|_1. \quad (14)$$

Agora, calculando o gradiente da função custo (14), tem-se

$$\nabla_{\mathbf{w}(n)} L_1(n) = \frac{1}{2} \nabla_{\mathbf{w}(n)} e^2(n) + \gamma \nabla_{\mathbf{w}(n)} \|\mathbf{w}(n)\|_1. \quad (15)$$

Sabendo que

$$\nabla_{\mathbf{w}(n)} \|\mathbf{w}(n)\|_1 = -\text{sgn}[\mathbf{w}(n)] \quad (16)$$

e substituindo (12) e (16) em (15) obtém-se

$$\nabla_{\mathbf{w}(n)} L_1(n) = -e(n)\mathbf{x}(n) - \gamma \text{sgn}[\mathbf{w}(n)], \quad (17)$$

em que a função  $\text{sgn}(w_i)$  é definida como

$$\text{sgn}(w_i) = \begin{cases} \frac{w_i}{|w_i|} & w_i \neq 0 \\ 0 & w_i = 0. \end{cases} \quad (18)$$

Logo, pelo método do Gradiente descendente, escreve-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} L_1(n) \quad (19)$$

e substituindo (15) em (19), obtém-se a expressão de atualização dos coeficientes do algoritmo ZA-LMS como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) + \rho \text{sgn}[\mathbf{w}(n)], \quad (20)$$

em que  $\rho = \mu\gamma$ .

#### 3.2 Algoritmo RZA-LMS

No algoritmo ZA-LMS, todos os coeficientes são uniformemente atraídos a zero, e isso leva a uma redução de desempenho em aplicações com plantas dispersivas (menor grau de esparsidade). Chen et al. (2009), propõem uma abordagem chamada LMS com atrator para zero reponderado (RZA-LMS – *reweighted zero-attracting LMS*).

O RZA-LMS origina-se então de uma nova função custo dada por

$$L_2(n) = \frac{1}{2} e^2(n) + \gamma \sum_{i=0}^N \log(1 + \varepsilon |w_i(n)|), \quad (21)$$

com  $\varepsilon > 0$ . Calculando o gradiente  $\nabla_{\mathbf{w}(n)}$  de  $L_2(n)$ , tem-se

$$\nabla_{\mathbf{w}(n)} L_2(n) = \frac{1}{2} \nabla_{\mathbf{w}(n)} e^2(n) + \gamma \nabla_{\mathbf{w}(n)} \sum_{i=0}^N \log(1 + \varepsilon |w_i(n)|). \quad (22)$$

A primeira parcela do lado direito de (22) é dada por (11), restando agora determinar a segunda parcela que, para fins didáticos, é denominada

$$\mathbf{S}_w = \nabla_{\mathbf{w}(n)} \sum_{i=0}^N \log(1 + \varepsilon |w_i(n)|). \quad (23)$$

Abrindo o somatório em (23), resulta

$$\mathbf{S}_w = \nabla_{\mathbf{w}(n)} [\log(1 + \varepsilon |w_1(n)|) + \log(1 + \varepsilon |w_2(n)|) + \dots + \log(1 + \varepsilon |w_N(n)|)]. \quad (24)$$

Assim, utilizando a definição de  $\nabla_{\mathbf{w}(n)}$ , obtém-se

$$\mathbf{S}_w = \left[ \frac{\varepsilon \text{sgn}\{w_1(n)\}}{1 + \varepsilon |w_1(n)|} \quad \frac{\varepsilon \text{sgn}\{w_2(n)\}}{1 + \varepsilon |w_2(n)|} \quad \dots \quad \frac{\varepsilon \text{sgn}\{w_N(n)\}}{1 + \varepsilon |w_N(n)|} \right]^T \quad (25)$$

e (12).

Desse modo, (22) pode ser escrita como

$$\nabla_{\mathbf{w}(n)} L_2(n) = \gamma \mathbf{S}_w - e(n) \mathbf{x}(n). \quad (26)$$

Considerando o método do gradiente descendente, tem-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} L_2(n) \quad (27)$$

e, substituindo (26) em (27), chega-se a expressão de atualização dos coeficientes do RZA-LMS, que é dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \rho \mathbf{S}_w + \mu e(n) \mathbf{x}(n), \quad (28)$$

em que  $\rho = \mu\gamma$ .

