

Metodologia para Projeto de Sistemas *Fuzzy* no Domínio Hipercomplexo Aplicada em Filtragem Adaptativa

Daiana Caroline dos Santos Gomes*
Ginalber Luiz de Oliveira Serra**

* Universidade Federal do Maranhão, Av. dos Portugueses, 1966, Bacanga, São Luís, Maranhão (e-mail: daianagomes159@gmail.com).

** Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, Av. Getúlio Vargas, 04, Monte Castelo, São Luís, Maranhão, (e-mail: ginalber@ifma.edu.br)

Abstract: In this paper, a *fuzzy* descent method applied to adaptive filtering algorithms in the hypercomplex domain is proposed. The adopted methodology consists of Mamdani *fuzzy* inference system design, in the hypercomplex domain, applied to the computation of adaptation step size to be adopted for the convergence of the adaptive filtering algorithm. The partitioning of the universe of discourse regarding to the components of input and output hypercomplex variables of *fuzzy* inference system, to define the rule base, is performed by the expert's knowledge about the application. For validation purposes, the proposed methodology was applied to a *Normalized Least Mean Square* (NLMS) filter variation, which is based on the minimization of cost function of the *Recursive Least Square* (RLS) algorithm. Computational results and comparative analysis illustrates the efficiency and applicability of the proposed methodology, providing lower computational cost and lower final misfit for estimating the optimal weights of the adaptive filter.

Resumo: Neste artigo é proposto um método de descida *fuzzy* aplicado a algoritmos de filtragem adaptativa no domínio hipercomplexo. A metodologia adotada consiste no projeto de um sistema de inferência *fuzzy* do tipo Mamdani, no domínio hipercomplexo, aplicado no cálculo do tamanho do passo de adaptação a ser adotado para a convergência de algoritmos de filtragem adaptativa. O particionamento do universo de discurso referente às componentes das variáveis hipercomplexas de entrada e saída do sistema de inferência *fuzzy*, para definição da base de regras, é realizado de acordo com o conhecimento do especialista sobre a aplicação. Para fins de validação, a metodologia proposta foi aplicada à uma variação do filtro *Normalized Least Mean Square* (NLMS), a qual é baseada na minimização da função custo do algoritmo *Recursive Least Square* (RLS). Resultados computacionais e análise comparativa ilustram a eficiência e aplicabilidade da metodologia proposta quando comparada a abordagens relevantes da literatura, proporcionando menor custo computacional e menor desajuste final para estimação dos pesos ótimos do filtro adaptativo.

Keywords: Intelligent systems; Fuzzy inference system; Adaptive Filters; Normalized Least Mean Square Algorithm; Hypercomplex domain.

Palavras-chaves: Sistemas Inteligentes; Sistema de inferência *fuzzy*; Filtros adaptativos; Algoritmo *Normalized Least Mean Square*; Domínio hipercomplexo.

1. INTRODUÇÃO

Devido sua simplicidade e robustez, o algoritmo *Least Mean Square* (LMS) foi largamente aplicado desde que foi proposto em 1959, servindo de base para o desenvolvimento de diversas outras técnicas de filtragem adaptativa (da Costa et al., 2021; Jalal et al., 2020; Zhang et al., 2020). O modo de operação *online* e adaptativo do algoritmo LMS torna seu uso adequado para o processamento de dados não-estacionários, com aplicações em diversos campos de pesquisa envolvendo redução de ruído, cancelamento de eco, equalização de canais, processamento de sinais de

radars, dentre outras (Huang et al., 2018; Tian et al., 2018).

Uma outra técnica também bastante utilizada em problemas de otimização é o algoritmo *Recursive Least Square* (RLS), o qual, assim como o LMS, também converge para a solução ótima, através da minimização do erro médio quadrático de estimação (Sayed, 2003). O algoritmo RLS é capaz de convergir para a solução ótima mais rapidamente que o algoritmo LMS e também apresenta um melhor desempenho de rastreamento em ambientes não-estacionários (Haykin, 2014). Por outro lado, o RLS apresenta as limitações de ser menos numericamente estável e possuir

um elevado custo computacional, ambas devido ao uso do lema de inversão de matrizes (Sayed, 2003). Embora exista uma grande quantidade de estudos envolvendo a aplicação dos algoritmos de otimização LMS e RLS atualmente, muitos pesquisadores têm se dedicado a propor melhorias para ambos os algoritmos (Lunglmayr et al., 2020). Em (Variddhisai and Mandic, 2017), uma metodologia unificada para o projeto de filtros adaptativos foi proposta combinando a alta taxa de convergência do RLS com a simplicidade computacional do LMS, obtendo-se um desempenho competitivo ao algoritmo RLS padrão.

No contexto de filtragem adaptativa e estimação de estados, uma área de pesquisa que têm chamado a atenção de muitos pesquisadores é a inteligência computacional onde os sistemas *fuzzy* vêm se destacando cada vez mais em aplicações na área de processamento de sinais, modelagem e controle (Lan et al., 2020; Feng et al., 2021). Este interesse crescente pelos sistemas *fuzzy* é devido a sua estrutura simplificada baseada em regras capaz de fazer a aproximação de funções e tratamento de não linearidades e incertezas (Serra, 2012).

