

Predição de Séries Temporais Caóticas baseada em Modelo ARMA via Algoritmo Adaptativo FVSS-NLMS

Rodrigo Possidônio Noronha

Departamento de Engenharia Elétrica

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão

Imperatriz, Brasil

rodrigo.noronha@ifma.edu.br

Resumo—Neste trabalho é utilizado o modelo ARMA (do inglês, *AutoRegressive Moving Average*) para a predição de séries temporais caóticas. A estimativa do vetor de pesos do modelo ARMA pode ser obtida através de um algoritmo de otimização baseado em gradiente descendente estocástico, de tal forma que o desempenho de predição de uma série temporal caótica realizada pelo modelo ARMA é influenciado pelo valor do tamanho de passo. Para isto, neste trabalho é proposta uma nova versão do algoritmo NLMS (do inglês, *Normalized Least Mean Square*) com o tamanho de passo adaptado por um MFIS (do inglês, *Mamdani Fuzzy Inference System*) para atualização da estimativa do vetor de pesos do modelo ARMA. O desempenho do modelo ARMA para a predição de séries temporais caóticas foi comparado com os resultados obtidos pelas versões tradicionais dos algoritmos LMS (do inglês, *Least Mean Square*) e NLMS.

Index Terms—Modelo ARMA, NLMS, Séries Temporais, Sistemas Caóticos, Sistemas Fuzzy, Tamanho de Passo.

I. INTRODUÇÃO

Uma série temporal pode ser definida como um conjunto de amostras ordenadas no tempo e obtidas em instantes de tempo determinados [1]. Muitas aplicações do mundo real estão inseridas no contexto de séries temporais. Uma aplicação bastante realizada e de grande interesse para a comunidade acadêmica e em geral é a predição de séries temporais, que é utilizada em diversas áreas do conhecimento como em sistemas biológicos [2], em econometria [3], na teoria de sistemas de controle [4], meteorologia [5], entre outras. A natureza caótica de algumas séries temporais é uma complexidade que motiva o desenvolvimento de modelos matemáticos que sejam aptos a representar a dinâmica não linear [6]. As principais características que definem uma série temporal como caótica é que seu comportamento deve ter uma natureza estacionária e ser sensível às variações das condições iniciais, de tal forma que uma pequena variação em suas condições iniciais faz com que os valores observados a cada instante de tempo evoluam dinamicamente de forma drástica.

O modelo ARMA (do inglês, *AutoRegressive Moving Average*) é um modelo matemático utilizado para predição de séries temporais de natureza estacionária [7] [8]. A ideia principal de realizar a predição de séries temporais baseado em modelo ARMA é inicialmente obter um modelo de trei-

namento para um conjunto de dados previamente coletados e, depois prever t passos a frente as próximas observações temporais baseada no modelo de treinamento obtido. O vetor de pesos de um modelo ARMA pode ser obtido, por exemplo, através de algoritmos adaptativos baseados em gradiente descendente estocástico. Para estimação do vetor de pesos de um modelo ARMA alguns algoritmos adaptativos baseados em gradiente descendente estocástico têm sido utilizados, podendo-se citar os mais tradicionais como os algoritmos LMS (do inglês, *Least Mean Square*) [9] e o NLMS (do inglês, *Normalized Least Mean Square*) [10]. O presente trabalho é delimitado a estudar o algoritmo NLMS para a estimação do vetor de pesos de um modelo ARMA, uma vez que ele apresenta uma menor sensibilidade às variações da potência do sinal de entrada e tem um bom desempenho em sinais correlacionados [11]. Em [12], [13] foi utilizado um modelo ARMA com o vetor de pesos estimado pelo algoritmo NLMS com o tamanho de passo fixo para predição de potência reativa necessária à operação de forno elétrico a arco.

O desempenho do algoritmo NLMS é influenciado pelo tamanho de passo, de tal maneira que a sua estabilidade e a convergência do vetor de pesos depende do valor de tamanho de passo. Se o tamanho de passo é grande a velocidade de convergência do algoritmo NLMS será rápida, mas o MSE (do inglês, *Mean Squared Error*) no regime estacionário será grande. Se o tamanho de passo é pequeno a velocidade de convergência do algoritmo NLMS será lenta, mas o MSE no regime estacionário será pequeno [14]. A influência do tamanho de passo no desempenho do algoritmo NLMS independe da natureza do problema analisado [15]. Como alternativa para solucionar esse problema, um bom desempenho do algoritmo NLMS pode ser obtido tornando o tamanho de passo variável [16]. Diversas metodologias têm apresentado propostas para tornar o tamanho de passo variável, como exemplo em [17]–[19]. Contudo, essas metodologias e outras publicadas na literatura geralmente necessitam de medidas estatísticas (como exemplo as funções de correlação, covariância, entre outros) que pode tornar inviável a aplicação do algoritmo NLMS em um contexto de operação em tempo real. Além disso, o ajuste do tamanho de passo variável é

dependente da natureza do problema e, conseqüentemente, há o incremento da complexidade de análise matemática do algoritmo adaptativo.