### 3.3 Algoritmo ZA-PNLMS

A ideia de incorporar uma penalidade também foi utilizada no algoritmo PNLMS. Em Das et al. (2016), visando obter uma regra de atualização dos coeficientes do PNLMS com atrator para Zero, o seguinte problema de otimização com restrição é proposto:

$$\min \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_{\mathbf{G}^{-1}}^2 \gamma \|\mathbf{G}^{-1} \mathbf{w}(n+1)\|_1 \quad (29)$$

sujeito a restrição

$$d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n) = 0, \quad (30)$$

em que  $\gamma \ll 1$  e a notação  $\|\mathbf{v}\|_{\mathbf{G}^{-1}}^2$  indica o produto interno generalizado  $\mathbf{v}^T \mathbf{G}^{-1} \mathbf{v}$ .

Utilizando (29) e (30) e o multiplicador de Lagrange  $\lambda$ , tem-se a seguinte função custo a ser minimizada:

$$J(n+1) = \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_{\mathbf{G}^{-1}}^2 \gamma \|\mathbf{G}^{-1} \mathbf{w}(n+1)\|_1 + \lambda [d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n)]. \quad (31)$$

Para minimização de (31), toma-se

$$\frac{\partial J(n+1)}{\partial \mathbf{w}(n+1)} = 0 \quad (32)$$

e consequentemente

$$\mathbf{G}^{-1} [\mathbf{w}(n+1) - \mathbf{w}(n)] + \mathbf{G}^{-1} \gamma \operatorname{sgn}(\mathbf{w}(n+1)) - \lambda \mathbf{x}(n) = 0. \quad (33)$$

Pré-multiplicando (33) por  $\mathbf{G}$  e, em seguida, isolando  $\mathbf{w}(n+1)$ , resulta

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \gamma \operatorname{sgn}(\mathbf{w}(n+1)) + \lambda \mathbf{G}(n) \mathbf{x}(n). \quad (34)$$

Pré-multiplicando (34) por  $\mathbf{x}^T(n)$ , usando a restrição (30) e isolando  $\lambda$ , obtém-se

$$\lambda = \frac{e(n) + \gamma \mathbf{x}^T(n) \operatorname{sgn}(\mathbf{w}(n+1))}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)}. \quad (35)$$

Agora, substituindo (35) em (34), tem-se:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n) \mathbf{G}(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} - \gamma \left[ \mathbf{I} - \frac{\mathbf{G}(n) \mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} \right] \operatorname{sgn}(\mathbf{w}(n+1)), \quad (36)$$

em que

$$\mathbf{B} = \frac{\mathbf{G}(n) \mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)}. \quad (37)$$

Das et al. (2016) sugerem que para se obter a equação de atualização de coeficientes, deve-se aproximar  $\operatorname{sgn}(\mathbf{w}(n+1))$  por uma estimativa, a saber,  $\operatorname{sgn}(\mathbf{w}(n))$  que é conhecido. Essa aproximação baseia-se na hipótese de que a maioria dos coeficientes não sofre mudança de sinal à medida que são atualizados de uma iteração para a outra. Ressalta-se também o fato de que, em (34), qualquer elemento da matriz  $\mathbf{B}$  possui magnitude muito menor do que 1, especialmente para filtros de grande ordem e, assim, essa matriz pode ser desprezada em comparação com  $\mathbf{I}$ .

Por fim, introduzindo o tamanho do passo do algoritmo  $\mu$  e um parâmetro de regularização  $\delta_p$  em (36), obtém-se então a seguinte equação de atualização dos coeficientes do algoritmo ZA-PNLMS:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n) \mathbf{G}(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n) + \delta_p} - \rho \operatorname{sgn}(\mathbf{w}(n)), \quad (38)$$

em que  $\rho = \mu\gamma$ .

### 3.4 Algoritmo UT-ZA-PNLMS

O algoritmo ZA-PNLMS baseado em um limite superior (UT - upper threshold) para os ganhos (UT-ZA-PNLMS), proposto por Das (2020), propõe uma modificação na maneira como a matriz de ganho  $\mathbf{G}(n)$  é obtida, seja calculada com a introdução de uma função de limite superior adaptativa  $\tau_U(n)$  dada pela equação

$$\tau_U(n) = \frac{\sum_{i=0}^{N-1} |w_i(n)| |d_i(n)|}{\sum_{i=0}^{N-1} |d_i(n)|} + \beta_1 (1 - e^{\beta_2 \epsilon(n)}), \quad (39)$$

em que

$$\epsilon(n) = \lambda_e \epsilon(n-1) + (1 - \lambda_e) e^2(n). \quad (40)$$

e

$$d_i(n) = \lambda_d d_i(n-1) + (1 - \lambda_d) \frac{e(n) x(n-i)}{\mathbf{x}^T(n) \mathbf{x}(n) + \delta_N} \quad (41)$$

onde  $0 < \lambda_d < 1$ ,  $\delta_N$  é uma constante positiva para evitar divisão por zero,  $\beta_1$  e  $\beta_2$  são parâmetros de controle e  $\lambda_e$  é um fator de esquecimento no cálculo de  $\epsilon(n)$ .