Alguns estudos envolvendo a utilização de sistemas *fuzzy* na área de filtragem adaptativa já foram desenvolvidos no intuito de melhorar o desempenho e reduzir o custo computacional dos algoritmos propostos (Abdygalievich et al., 2019; Benhalima et al., 2019). Entretanto, com o aumento da complexidade das aplicações do mundo real, a proposição de metodologias em filtragem adaptativa capazes de agregar as vantagens do RLS e LMS tradicionais e, ainda, realizar o processamento de incertezas presentes nos dados experimentais, ainda está em aberto. Diante disso, a metodologia apresentada neste artigo está baseada no projeto de um sistema *fuzzy*, no domínio hipercomplexo, aplicado na determinação do tamanho do passo de adaptação a ser adotado para a convergência do filtro LMS normalizado, o qual minimiza a função custo do algoritmo RLS, a fim de garantir maior robustez ao filtro desenvolvido em (Variddhisai and Mandic, 2017). As principais contribuições deste artigo que garantem a originalidade da metodologia proposta são:

- Desenvolvimento de um método de descida *fuzzy* aplicado a algoritmos de filtragem adaptativa, o qual é capaz de determinar o melhor tamanho do passo de adaptação em função das incertezas presentes em dados experimentais.
- Formulação de uma metodologia para o projeto de um sistema *fuzzy* no domínio hipercomplexo, com aplicações em filtragem adaptativa, o que oferece a vantagem de aumentar o campo de aplicações para o método proposto.

2. METODOLOGIA

Nesta seção, será apresentada a metodologia para o projeto do sistema *fuzzy* para determinação do tamanho do passo de adaptação aplicado ao filtro NLMS baseado em RLS.

2.1 Filtro Adaptativo NLMS baseado em RLS

A formulação matemática para o filtro adaptativo NLMS baseado no algoritmo RLS, no domínio hipercomplexo, foi

proposta pela primeira vez em (Variddhisai and Mandic, 2017), onde a função custo a ser minimizada pelo algoritmo NLMS é dada por:

$$J_n(\hat{\mathbf{w}}) \approx 2\Phi_n(\hat{\mathbf{w}}) \quad (1)$$

onde $\Phi_n(\hat{\mathbf{w}})$ corresponde à função custo do algoritmo RLS dada pela seguinte expressão:

$$\Phi_n(\hat{\mathbf{w}}) = \sum_{k=1}^n \lambda^{n-k} \|\xi_k(\hat{\mathbf{w}})\|^2 \quad (2)$$

onde $0 \leq \lambda \leq 1$ é o fator de esquecimento usado para reduzir o efeito de amostras de dados mais antigas e $\xi_k(\hat{\mathbf{w}})$ é o erro de estimação. Seja y_n o sinal desejado (saída) e x_n o sinal de entrada (variáveis hipercomplexas pertencentes ao domínio \mathbb{H}), $n = 1, \dots, N$, a estimação linear de y_n , denotado por \hat{y}_n , pode ser expressa como

$$\hat{y}_n = \langle \hat{\mathbf{w}}, \mathbf{q}_n \rangle = \hat{\mathbf{w}}^H \mathbf{q}_n \quad (3)$$

onde $\hat{\mathbf{w}}$ é uma estimativa para a solução ótima \mathbf{w}^{op} e \mathbf{q}_n é definido como (Haykin, 2014):

$$\mathbf{q}_n = [\mathbf{x}_n^T \quad \mathbf{x}_n^{iT} \quad \mathbf{x}_n^{jT} \quad \mathbf{x}_n^{kT}]^T \quad (4)$$

onde $\mathbf{x}_n = [x_n, x_{n-1}, \dots, x_{n-M+1}]$ representa o vetor de regressores para um filtro de tamanho M . O erro de estimação será então dado por:

$$\xi_n(\hat{\mathbf{w}}) = y_n - \hat{y}_n = y_n - \hat{\mathbf{w}}^H \mathbf{q}_n \quad (5)$$

Substituindo as Eqs. (5) e (3) em (1), obtém-se

$$J_n(\hat{\mathbf{w}}) = \hat{\mathbf{w}}^H \mathbf{R}_n \hat{\mathbf{w}} - 2\Re\{\hat{\mathbf{w}}^H \mathbf{r}_n\} + \sum_{k=1}^n \lambda^{n-k} |y_k|^2 \quad (6)$$

onde $\Re\{\cdot\}$ e H são, respectivamente, os operadores de parte real e transposto Hermitiano tal que

$$\mathbf{R}_n = \sum_{k=1}^n \lambda^{n-k} \mathbf{q}_k \mathbf{q}_k^H = \lambda \mathbf{R}_{n-1} + \mathbf{q}_n \mathbf{q}_n^H \quad (7)$$

$$\mathbf{r}_n = \sum_{k=1}^n \lambda^{n-k} \mathbf{q}_k y_k^* = \lambda \mathbf{r}_{n-1} + \mathbf{q}_n y_n^* \quad (8)$$

O procedimento de otimização realizado pelo algoritmo NLMS, para a minimização da sua função custo dada na Eq. (6), é realizado a partir do cálculo do gradiente *a priori* como

$$\mathbf{g}_{n|n-1} = \nabla_{\hat{\mathbf{w}}^*} J_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}_{n-1}} = \mathbf{R}_n \mathbf{w}_{n-1} - \mathbf{r}_n \quad (9)$$