Este artigo tem como objetivo propor a predição de séries temporais caóticas baseado em modelo ARMA, com a atualização da estimativa do vetor de pesos realizada através do algoritmo NLMS com o tamanho de passo variável adaptado por um MFIS (do inglês, *Mamdani Fuzzy Inference System*). A justificativa para adaptação do tamanho de passo variável através de um MFIS é que é possível, através de uma descrição linguística implementada em uma base regras *fuzzy*, solucionar problemas complexos de difícil formulação matemática. Essa nova versão do algoritmo NLMS é intitulada FVSS-NLMS (*Fuzzy Variable Step Size - Normalized Least Mean Square*).

De acordo com o levantamento bibliográfico realizado pelo autor, na literatura é possível encontrar poucas metodologias que propõem a utilização de um MFIS com o tamanho de passo variável adaptado por um MFIS. Em [20] foi utilizado um MFIS para o ajuste do tamanho de passo variável para aplicação na equalização adaptativa de canais. Nessa metodologia o tamanho de passo é ajustado somente em função do erro quadrático obtido a cada instante de tempo, não sendo levado em consideração nenhuma informação sobre a quantidade de instantes de tempo necessários para a convergência do vetor de pesos de um determinado modelo. Uma vez que se espera que ao fim dos instantes de tempo ocorra a convergência do vetor de pesos de um modelo, a quantidade de instantes de tempo é um parâmetro importante para a otimização do funcional de custo descrito pelo gradiente estocástico. No algoritmo proposto, o tamanho de passo variável é adaptado em função do erro quadrático e do instante de tempo normalizado pelo método Min-Max. Através de uma descrição linguística implementada em uma base de regras *fuzzy*, é possível obter uma bom desempenho do algoritmo na etapa de treinamento do modelo ARMA e, conseqüentemente, na etapa de predição, quando comparado ao uso do tamanho de passo fixo. Com o objetivo de comparação de desempenho do algoritmo FVSS-NLMS, os resultados obtidos durante a etapa de treinamento e de predição foram comparados com as versões tradicionais dos algoritmos LMS e NLMS com o tamanho de passo fixo. Esse artigo é organizado como segue: na Seção II, é apresentada o modelo ARMA; na Seção III, é apresentado o algoritmo FVSS-NLMS; na Seção IV é apresentado o procedimento para predição de séries temporais baseado em modelo ARMA via algoritmo FVSS-NLM; na Seção V são apresentados os resultados computacionais obtidos.

II. MODELO ARMA

Box e Jenkins [21] descreveram que um modelo autorregressivo com média móvel pode representar um processo estocástico estacionário, em que a amostra $y(k)$ observada no instante k não depende apenas das amostras observadas nos instantes passados $y(k-1), y(k-2), \dots, y(k-n)$, mas também possui uma relação interdependente com os ruídos observados $\epsilon(k-1), \epsilon(k-2), \dots, \epsilon(k-n)$. O modelo autoregressivo com média móvel é conhecido como modelo ARMA. Após a

parametrização do modelo ARMA baseado em um conjunto de dados de treinamento, então pode-se obter as estimativas futuras.

Para uma série temporal estacionária representada por um conjunto de amostras, o modelo ARMA(p, q) pode ser expressado da seguinte forma:

$$y(k) = \psi_1 y(k-1) + \psi_2 y(k-2) + \dots + \psi_p y(k-p) + \epsilon(k) - \vartheta_1 \epsilon(k-1) - \vartheta_2 \epsilon(k-2) - \dots - \vartheta_q \epsilon(k-q), \quad (1)$$

em que a ordem p indica que há p versões atrasadas de $y(k)$ e a ordem q indica que há q versões atrasadas de $\epsilon(k)$. Além disso, $\psi(k)$ e $\vartheta(k)$ são os coeficientes ou pesos do modelo ARMA(p, q) que satisfazem as condições de estacionariedade e invertibilidade, e $\epsilon(k)$ é um ruído descrito por um processo estocástico com média zero e variância σ_ϵ^2 constante. Uma outra maneira de reescrever (1) é dada na forma vetorial, conforme a seguir:

$$y(k) = \Theta^T(k)\Gamma(k) = \Gamma^T(k)\Theta(k), \quad (2)$$

em que $\Gamma(k) = [y(k-1) \ y(k-2) \ \dots \ y(k-p) \ -\epsilon(k) \ -\epsilon(k-1) \ -\epsilon(k-2) \ \dots \ -\epsilon(k-p)] \in \mathbb{R}^{(p+q+1) \times 1}$ é o vetor de regressores e $\Theta(k) = [\psi_1 \ \dots \ \psi_p \ 1 \ \vartheta_1 \ \dots \ \vartheta_q] \in \mathbb{R}^{(p+q+1) \times 1}$ é o vetor de pesos do modelo ARMA(p, q). De acordo com o critério de Wiener, a atualização da estimativa do vetor de pesos do modelo ARMA(p, q) pode ser obtida da seguinte forma:

$$\Theta(k+1) = \Theta(k) - \frac{1}{2}\mu \nabla_{\Theta(k)}(E[e^2(k)]), \quad (3)$$

em que $J = \nabla_{\Theta(k)}(E[e^2(k)])$ é o funcional de custo descrito pelo gradiente estocástico do erro quadrático $e^2(k) = (d(k) - y(k))^2$ e μ é o tamanho de passo de atualização da estimativa do vetor de pesos. Para aplicação em tempo real, é bastante dispendioso a acumulação de amostras para trabalhar com a esperança matemática $E[\bullet]$ do erro quadrático. No lugar de trabalhar com a esperança matemática $E[\bullet]$ do erro quadrático, pode-se trabalhar com valores instantâneos do erro quadrático, surgindo assim o algoritmo adaptativo LMS. Dessa forma, o funcional de custo J é reescrito como:

$$J = \nabla_{\Theta(k)}(e^2(k)), \quad (4)$$

em que substituindo (4) em (3), obtém-se que:

$$\begin{aligned} \Theta(k+1) &= \Theta(k) - \frac{1}{2}\mu \nabla_{\Theta(k)}(e^2(k)) \\ &= \Theta(k) - \frac{1}{2}\mu \nabla_{\Theta(k)}[(d(k) - \Theta^T(k)\Gamma(k))^2] \\ &= \Theta(k) + \mu e(k)\Gamma(k), \end{aligned} \quad (5)$$

em que $\nabla_{\Theta(k)}(e^2(k)) = -2e(k)\Gamma(k)$. Para obtenção do algoritmo NLMS é necessário normalizar o algoritmo LMS pela potência do sinal de entrada, conforme a seguir:

$$\Theta(k+1) = \Theta(k) + \mu \frac{e(k)\Gamma(k)}{\Gamma^T(k)\Gamma(k)} \quad (6)$$

III. ALGORITMO ADAPTATIVO FVSS-NLMS

Nesta seção é proposta uma nova versão do algoritmo NLMS com adaptação *fuzzy* do tamanho de passo variável. Na Seção II foram desenvolvidas as equações de diferenças para a obtenção do modelo ARMA(p, q). Para que o modelo ARMA(p, q) possa rastrear a dinâmica do conjunto de dados de treinamento de uma série temporal, é necessário que a estimativa do vetor de pesos $\Theta(k)$ seja atualizada a cada instante de tempo. Nesse trabalho, a atualização da estimativa do vetor de pesos $\Theta(k)$ é realizada pelo algoritmo FVSS-NLMS, como segue:

$$\begin{aligned} \mu(k) &= \text{MFIS}(e^2(k), \mathcal{K}(k)) \\ \Theta(k+1) &= \begin{cases} \Theta(k) + \mu(k) \frac{e(k)\Gamma(k)}{h(k)}, & \text{se } h(k) \neq 0 \\ \Theta(k), & \text{se } h(k) = 0 \end{cases} \\ k &\in [1, K], \end{aligned} \quad (7)$$

em que $\mu(k)$ é o tamanho de passo adaptado pelo MFIS, $h(k) = \Gamma^T(k)\Gamma(k)$, $\Theta(k)$ é o vetor de pesos do modelo ARMA(p, q), $e^2(k)$ é o erro quadrático, $\Gamma(k)$ é o vetor de regressores, $\mathcal{K}(k)$ é o instante de tempo normalizado pelo método Min-Max (8) e K é o número total de instantes de tempo. Conforme pode ser notado em (7), o MFIS é composto por duas variáveis de entrada, que são $e^2(k)$ e $\mathcal{K}(k)$.

$$\mathcal{K}(k) = \frac{k-1}{K-1} \quad (8)$$

Definição 1 (Função de Pertinência). *Seja U um universo de discurso e $x \in U$. Um conjunto fuzzy F definido em U é caracterizado por uma função:*

$$m_F(x) : U \rightarrow [0, 1] \quad (9)$$

que é chamada de função de pertinência ou MBF (do inglês, Membership Function) do conjunto fuzzy F .