Desse modo, a função de proporcionalidade passa a ser

$$\phi_i(n) = \min\{\tau_U(n), \max[f(n), |w_i(n)|]\}, \quad (42)$$

em que o fator de ativação  $f(n)$  é dado por

$$f(n) = \rho_g \max[\delta_g, |w_1(n)|, \dots, |w_N(n)|], \quad (43)$$

e os ganhos individuais

$$g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}, \quad i = 1, 2, \dots, N \quad (44)$$

e a matriz de ganhos

$$\mathbf{G}(n) = \operatorname{diag}[g_1(n), g_2(n) \dots g_N(n)]. \quad (45)$$

O algoritmo UT-ZA-PNLMS melhora a taxa de convergência oferecendo maiores ganhos para coeficientes ativos de menor magnitude à medida que as iterações progridem, e reduz as flutuações nos coeficientes ativos de maior magnitude (Das, 2020).

#### 4. ABORDAGEM PROPOSTA

A proposta desse artigo é implementar uma regra de atualização que leve os fatores de ativação individuais  $f_i(n)$  do algoritmo UT-ZA-PNLMS tenderem às respectivas magnitudes  $|w_i(n)|$ . Para tal, toma-se como base o procedimento adotado por Souza et al. (2009), o qual estabelece duas condições (C1 e C2) para obter fatores de ativação individuais em um algoritmo proporcional. Assim, considere as condições:

C1)  $f_i(n)$  deve convergir para a correspondente magnitude do coeficiente  $|w_i(n)|$ , ou seja,

$$[|w_i(n)| - f_i(n)] \rightarrow 0, \text{ quando } n \rightarrow \infty \quad (46)$$

C2)  $f_i(n)$  deve sempre ser maior do que zero.

Pode-se mostrar que a condição C1 é satisfeita se a seguinte função custo for minimizada:

$$L(n) = \frac{1}{2} [|\mathbf{w}(n)| - \mathbf{f}(n)]^T [|\mathbf{w}(n)| - \mathbf{f}(n)], \quad (47)$$

em que  $\mathbf{f}(n) = [f_1(n) \ f_2(n) \ \dots \ f_N(n)]^T$  é o vetor dos coeficientes de ativação individuais  $f_i(n)$ .

Consequentemente,

$$\nabla_{\mathbf{f}(n)} L(n) = -[|\mathbf{w}(n)| - \mathbf{f}(n)]. \quad (48)$$

Utilizando o método do gradiente descendente para encontrar  $\mathbf{f}(n)$  que minimiza  $L(n)$ , tem-se:

$$\mathbf{f}(n+1) = \mathbf{f}(n) - \varepsilon \nabla_{\mathbf{f}(n)} L(n). \quad (49)$$

Substituindo (48) em (49), resulta

$$\mathbf{f}(n+1) = \mathbf{f}(n) + \varepsilon [|\mathbf{w}(n)| - \mathbf{f}(n)]. \quad (50)$$

Portanto,

$$\mathbf{f}(n+1) = (1 - \varepsilon)\mathbf{f}(n) + \varepsilon |\mathbf{w}(n)|. \quad (51)$$

Note que a expressão obtida para atualização de  $\mathbf{f}(n)$  também satisfaz C2 desde que  $0 < \varepsilon < 1$  e  $f_i(n) > 0$ , para todo  $i = 1, 2, 3, \dots, N$  ( $N$  é comprimento do filtro adaptativo).

Sousa et al. (2010) destaca que pode ocorrer de  $|w_i(n)|$  não ser proporcional a  $|p_i(n)|$  no início do processo de adaptação. Para contornar essa limitação, faz-se a atualização periódica dos fatores de ativação individuais somente após períodos de aprendizagem de  $N$  amostras. Dessa forma, a expressão de atualização dada em (49) é reescrita como

$$\mathbf{f}(n+1) = \begin{cases} (1 - \varepsilon)\mathbf{f}(n) + \varepsilon |\mathbf{w}(n)| & n = mN, \ m = 1, 2, \dots \\ \mathbf{f}(n) & \text{caso contrário.} \end{cases} \quad (52)$$

Portanto, considerando (52), obtém-se um algoritmo que combina as principais características do IAF-PNLMS e UT-ZA-PNLMS, ou seja, a característica principal do IAF-PNLMS, que é a utilização de um fator de ativação para cada coeficiente do filtro adaptativo, é inserida no algoritmo UT-ZA-PNLMS. Além disso, mantém-se o cômputo da matriz de ganho  $\mathbf{G}(n)$ , utilizando uma função adaptativa,  $\tau_U(n)$ , que limita superiormente os ganhos individuais do algoritmo.