A expressão para a atualização dos pesos hipercomplexos é dada por:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{d}_n \alpha_n \quad (10)$$

onde \mathbf{d}_n é a direção de descida e $\alpha_n \in \mathbb{R}$ é o tamanho do passo aplicado na iteração n do algoritmo, dado por:

$$\alpha_n = -\frac{\langle \mathbf{d}_n, \mathbf{g}_{n|n-1} \rangle}{\langle \mathbf{d}_n, \mathbf{h}_{n|n} \rangle} \quad (11)$$

onde

$$\mathbf{h}_{n|n} = \mathbf{R}_n \mathbf{d}_n \quad (12)$$

Em seguida, o vetor gradiente é atualizado para o gradiente *a posteriori* $\mathbf{g}_{n|n}$ da seguinte forma:

$$\mathbf{g}_{n|n} = \nabla_{\hat{\mathbf{w}}} * J_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}} = \mathbf{R}_n \mathbf{w}_n - \mathbf{r}_n \quad (13)$$

O algoritmo recursivo de atualização é dado por:

$$\mathbf{g}_{n|n-1} = \lambda \mathbf{g}_{n-1|n-1} - \mathbf{q}_n e_n^* \quad (14)$$

$$\mathbf{g}_{n|n} = \mathbf{g}_{n|n-1} + \mathbf{h}_{n|n} \alpha_n \quad (15)$$

2.2 Escolha da Direção de Descida

Em processos não-estacionários, devido a matriz de autocorrelação \mathbf{R}_n ser estatisticamente variante no tempo, muitos algoritmos de descida tornam-se ineficientes para a estimação do vetor de pesos ótimo \mathbf{w}^{op} (Chang and Willson, 2000). Diante disso, novos métodos tem sido desenvolvidos não só para superar esta questão como também garantir uma rápida velocidade de convergência e um pequeno desajuste final a estimativa de pesos ótimo. Dois métodos distintos de descida comumente adotados em filtragem adaptativa são o Conjugado de Markov e o *Steepest Descent*, os quais são brevemente descritos a seguir.

Conjugado de Markov (CM) A direção de descida para o método de Conjugado de Markov é expresso por (Varidhisai and Mandic, 2017):

$$\mathbf{d}_n = -\mathbf{g}_{n|n-1} + \mathbf{d}_{n-1} \beta_n \quad (16)$$

onde β_n é dado por:

$$\beta_n = \frac{\langle \mathbf{d}_{n-1}, \mathbf{v}_n \rangle}{\langle \mathbf{d}_{n-1}, \mathbf{h}_{n|n-1} \rangle} \quad (17)$$

com

$$\mathbf{h}_{n|n-1} = \mathbf{R}_n \mathbf{d}_{n-1} \quad (18)$$

$$\mathbf{v}_n = \mathbf{R}_n \mathbf{g}_{n|n-1} \quad (19)$$

chegando-se ao seguinte algoritmo recursivo de atualização:

$$\mathbf{h}_{n|n-1} = \lambda \mathbf{h}_{n-1|n-1} - \mathbf{q}_n (\mathbf{q}_n^H \mathbf{d}_{n-1}) \quad (20)$$

$$\mathbf{h}_{n|n} = -\mathbf{v}_n + \mathbf{h}_{n|n-1} \beta_n \quad (21)$$

Steepest Descent (SD) Este método é resultado de uma simplificação do Conjugado de Markov, onde $\beta_n = 0, \forall n$ resultando em:

$$\mathbf{d}_n = -\mathbf{g}_{n|n-1} \quad (22)$$

e o tamanho do passo é obtido por:

$$\alpha_n = \frac{1}{\pi_n + \epsilon_n}, \quad \pi_n = \frac{\langle \mathbf{g}_{n|n-1}, \mathbf{v}_n \rangle}{\|\mathbf{g}_{n-1}\|^2} \quad (23)$$

onde $\epsilon_n \in \mathbb{R}$ é um regularizador adaptativo (Mandic, 2004).

2.3 Parametrização do Método de Descida Fuzzy no Domínio Hipercomplexo

O modelo *fuzzy* do tipo Mamdani adotado para implementação do método de descida, no domínio dos hipercomplexo, apresenta a l [l=1,2,...,r]-ésima regra *fuzzy* dada por:

$$\text{SE } z_{1_n} \text{ É } M_{l1} \text{ e } \dots \text{ e } z_{p_n} \text{ É } M_{lq} \text{ ENTÃO } \alpha_n^l \text{ É } A_l \quad (24)$$

onde $z_{1_n} \dots z_{p_n} \in \mathbb{H}$ são as variáveis linguísticas do antecedente, $M_{lq} = M_{lq}^R + iM_{lq}^I + jM_{lq}^J + kM_{lq}^K$ são os respectivos termos linguísticos hipercomplexos da proposição do antecedente, $\alpha_n^l \in \mathbb{H}$ é a variável de saída e $A_l = A_l^R + iA_l^I + jA_l^J + kA_l^K$ são os termos linguísticos hipercomplexos definidos na proposição do consequente.

A estrutura para o sistema de inferência *fuzzy* adotado é mostrada na Figura 1, cujas etapas são detalhadas a seguir.