Definição 2 (Variável Linguística). *Seja uma variável linguística $x \in U$. Intuitivamente, uma variável linguística é definida como um substantivo que recebe adjetivos ou atributos que são representados por conjuntos fuzzy definidos em U .*

Definição 3 (Regra Fuzzy). *As regras fuzzy do tipo Se (proposições do antecedente) **então** (proposições do consequente), para um MFIS, podem ser definidas por especialistas e nelas está representado o conhecimento humano subjetivo. O antecedente e consequente são definidos por proposições fuzzy do tipo x é A , em que A é um valor linguístico representado pelo conjunto fuzzy \tilde{A} definido em U . Através do conectivo lógico “E”, as proposições fuzzy podem ser conectadas, em que “E” representa a interseção fuzzy caracterizada pela seguinte MBF:*

$$m(x, y) = t[m_{\tilde{A}}(x), m_{\tilde{B}}(y)] = \min[m_{\tilde{A}}(x), m_{\tilde{B}}(y)], \quad (10)$$

em que y é uma variável linguística definida no universo de discurso V e B é um valor linguístico representado pelo conjunto fuzzy \tilde{B} definido em V . A função descrita em (10) é

uma MBF da interseção fuzzy que é representada pela regra **Se** x é A **E** y é B **e** $t : [0, 1] \times [0, 1] \rightarrow [0, 1]$ é uma norma- t .

Através da fuzzificação, cada variável linguística do antecedente recebe cada valor linguístico com um certo grau de pertinência através dos seguintes mapeamentos $m_j(\mathcal{K}(k)) : U \rightarrow [0, 1]$ e $m_j(e^2(k)) : V \rightarrow [0, 1]$, com os universos de discurso $U = [0, 1]$ e $V = [1 \times 10^{-3}, 1, 3]$. Três MBFs do tipo triangular foram definidas para cada variável linguística do antecedente, com os valores linguísticos “Pequeno” (P) para $j = 1$, “Médio” (M) para $j = 2$ e “Grande” (G) para $j = 3$. Na Tabela I são apresentados os parâmetros que definem as três MBFs do tipo triangular para os valores linguísticos do antecedente.

A variável linguística do consequente $\lambda(k)$ recebe cada valor linguístico do consequente com um certo grau de pertinência pertencente ao intervalo $[0, 1]$ através do seguinte mapeamento $m_j(\lambda(k)) : Z \rightarrow [0, 1]$, com o universo de discurso $Z = [1 \times 10^{-1}, 2]$. Três MBFs do tipo triangular foram definidas para a variável linguística do consequente, com os valores linguísticos “Pequeno” (P) para $j = 1$, “Médio” (M) para $j = 2$ e “Grande” (G) para $j = 3$. Na Tabela I são apresentados os parâmetros que definem as três MBFs do tipo triangular para os valores linguísticos do consequente.

Tabela I: Intervalo das MBFs triangulares.

$\mathcal{K}(k)$		$e^2(k)$		$\lambda(k)$	
Intervalo		Intervalo		Intervalo	
P	[0, 0,2, 0,3]	P	[0,05, 0,35, 0,65]	P	[0,1, 0,5, 1]
M	[0,2, 0,3, 0,5]	M	[0,35, 0,65, 0,95]	M	[0,5, 1, 1,5]
G	[0,3, 0,5, 1,0]	G	[0,65, 0,95, 1,25]	G	[1, 1,5, 2]