A proposta, que utiliza (52) no lugar de (43), combina as melhores características dos algoritmos IAF-PNLMS e UT-ZA-PNLMS, sendo então denominado algoritmo IAF-UT-ZA-PNLMS.

#### 5. RESULTADOS DE SIMULAÇÃO

As simulações computacionais são realizadas considerando três plantas diferentes. Em todos os cenários de simulação o objetivo principal é identificar uma planta esparsa com  $N = 100$  coeficientes.

O grau de esparsidade de cada planta é determinada pela seguinte definição (Gay, 1998; Huang, 2006):

$$S(\mathbf{p}) = \frac{N}{\sqrt{N-N}} \left( 1 - \frac{\|\mathbf{p}\|_1}{\sqrt{N}\|\mathbf{p}\|_2} \right) \quad (53)$$

onde  $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_N]^T$  é o vetor da resposta impulsiva da planta. Como entrada utiliza-se um sinal correlacionado autorregressivo de ordem 2 de variância unitária dado por

$$x(n) = 0,4x(n-1) - 0,4x(n-2) + v(n), \quad (54)$$

sendo  $v(n)$  um ruído branco gaussiano de variância  $\sigma_v^2 = 0,77$ . O ruído de medição,  $z(n)$ , é gaussiano branco com variância  $\sigma_z^2 = 10^{-3}$  (SNR = 30 dB).

São realizadas simulações de Monte Carlo (média de 100 realizações independentes). Os valores das variáveis avaliadas são obtidos por

$$E\{V(n)\} = \frac{1}{R} \sum_{r=1}^R V_r(n), \quad (55)$$

sendo  $E\{\cdot\}$  o operador esperança,  $V_r(n)$  é o valor da variável avaliada  $V(n)$  na  $n$ -ésima iteração da  $r$ -ésima realização e  $R$  é o número de realizações.

A figura de mérito utilizada para comparar o desempenho do algoritmo IAF-UT-ZA-PNLMS e comparar com o IAF-PNLMS e UT-ZA-PNLMS é o desalinhamento normalizado em dB dado por (Branco, 2016):

$$\eta(n) = 10 \log_{10} \left( \frac{\|\mathbf{p} - \mathbf{w}(n)\|_2^2}{\|\mathbf{p}\|_2^2} \right). \quad (56)$$

O tamanho do passo  $\mu$  é 0.5 para todos os algoritmos. Para os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS, são utilizados os seguintes parâmetros comuns aos dois algoritmos:  $\varepsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ . Os valores desses parâmetros definem o melhor desempenho em estado estacionário do algoritmo UT-ZA-PNLMS quando combinado com  $\delta_g = \rho_g = 0.01$  (Das, 2020).

Especificamente, para o algoritmo IAF-UT-ZA-PNLMS, o parâmetro  $\varepsilon$  foi definido como 0.01 e  $f_i(0) = 10^{-3}$  (para todo  $i$ ). Para o IAF-PNLMS, são considerados  $f_i(0) = 10^{-4}$  e como parâmetro de inicialização  $\xi = 0.001$ .

##### 5.1 Exemplo 1

O objetivo deste exemplo é realizar a estimação dos coeficientes da planta  $\mathbf{p}$  com coeficientes ativos  $\{1, 0.1\}$  nas posições  $\{30, 85\}$ . Após  $n = 2500$ , a planta  $\mathbf{p}$  sofre uma perturbação tornando-se  $-\mathbf{p}$ . O grau de esparsidade da resposta ao impulso para essa planta é  $S(\mathbf{p}) = 0.9895$ .

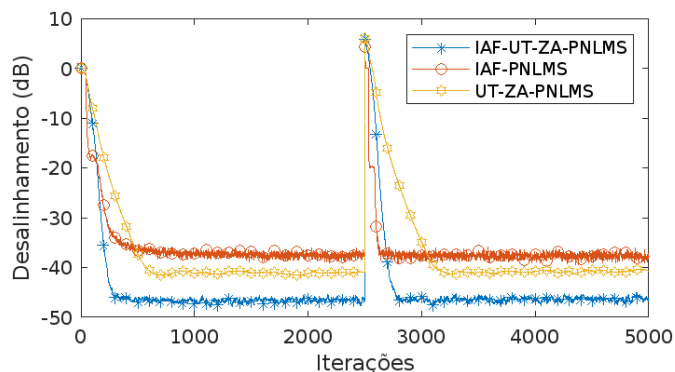


Figura 2. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Nos resultados apresentados na Figura 2, é possível observar que o algoritmo proposto nesse artigo, nos transitórios tanto antes quanto depois da perturbação, tem um decaimento similar ao do IAF-PNLMS, porém em regime estacionário, a proposta apresenta menor nível de desalinhamento.