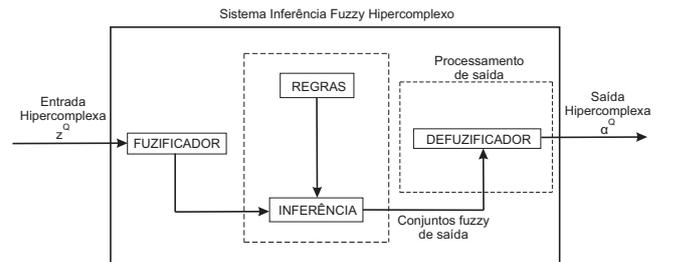


Figura 1. Sistema de inferência *fuzzy* Mamdani no domínio hipercomplexo.

Fuzificador Esta etapa é responsável por realizar o mapeamento das variáveis de entrada $z_{1_n} \dots z_{p_n}$ nos conjuntos *fuzzy* hipercomplexos definidos no antecedente das regras para a obtenção dos graus de pertinência correspondente a l -ésima regra. Para o caso particular onde as funções de pertinência são triangulares, o grau de pertinência em cada conjunto é obtido por (Hata et al., 2013):

$$M_{lq}^Q(z_{lq}^Q; a_{lq}^Q, b_{lq}^Q, c_{lq}^Q) = \begin{cases} 0, & z_{lq}^Q \leq a_{lq}^Q \\ \frac{z_{lq}^Q - a_{lq}^Q}{b_{lq}^Q - a_{lq}^Q}, & a_{lq}^Q \leq z_{lq}^Q \leq b_{lq}^Q \\ \frac{c_{lq}^Q - z_{lq}^Q}{c_{lq}^Q - b_{lq}^Q}, & b_{lq}^Q \leq z_{lq}^Q \leq c_{lq}^Q \\ 0, & c_{lq}^Q \leq z_{lq}^Q \end{cases} \quad (25)$$

onde $Q = R, I, J, K$ denota as partes real e imaginárias das variáveis hipercomplexas e a_{lq}^Q , b_{lq}^Q e c_{lq}^Q são os pontos inicial, de pico e final da função de pertinência triangular, respectivamente. A partir da Equação (25), os graus de pertinência são valores independentes para as partes real e imaginárias da variável linguística de entrada.

Base de Regras Este bloco contém as r regras do sistema de inferência *fuzzy* na forma da Eq. (24), cujas proposições do antecedente e consequente são formados por conjuntos *fuzzy* hipercomplexos linguísticos.

Inferência A etapa de inferência realiza uma composição das regras *fuzzy* definidas na base de regras. Os graus de ativação das regras *fuzzy* são processados na etapa de inferência da seguinte forma (Hata et al., 2013, 2018):

$$\mu_l^Q = \prod_{n=1}^q M_{l_n}^Q \quad (26)$$

resultando no seguinte grau de ativação para a l -ésima regra:

$$\mu_l^Q = \mu_l^R + i\mu_l^I + j\mu_l^J + k\mu_l^K \quad (27)$$

Defuzificador A etapa de defuzificação é responsável por extrair um valor numérico para a saída do sistema *fuzzy*. O valor de saída para $\alpha_k \in \mathbb{H}$ é obtido a partir do método de centro de gravidade, definido por (Hata et al., 2018; Moura et al., 2013):

$$\alpha_k^Q = \frac{\sum_{l=1}^r (\mu_l^R + i\mu_l^I + j\mu_l^J + k\mu_l^K)(A_l^R + iA_l^I + jA_l^J + kA_l^K)}{\sum_{l=1}^r (\mu_l^R + i\mu_l^I + j\mu_l^J + k\mu_l^K)} \quad (28)$$

Por fim, um valor no domínio real \mathbb{R} para a variável de saída α_k é obtido como segue:

$$\alpha_k = (\alpha_k^R - \alpha_k^I)^2 + (\alpha_k^J - \alpha_k^K)^2 \quad (29)$$

A implementação do NLMS *fuzzy* baseado em RLS, o qual é referenciado como m -FNLMS, isto é, o NLMS *fuzzy* com memória, de acordo com a metodologia proposta, é realizada conforme mostrado no Algoritmo 1.

Algoritmo 1: Filtro Adaptativo NLMS *Fuzzy* com Memória (m -FNLMS) no Domínio Hipercomplexo

Entrada: $x_n, y_n, M, \lambda, \epsilon$ e δ

Saída: \hat{y} e \mathbf{w}^{op}

início

$\mathbf{w}_0 = \mathbf{d}_0 = \mathbf{g}_{0|0} = \mathbf{h}_{0|0}$ e $\mathbf{R}_0 = 0$; $n = 0$

repita

$n = n + 1$;

Atualizar \mathbf{q}_n de acordo com a Eq. (4);

$\hat{y} = (\mathbf{w}_{n-1})^H \mathbf{q}_n^H$;

$e_n = y_n - \hat{y}$;

$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \mathbf{q}_n \mathbf{q}_n^H$;

$\mathbf{g}_{n|n-1} = \lambda \mathbf{g}_{n-1|n-1} - \mathbf{q}_n e_n^*$;

$\mathbf{v}_n = \mathbf{R}_n \mathbf{g}_{n|n-1}$;

Esquema de Descida Fuzzy

$\mathbf{d}_n = -\mathbf{g}_{n|n-1}$;

Calcular α_k^Q através da Eq. (28);