Os parâmetros das MBFs foram definidas, por tentativa e erro, em função do conhecimento do especialista obtido através de experiências passadas durante as simulações do problema analisado, com o objetivo de um obter um bom desempenho na atualização da estimativa do vetor de pesos do modelo ARMA. A base de regras desenvolvida para adaptação do tamanho de passo $\mu(k)$ pode ser vista em (15). Através da implicação *fuzzy* é possível modelar a relação *fuzzy* entre os universos de discursos do antecedente e do consequente. A entrada da implicação *fuzzy* é o grau de ativação da i -ésima regra dado pela norma norma- t :

$$\alpha^i = m(\mathcal{K}(k), e^2(k)) = \min[m_j(\mathcal{K}(k)), m_j(e^2(k))] \quad (11)$$

e a saída da implicação *fuzzy* é uma MBF que é obtida através do método do corte- α^i , dado por:

$$m_{\mathcal{R}^i} = \min[\alpha^i, m_j(\lambda(k))] \quad (12)$$

A implicação *fuzzy* é realizada para cada regra, então uma MBF $m_{\mathcal{R}^i}$ é obtida para cada regra *fuzzy*. Com o objetivo de obter uma única MBF que represente a resposta total do MFIS, é realizada a agregação *fuzzy*:

$$m_{Total} = \max[m_{\mathcal{R}^1}, m_{\mathcal{R}^2}, \dots, m_{\mathcal{R}^g}] \quad (13)$$

Após isso, é realizada a defuzzificação de m_{Total} através do método do centroide, conforme a seguir:

$$\mu(k) = \frac{\sum_{i=1}^9 \lambda^i(k) m_{Total}(\lambda^i(k))}{\sum_{i=1}^9 m_{Total}(\lambda^i(k))} \quad (14)$$

- \mathcal{R}^1 : Se $\mathcal{K}(k)$ é P E $e^2(k)$ é P então $\lambda(k)$ é M
 \mathcal{R}^2 : Se $\mathcal{K}(k)$ é P E $e^2(k)$ é M então $\lambda(k)$ é M
 \mathcal{R}^3 : Se $\mathcal{K}(k)$ é P E $e^2(k)$ é G então $\lambda(k)$ é M
 \mathcal{R}^4 : Se $\mathcal{K}(k)$ é M E $e^2(k)$ é P então $\lambda(k)$ é P
 \mathcal{R}^5 : Se $\mathcal{K}(k)$ é M E $e^2(k)$ é M então $\lambda(k)$ é P
 \mathcal{R}^6 : Se $\mathcal{K}(k)$ é M E $e^2(k)$ é G então $\lambda(k)$ é G
 \mathcal{R}^7 : Se $\mathcal{K}(k)$ é G E $e^2(k)$ é P então $\lambda(k)$ é P
 \mathcal{R}^8 : Se $\mathcal{K}(k)$ é G E $e^2(k)$ é M então $\lambda(k)$ é M
 \mathcal{R}^9 : Se $\mathcal{K}(k)$ é G E $e^2(k)$ é G então $\lambda(k)$ é G

IV. PROCEDIMENTO PARA PREDIÇÃO BASEADO EM MODELO ARMA VIA ALGORITMO ADAPTATIVO FVSS-NLMS

A seguir são apresentados os procedimentos para obtenção do modelo ARMA via algoritmo adaptativo FVSS-NLMS e realização da predição:

- **Passo 1:** Pré processamento: O modelo ARMA foi projetado para aplicação em uma sequência de amostras estacionárias. Caso a série temporal não seja estacionária, é necessário torná-la estacionária.
- **Passo 2:** Definir a ordem do modelo: é necessário definir o parâmetros p e q que definem os máximos atrasos dos termos autoregressivos e de média móvel do modelo ARMA(p, q). A definição pode ser realizada por tentativa e erro ou baseado em algum critério de otimalidade.
- **Passo 3:** Parametrizar o modelo ARMA(p, q): a cada amostra de um conjunto de dados de treinamento é realizada a atualização da estimativa do vetor de pesos $\Theta(k)$, de acordo com (7). É importante notar que a cada amostra do conjunto de dados de treinamento é realizada a adaptação do tamanho de passo através do algoritmo FVSS-NLMS.
- **Passo 4:** Avaliar o modelo obtido: após obtido o modelo ARMA(p, q) para o conjunto de dados de treinamento, é necessário avaliá-lo através de métricas estatísticas. Caso o desempenho do modelo seja insatisfatório, é necessário voltar para a **Passo 2** e realizar novamente a parametrização do modelo ARMA(p, q).
- **Passo 5:** Realizar a predição: é utilizado o modelo obtido no **Passo 3** para realizar a predição t passos a frente com base nas amostras observadas nos instantes anteriores.