Para uma análise minuciosa, nas Figuras 3 a 5, observa-se o comportamento dos ganhos individuais dos coeficientes ativos  $\{30, 85\}$  e de um inativo  $\{2\}$ . No caso do ganho inativo  $g_2(n)$ , note que os ganhos fornecidos pelos algoritmos do tipo UT apresentam uma limitação superior quando comparados aos ganhos fornecidos pelo IAF-PNLMS, conforme ilustra a Figura 3.

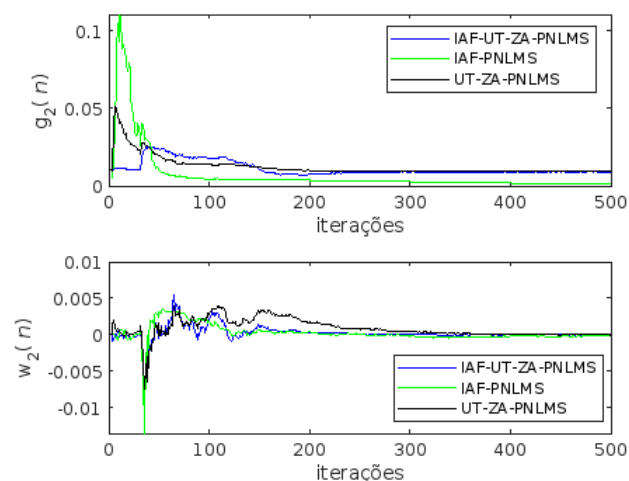


Figura 3. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Comparando o comportamento dos ganhos associados ao coeficiente ativo  $w_{30}(n)$ , observa-se que o algoritmo proposto impõe menores ganhos ao coeficiente ativo de maior magnitude comparado ao algoritmo IAF-PNLMS; entretanto, os ganhos fornecidos pela proposta são maiores do aqueles fornecidos pelo UT-ZA-PNLMS (veja a Figura 4). O comportamento do algoritmo proposto está de acordo com o esperado, isto é, uma distribuição de ganhos intermediária, ficando entre os algoritmos UT-ZA-PNLMS e IAF-PNLMS.

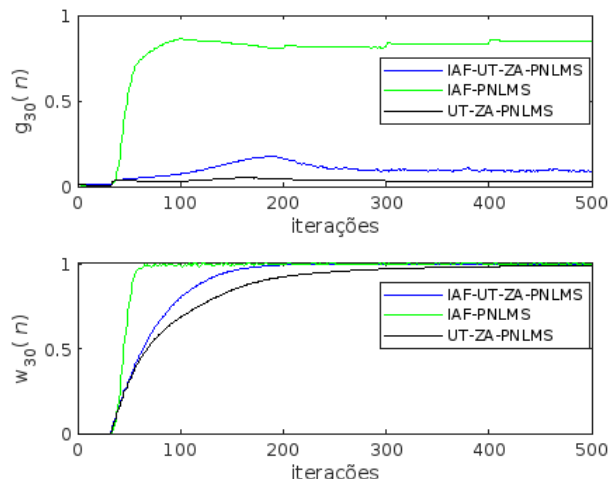


Figura 4. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Da Figura 4, note que o algoritmo IAF-PNLMS impõe maior ganho relativo ao coeficiente  $w_{30}(n)$  e com isso a convergência deste coeficiente é mais rápida do que os demais. Nesse caso, em especial, a limitação dos ganhos fornecidos pelo IAF-UT-ZA-PNLMS e UT-ZA-PNLMS retarda a convergência desse coeficiente.

Na Figura 5, o ganho  $g_{85}(n)$  associado ao coeficiente ativo  $w_{85}(n)$  é analisado. Nota-se que os resultados apresentados pelo o IAF-UT-ZA-PNLMS em termos de velocidade de convergência é superior aos demais algoritmos devido à redistribuição de ganhos imposta pela proposta.