$\alpha_k = (\alpha_k^R - \alpha_k^I)^2 + (\alpha_k^J - \alpha_k^K)^2$

até $\|e_n\| > \delta$ ou $n = N$;

fim

$\mathbf{w}^{op} = \mathbf{w}_n$

3. RESULTADOS COMPUTACIONAIS

Nesta seção, resultados da metodologia proposta aplicada na estimação do vetor de pesos ótimo de um modelo de média móvel, no domínio hipercomplexo, são apresentados. Para ilustrar a eficiência da metodologia proposta, será realizada uma análise comparativa entre o m -FNLMS proposto e o filtro adaptativo apresentado em (Varidhdisai and Mandic, 2017), o qual utiliza os métodos de descida pelo Conjugado de Markov e *Steepest Descent*. A metodologia proposta também será comparada com outros algoritmos de filtragem adaptativa já consolidados na literatura. O modelo de média móvel utilizado foi o *Widely Linear Quaternion Moving Average* de ordem 3 - WLQMA(3), descrito por (Navarro-Moreno, 2008):

$$y(n) = \sum_{j=0}^{\kappa} w_{1,j} x_{n-j} + \sum_{j=0}^{\kappa} w_{2,j} x_{n-j}^* \quad (30)$$

onde κ é a ordem do modelo, $w_{1,j}$ e $w_{2,j}$ são os coeficientes do modelo, x_{n-j} e x_{n-j}^* denotam, respectivamente, os regressores que compõe o vetor de entrada para o modelo e seus complexos conjugados. A entrada x_n consiste de 200 amostras com distribuição $\mathcal{N}(0, 1)$ para cada uma de suas componentes no domínio hipercomplexo (r, i, j, k), com um SNR moderado de 40 dB. Em cada experimento, foram obtidas as estimativas do vetor de pesos ótimo utilizando a metodologia proposta neste artigo, através da média dos resultados de 100 realizações independentes. A variável linguística de entrada para o sistema de inferência *fuzzy* hipercomplexo foi considerada como o módulo do erro de estimação $e_n^Q|_{Q=1,i,j,k}$. Foram definidos 3 conjuntos *fuzzy* com formato triangular no universo de discurso correspondente a cada uma das componentes real e complexas do erro $e_n^Q|_{Q=1,i,j,k}$, conforme ilustrado na Figura 2.

A variável de saída do sistema de inferência *fuzzy* corresponde ao tamanho do passo $\alpha_n \in \mathbb{H}$ a ser aplicado na iteração seguinte do algoritmo do filtro m -FNLMS. Foram

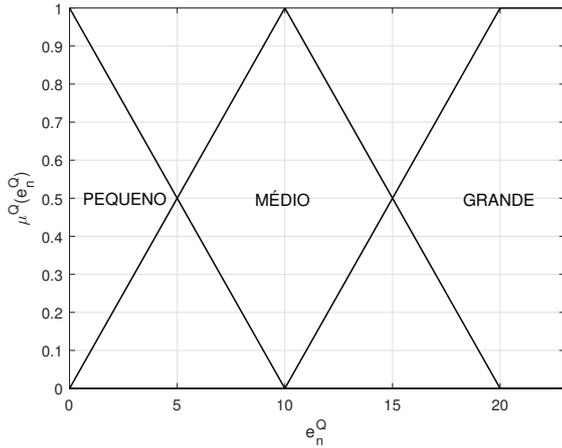


Figura 2. Funções de pertinência triangulares definidos sobre o universo de discurso da variável de entrada $e_n^Q |_{Q=1,i,j,k}$.

definidos três conjuntos *fuzzy* do tipo *singleton* para a variável de saída, os quais são mostrados na Figura 3. A

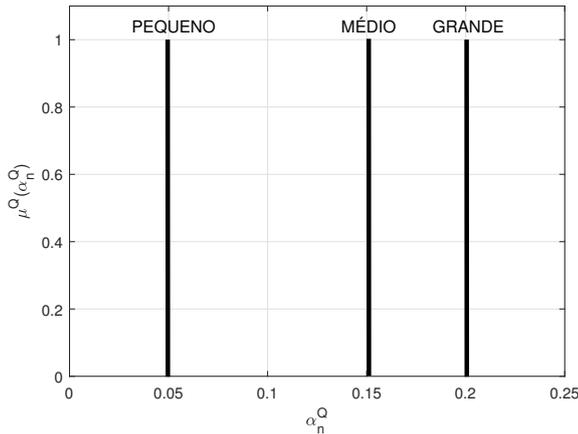


Figura 3. Funções de pertinência *fuzzy singleton* definidas sobre o universo de discurso da variável de saída $\alpha_n^Q |_{Q=1,i,j,k}$.

base de regras do sistema de inferência *fuzzy* Mamdani, obtida a partir do conhecimento do especialista, é dada por:

REGRA 1: SE e_n^Q É pequeno ENTÃO α_n^Q É grande

REGRA 2: SE e_n^Q É médio ENTÃO α_n^Q É médio

REGRA 3: SE e_n^Q É grande ENTÃO α_n^Q É pequeno

Para avaliar o desempenho dos resultados obtidos para estimação do vetor de pesos ótimo, foi utilizada a métrica do desajuste normalizado, η_m , medido em decibel *dB*, a qual é definida como:

$$\eta_m = \frac{\|\mathbf{w}_n - \mathbf{w}^{op}\|^2}{\|\mathbf{w}^{op}\|^2} \quad (dB) \quad (31)$$

onde \mathbf{w}_n é a estimação do n -ésimo elemento do vetor de pesos ótimo e \mathbf{w}^{op} é o seu valor ótimo.

3.1 Experimento 1

No primeiro experimento, o m -FNLMS foi comparado com o filtro adaptativo m -NLMS proposto em (Varidhisai and Mandic, 2017), o qual utiliza os esquemas de descida por Conjugado de Markov e *Steepest Descent*, bem como também foi comparado com os algoritmos NLMS e RLS padrão considerando o fator de esquecimento $\lambda = 0.95$ e $\epsilon = 0.01$, conforme mostrado na Figura 4. É possível perceber, através da Figura 4, que o m -FNLMS proposto neste artigo converge rapidamente para a solução ótima atingindo o menor desajuste final dentre todos os algoritmos avaliados. Na Figura 5 é mostrado o desempenho dos mesmos algoritmos considerando o fator de esquecimento como $\lambda = 0.99$. Para este caso, novamente o m -FNLMS proposto apresenta elevada taxa de convergência, apresentando o menor desajuste final na estimação do vetor de pesos ótimo. O acréscimo no valor de λ reflete na velocidade de convergência dos algoritmos avaliados neste primeiro experimento. A variação do tamanho do passo de adaptação $\alpha_n^Q |_{Q=1,i,j,k}$ durante a implementação da metodologia proposta neste artigo para o experimento 1 é mostrada na Figura 6.

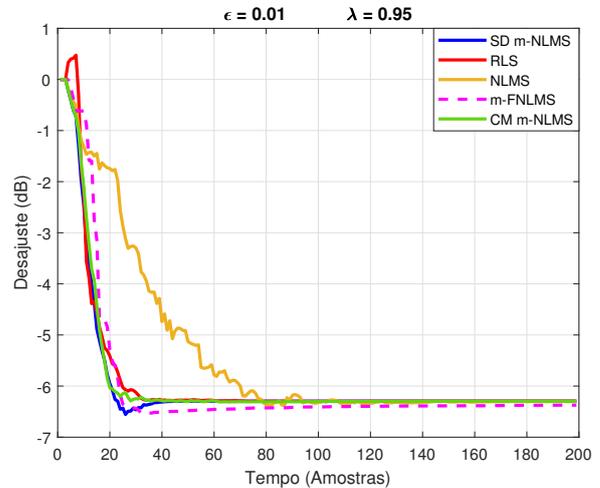


Figura 4. Comparação entre o desalinhamento dos algoritmos NLMS, RLS, m -NLMS e m -FMNLS, considerando $\lambda = 0.95$.

3.2 Experimento 2

Nas Figura 7 e 8 são mostrados os resultados para o segundo experimento realizado. Neste caso, foi examinado o desempenho do m -FNLMS comparado a algoritmos de passo fixo e variável considerando diferentes valores para o fator de esquecimento ($\lambda = 0.95$ e $\lambda = 0.99$), respectivamente. As rotinas de passo fixo consideradas foram o RLS e o m -NLMS com esquema de descida por *Steepest Descent*, enquanto que as rotinas de passo

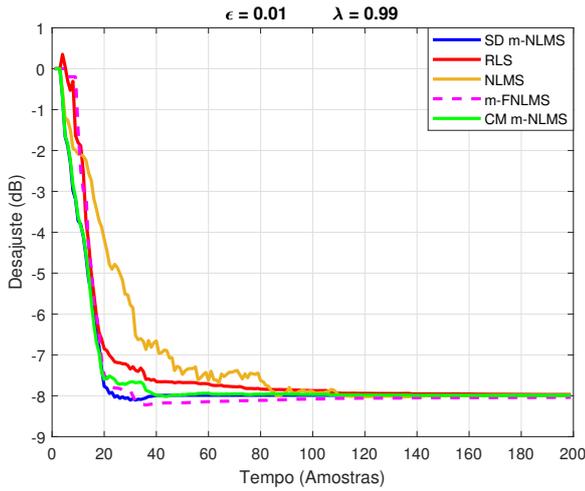


Figura 5. Comparação entre o desalinhamento dos algoritmos NLMS, RLS, m -NLMS e m -FMNLS, considerando $\lambda = 0.99$.

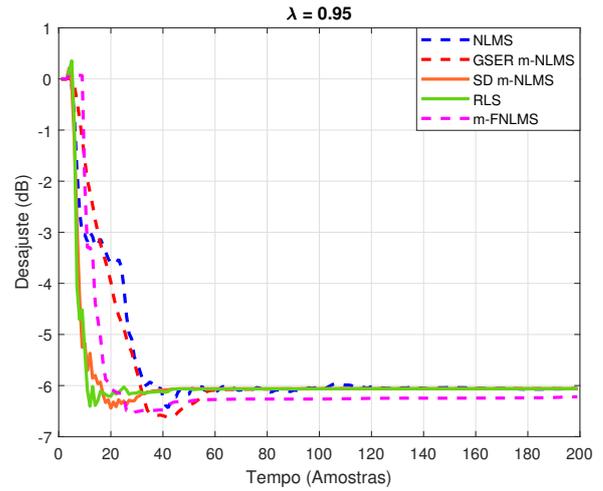


Figura 7. Comparação entre o desalinhamento dos algoritmos NLMS, RLS, m -NLMS e m -FMNLS considerando $\lambda = 0.95$: passo fixo (linha sólida); passo variável (linha tracejada).