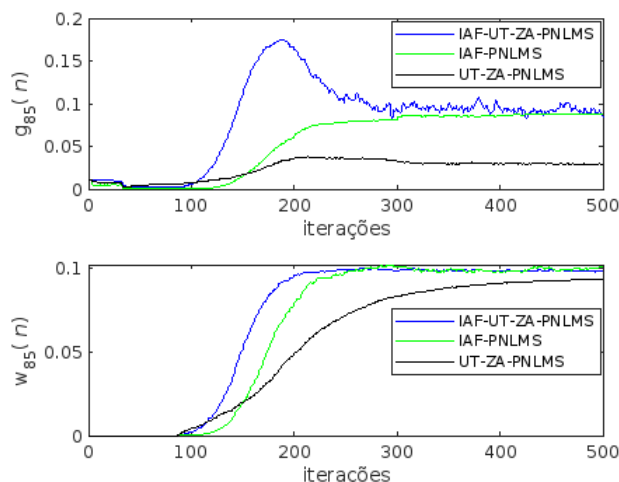


Figura 5. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

### 5.2 Exemplo 2

Neste exemplo, considera-se a identificação de  $\mathbf{p}$ ,  $\{0.1, 1.0, -0.5, 0.1\}$  nas posições  $\{1, 30, 35, 85\}$ . Após  $n = 2500$ , a planta  $\mathbf{p}$  sofre uma perturbação tornando-se  $-\mathbf{p}$ . O grau de esparsidade desta planta é  $S(\mathbf{p}) = 0.9435$ .

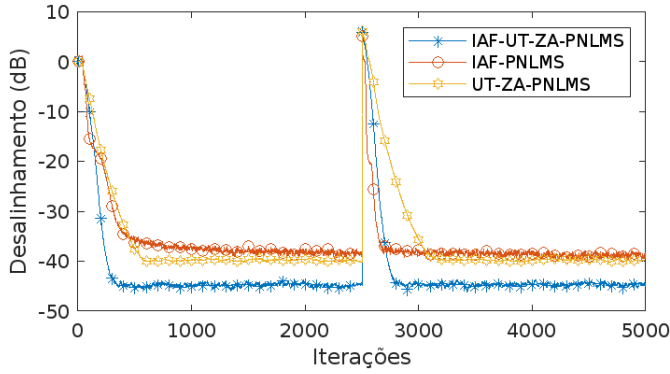


Figura 6. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Os resultados do Exemplo 2 ratificam o que foi observado na Figura 2, ou seja, o algoritmo proposto preserva a convergência rápida do IAF-PNLMS e o menor desalinhamento, tanto antes quanto depois da perturbação.

### 5.3 Exemplo 3

Para o terceiro exemplo, realiza-se a estimação da planta  $\mathbf{p}$  com coeficientes ativos  $\{0.1, 1.0, -0.5, 0.1, -0.1\}$  nas posições  $\{1, 30, 35, 85, 95\}$ . Após  $n = 2500$ , a planta  $\mathbf{p}$  sofre uma perturbação tornando-se  $-\mathbf{p}$ . O grau de esparsidade da resposta ao impulso para essa planta é  $S(\mathbf{p}) = 0,9343$ .

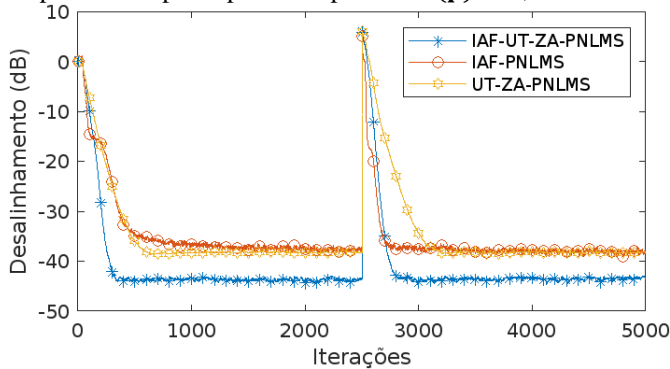


Figura 7. Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

De acordo com os testes realizados para este cenário de simulação (Exemplo 3), observa-se da Figura 7 que a característica em manter a convergência elevada e o menor desalinhamento possível em estado estacionário permanece. Tal característica pode ser observada também antes e depois da perturbação.

### 5.4 Exemplo 4

Neste exemplo, a capacidade de rastreamento dos algoritmos IAF-PNLMS, UT-ZA-PNLMS e IAF-UT-ZA-PNLMS é avaliada. Aqui, a simulação começa com a planta  $\mathbf{p}$  com coeficientes ativos  $\{0.1, 1.0, -0.5, 0.1\}$  nas posições  $\{1, 30, 35, 85\}$ , porém dois de seus coeficientes são modulados de acordo com a Equação (57), proposta em Martin (2002) e adaptada por Souza et al. (2010).

$$q_i = \begin{cases} p_i \left[ 1 + 0.5 \operatorname{sen} \left( \frac{2\pi n}{256} \right) \right] & i = 1, 30 \\ p_i & i = 35, 85 \\ 0 & \text{caso contrário} \end{cases} \quad (57)$$

Observando o desempenho dos algoritmos quanto ao rastreamento do primeiro coeficiente da planta (Figura 8), nota-se uma superioridade dos algoritmos IAF-PNLMS e do algoritmo proposto nesse artigo.