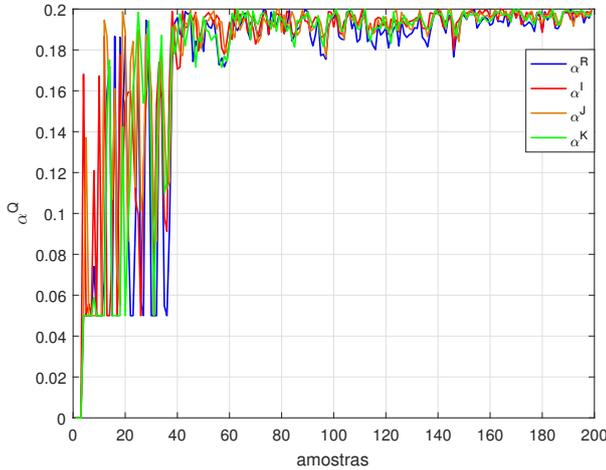


Figura 6. Variação do tamanho do passo de adaptação $\alpha_n^Q |_{Q=1,i,j,k}$.

variável foram o NLMS e o m -NLMS utilizando o esquema de descida *Generalized Square-Error-Regularized* - GSER (J. Lee and Huang, 2009). A partir das Figuras 7 e 8, e comparando-as com os resultados obtidos no experimento 1 (Figuras 4 e 5), percebe-se que o tamanho do passo variável reduz tanto o desajuste final como também a taxa de convergência do algoritmo. Neste segundo experimento, o m -FNLMS proposto convergiu mais rapidamente que o RLS e o m -NLMS com esquema de descida GSER, obtendo novamente o menor desajuste dentre todos os algoritmos avaliados.

3.3 Experimento 3

Outro método de descida também foi avaliado quando aplicado ao algoritmo m -NLMS proposto em (Varidhisai and Mandic, 2017). A técnica *Generalized Normalized Gradient Descent* - GNGD (Mandic, 2004) foi aplicada como método de descida no intuito de reduzir o desajuste final e obter uma velocidade de convergência maior para o m -NLMS. Na Figura 9 são mostrados os resultados para

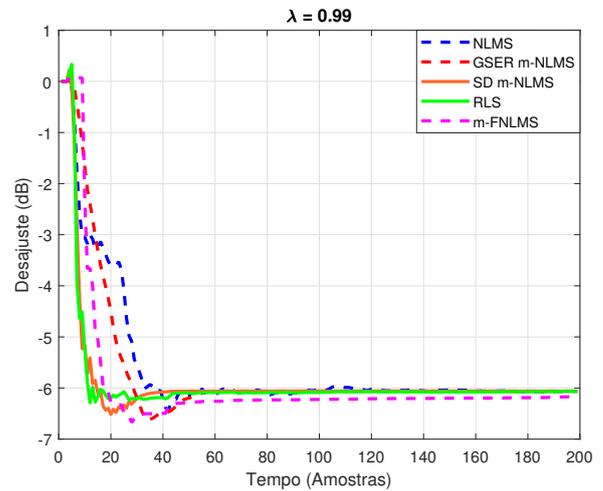


Figura 8. Comparação entre o desalinhamento dos algoritmos NLMS, RLS, m -NLMS e m -FMNLS considerando $\lambda = 0.99$: passo fixo (linha sólida); passo variável (linha tracejada).

este experimento, onde o m -FNLMS apresentou maior taxa de convergência que o m -NLMS com esquema de descida por GNGD, bem como apresentou menor desajuste final.

4. CONCLUSÃO

Neste artigo foi proposto um método de descida *fuzzy*, no domínio hipercomplexo, aplicado ao algoritmo de filtragem adaptativa m -FNLMS. A partir dos resultados obtidos, o m -FNLMS mostrou-se bastante eficiente quando aplicado na estimação do vetor de pesos ótimo de um modelo hipercomplexo de média móvel, obtendo resultados competitivos a outros métodos de descida já desenvolvidos na literatura. A metodologia proposta contribui com os algoritmos de filtragem adaptativa sob o aspecto de redução tanto do desajuste final como do custo computacional, apresentando elevada taxa de convergência, quando

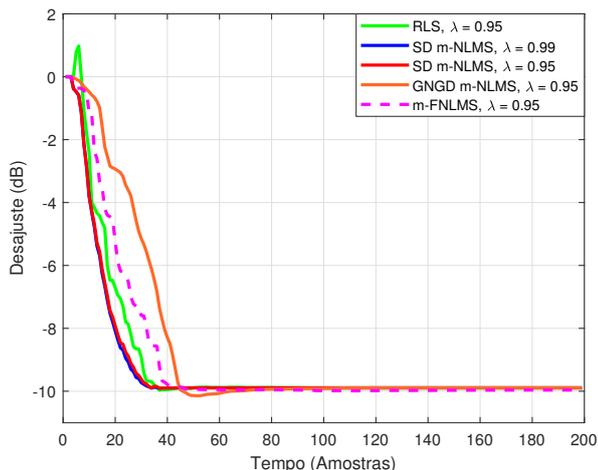


Figura 9. Comparação entre o desalinhamento dos algoritmos RLS, m -NLMS com diferentes esquemas de descida e m -FMNLS.

comparado à outras técnicas que envolvem cálculos mais complexos. Para fins de trabalhos futuros, a formulação da metodologia proposta utilizando sistemas de inferência *fuzzy* do tipo Takagi-Sugeno é de particular interesse.

AGRADECIMENTOS

Os autores agradecem à CAPES pelo apoio financeiro, e ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Maranhão (PPGEE-UFMA) pelo suporte ao desenvolvimento desta pesquisa.

REFERÊNCIAS

Abdygalievich, A.S., Barlybayev, A., and Amanzholovich, K.B. (2019). Quality evaluation fuzzy method of automated control systems on the LMS example. *IEEE Access*, 7, 138000–138010. doi:10.1109/access.2019.2943000.

Benhalima, S., Chandra, A., and Rezkallah, M. (2019). Real-time experimental implementation of an LMS-adaline-based ANFIS controller to drive PV interfacing power system. *IET Renewable Power Generation*, 13(7), 1142–1152. doi:10.1049/iet-rpg.2018.5245.

Chang, P.S. and Willson, A. (2000). Analysis of conjugate gradient algorithms for adaptive filtering. *IEEE Transactions on Signal Processing*, 48(2), 409–418. doi:10.1109/78.823968.

da Costa, P.U., Paim, G., Rocha, L.M.G., da Costa, E.A.C., de Almeida, S.J.M., and Bampi, S. (2021). Fixed-point NLMS and IPNLMS VLSI architectures for accurate FECG and FHR processing. *IEEE Transactions on Biomedical Circuits and Systems*, 15(5), 898–911. doi:10.1109/tbcas.2021.3120237.

Feng, S., Chen, C.L.P., Xu, L., and Liu, Z. (2021). On the accuracy–complexity tradeoff of fuzzy broad learning system. *IEEE Transactions on Fuzzy Systems*, 29(10), 2963–2974. doi:10.1109/tfuzz.2020.3009757.

Hata, R., Akhand, M.A.H., Islam, M.M., and Murase, K. (2018). Simplified real-, complex-, and quaternion-valued neuro-fuzzy learning algorithms. *International Journal of Intelligent Systems and Applications*, 10(5), 1–13. doi:10.5815/ijisa.2018.05.01.

Hata, R., Islam, M.M., and Murase, K. (2013). Quaternion neuro-fuzzy learning algorithm for fuzzy rule generation. In *2013 Second International Conference on Robot, Vision and Signal Processing*. IEEE. doi:10.1109/rvsp.2013.22.

Haykin, S. (2014). *Adaptive filter theory*. Pearson, Upper Saddle River, New Jersey.

Huang, W., Li, L., Li, Q., and Yao, X. (2018). Diffusion robust variable step-size LMS algorithm over distributed networks. *IEEE Access*, 6, 47511–47520. doi:10.1109/access.2018.2866857.

J. Lee, J.W.C. and Huang, H.C. (2009). Performance comparison of variable step-size nlms algorithms. *Proceedings of the World Congress on Engineering and Computer Science Vol I*.

Jalal, B., Yang, X., Liu, Q., Long, T., and Sarkar, T.K. (2020). Fast and robust variable-step-size LMS algorithm for adaptive beamforming. *IEEE Antennas and Wireless Propagation Letters*, 19(7), 1206–1210. doi:10.1109/lawp.2020.2995244.

Lan, L.T.H., Tuan, T.M., Ngan, T.T., Son, L.H., Giang, N.L., Ngoc, V.T.N., and Hai, P.V. (2020). A new complex fuzzy inference system with fuzzy knowledge graph and extensions in decision making. *IEEE Access*, 8, 164899–164921. doi:10.1109/access.2020.3021097.

Lunglmayr, M., Paireder, T., and Huemer, M. (2020). Null space projection enhanced LMS filters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(12), 3507–3511. doi:10.1109/tcsii.2020.2979034.

Mandic, D. (2004). A generalized normalized gradient descent algorithm. *IEEE Signal Processing Letters*, 11(2), 115–118. doi:10.1109/lsp.2003.821649.

Moura, R.P.A., Bergamaschi, F.B., Santiago, R.H.N., and Bedregal, B.R.C. (2013). Fuzzy quaternion numbers. In *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. doi:10.1109/fuzz-ieee.2013.6622400.

Navarro-Moreno, J. (2008). ARMA prediction of widely linear systems by using the innovations algorithm. *IEEE Transactions on Signal Processing*, 56(7), 3061–3068. doi:10.1109/tsp.2008.919396.

Sayed, A.H. (2003). *Fundamentals of Adaptive Filtering*. Wiley-IEEE Press.

Serra, G.L.O. (ed.) (2012). *Frontiers in Advanced Control Systems*. Intech.

Tian, Y., Han, X., Yin, J., and Li, Y. (2018). Adaption penalized complex LMS for sparse under-ice acoustic channel estimations. *IEEE Access*, 6, 63214–63222. doi:10.1109/access.2018.2875693.

Varidhithai, T. and Mandic, D.P. (2017). On an RLS-like LMS adaptive filter. In *2017 22nd International Conference on Digital Signal Processing (DSP)*. IEEE. doi:10.1109/icdsp.2017.8096130.

Zhang, S., Zhang, J., and So, H.C. (2020). Low-complexity decorrelation NLMS algorithms: Performance analysis and AEC application. *IEEE Transactions on Signal Processing*, 68, 6621–6632. doi:10.1109/tsp.2020.3039595.