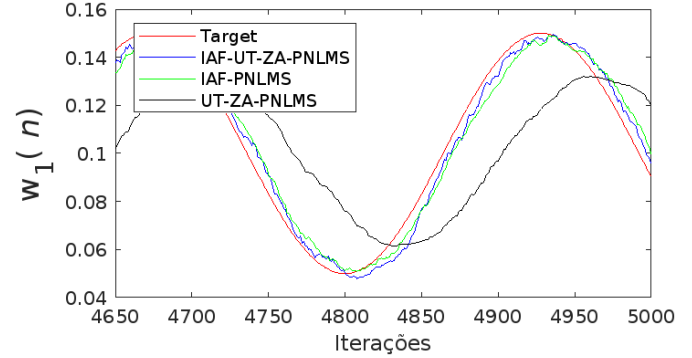


Figura 8. Comportamento de  $w_1(n)$  para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Uma análise considerando um coeficiente inativo também foi realizada, como mostra a Figura 9.

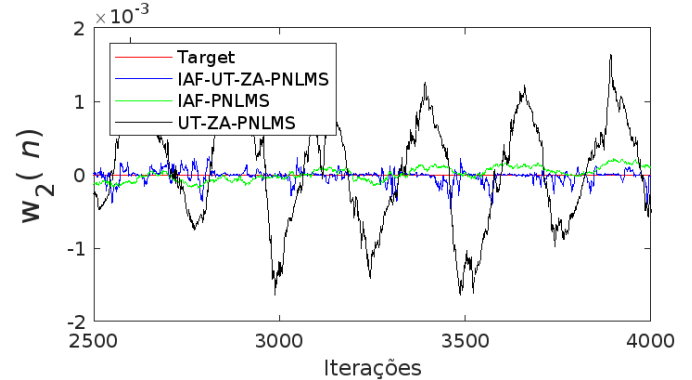


Figura 9. Comportamento do coeficiente inativo  $w_2(n)$  para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

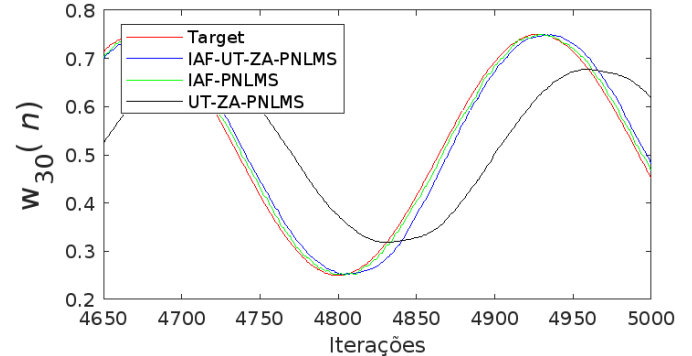


Figura 10. Comportamento de  $w_{30}(n)$  para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

Verifica-se da Figura 9 que o algoritmo IAF-UT-ZA-PNLMS juntamente com o IAF-UT-ZA-PNLMS convergem rapidamente no início das interações para o coeficiente inativo ilustrado.

O coeficiente ativo  $w_{30}(n)$  é observado na Figura 10, onde evidencia-se mais uma vez a superioridade do IAF-PNLMS e do proposto em comparação ao UT-ZA-PNLMS.

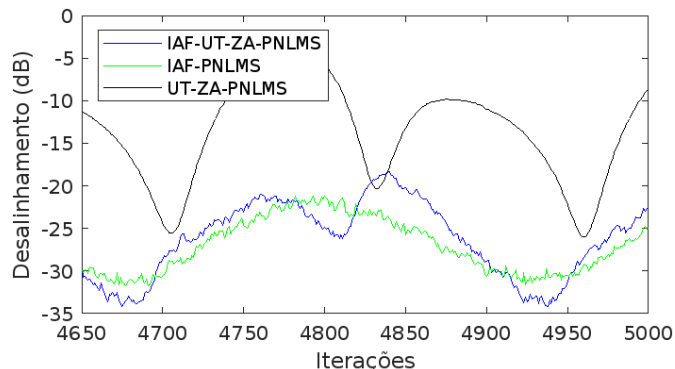


Figura 11. Curvas de desalinhamento normalizado durante a tarefa de rastreamento para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com  $\mu = 0.5$ ,  $\epsilon = \delta_N = 0.001$ ,  $\rho = 65 \cdot 10^{-6}$ ,  $\lambda_d = \lambda_e = 0.95$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.02$ ,  $\delta_g = \rho_g = 0.01$ ,  $\epsilon = 0.01$  e  $f_i(0) = 10^{-3}$ . Para o IAF-PNLMS,  $\xi = 0.001$  e  $f_i(0) = 10^{-4}$ .

A fim de se fazer uma análise mais geral do rastreamento, apresenta-se na Figura 11 o desalinhamento normalizado dos algoritmos durante o processo de adaptação.

## 6. CONCLUSÕES

Neste trabalho um algoritmo que combina as melhores características dos algoritmos UT-ZA-PNLMS e IAF-PNLMS foi proposto. Nesta proposta o limite superior adaptativo para cálculo dos ganhos do filtro UT-ZA-PNLMS foi combinado com forma de cálculo dos fatores de ativação individual para cada coeficiente, o que é característico do IAF-PNLMS. O algoritmo proposto, denominado IAF-UT-ZA-PNLMS, apresentou convergência semelhante ao IAF-PNLMS e comportamento em estado estacionário condizente ao apresentado pelo UT-ZA-PNLMS. Tais resultados preliminares podem ser interpretados como um algoritmo melhorado para plantas esparsas. Como trabalhos futuros, pretende-se realizar estudo teóricos de convergência do algoritmo proposto, visando generalizar os resultados obtidos.

## AGRADECIMENTOS

Os autores agradecem à CAPES, FAPEMA, CNPq e UFMA.

## REFERÊNCIAS

Benesty, J. and Gay, S.L. (2002, May). An improved PNLMS algorithm. In *2002 IEEE international conference on acoustics, speech, and signal processing* (Vol. 2, pp. II-1881). IEEE.

Branco, C. A. S. C. (2016). Algoritmos adaptativos LMS normalizados proporcionais: proposta de novos algoritmos para identificação de plantas esparsas.

Chen, Y., Gu, Y. and Hero, A.O. (2009, April). Sparse LMS for system identification. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3125-3128). IEEE.

Das, R. L. and M. Chakraborty. (2016, July). Improving the performance of the PNLMS algorithm using 1 norm regularization. *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 7, pp. 1280-1290, Jul.

Das, R.L. and Trivedi, V. (2020, October). An adaptive upper threshold based gain function for the ZA-PNLMS algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(10), pp.2274-2278.

De Souza, F.D.C., Seara, R. and Morgan, D.R. (2012). An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification. *IEEE transactions on signal processing*, 60(6), pp.3301-3307.

De Souza, F.D.C., Tobias, O.J., Seara, R. and Morgan, D.R. (2009, August). Alternative approach for computing the activation factor of the PNLMS algorithm. In *2009 17th European Signal Processing Conference* (pp. 2633-2637). IEEE.

De Souza, F.D.C., Tobias, O.J., Seara, R. and Morgan, D.R. (2010). A PNLMS algorithm with individual activation factors. *IEEE Transactions on Signal Processing*, 58(4), pp.2036-2047.

Duttweiler, D.L. (2000). Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Transactions on speech and audio processing*, 8(5), pp.508-518.

Gay, S.L. (1998, November). An efficient, fast converging adaptive filter for network echo cancellation. In *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284)* (Vol. 1, pp. 394-398). IEEE.

Gay, S.L. and Douglas, S.C. (2002, May). Normalized natural gradient adaptive filtering for sparse and non-sparse systems. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 2, pp. II-1405). IEEE.

Haykin, S.S. (2014). *Adaptive filter theory*. Pearson Education India.

Huang, Y., Benesty, J., and Chen, J. (2006). *Acoustic MIMO signal processing*. Springer Science & Business Media.

Li, Y., Wang, Y., Albu, F. and Jiang, J. (2017). A general zero attraction proportionate normalized maximum correntropy criterion algorithm for sparse system identification. *Symmetry*, 9(10), p.229.

Martin, R.K., Sethares, W.A., Williamson, R.C. and Johnson, C.R. (2002). Exploiting sparsity in adaptive filters. *IEEE Transactions on Signal Processing*, 50(8), pp.1883-1894.

Müller, C. and Piber, P. R.V. (2006). Uma Comparação do Desempenho da Filtragem Adaptativa em Banda Cheia e a Filtragem Adaptativa em Subbandas, Aplicada ao Cancelamento do Eco Elétrico. *Revista Tecnologia e Tendências*, v. 5, p. 52-71.

Paleologu, C., Benesty, J. and Ciochina, S. (2010). Sparse adaptive filters for echo cancellation. *Synthesis Lectures on Speech and Audio Processing*, 6(1), pp.1-124.

Widrow, B. and Stearns, S. D. (1985). *Adaptive Signal Processing*. Englewood Cliffs, New Jersey: Prentice Hall.