V. RESULTADOS COMPUTACIONAIS

Nesta seção são apresentados os resultados computacionais obtidos através da implementação da metodologia proposta. A obtenção de resultados é dividida em duas etapas, a primeira etapa é referente à etapa de treinamento e a segunda etapa é referente à etapa de predição baseado no modelo ARMA(p, q) obtido na etapa de treinamento. O desempenho do modelo

ARMA(p, q) obtido para as duas etapas foi avaliado através das seguinte métricas:

- VAF (do inglês, *Variance Accounted For*):

$$\text{VAF}(\%) = \left[1 - \frac{\text{var}(\mathbf{y} - \hat{\mathbf{y}})}{\text{var}(\mathbf{y})} \right] \times 100, \quad (16)$$

em que $\text{var}(\bullet)$ é a variância, \mathbf{y} é o vetor de dados de uma série temporal, $\hat{\mathbf{y}}$ é o vetor de dados estimado pelo modelo ARMA(p, q).

- MSE (do inglês, *Mean Squared Error*):

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2, \quad (17)$$

em que N é a quantidade de amostras do vetor de dados.

- NRMSE (do inglês, *Normalized Root Mean Squared Error*):

$$\text{NRMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^N \left[\frac{(y(k) - \hat{y}(k))^2}{\max(\mathbf{y}) - \min(\mathbf{y})} \right]} \quad (18)$$

- NDEI (do inglês, *Non-Dimensional Error Index*):

$$\text{NDEI} = \frac{\text{RMSE}}{\text{std}(\mathbf{y})}, \quad (19)$$

em que $\text{std}(\bullet)$ é o desvio padrão.

- FIT (do inglês, *Best Fit Criterera*):

$$\text{FIT}(\%) = \left(1 - \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|}{\|\mathbf{y} - \bar{\mathbf{y}}\|} \right) \times 100, \quad (20)$$

em que $\bar{\mathbf{y}}$ é o valor médio de \mathbf{y} e $\|\bullet\|$ é o operador norma Euclidiana.

Nesse trabalho a série temporal caótica de Lorenz foi escolhida para avaliar a capacidade de predição realizada pelo modelo ARMA(p, q), com o vetor de pesos estimado pelo algoritmo FVSS-NLMS. A série temporal de Lorenz é obtida através de um conjunto de três equações diferenciais não lineares de convecção térmica na baixa atmosfera, dadas a seguir:

$$\begin{aligned} \dot{x}(t) &= \alpha[y(t) - x(t)] \\ \dot{y}(t) &= \beta x(t) - y(t) - x(t)z(t) \\ \dot{z}(t) &= x(t)y(t) - \gamma z(t), \end{aligned} \quad (21)$$

em que $x(t)$ é proporcional à velocidade de fluxo do fluido, $y(t)$ caracteriza a diferença de temperatura entre os elementos ascendente e descendente do fluido, e $z(t)$ é proporcional aos desvios de temperatura vertical do valor de equilíbrio. Associado ao número de Prandtl, o parâmetro α relaciona viscosidade e condutividade térmica, β está relacionado com o gradiente de temperatura e γ é um fator geométrico. Os valores dos parâmetros foram estabelecidos iguais a $\alpha = 18$, $\beta = 28$ e $\gamma = 8/3$. Nesse trabalho, o conjunto de equações diferenciais não lineares em (21) é resolvido pelo método Runge-Kutta de quarta ordem com o passo temporal definido igual a $\Delta t = 0,01$.

Com o objetivo de comparar os resultados obtidos através do algoritmo FVSS-NLMS com os algoritmos LMS e NLMS

Tabela II: Resultados das métricas estatísticas obtidas para a etapa de treinamento.

Métricas	$x(k)$			$y(k)$			$z(k)$		
	LMS	NLMS	FVSS-NLMS	LMS	NLMS	FVSS-NLMS	LMS	NLMS	FVSS-NLMS
VAF(%)	93,9361	95,8098	97,7622	93,9975	95,8429	97,9290	94,4875	96,8104	98,4114
MSE	0,2393	0,2097	0,0520	0,2405	0,2109	0,0531	0,2291	0,2059	0,0351
NRMSE	0,0947	0,0760	0,0147	0,0977	0,0775	0,0168	0,0796	0,0544	0,0098
NDEI	0,0833	0,0702	0,0117	0,0858	0,0723	0,0128	0,0801	0,0567	0,0104
FIT(%)	94,2332	95,9792	98,6172	94,6976	96,0027	98,6304	94,9960	96,6448	98,8427

Tabela III: Resultados das métricas estatísticas obtidas para a etapa de predição.

Métricas	$x(k)$			$y(k)$			$z(k)$		
	LMS	NLMS	FVSS-NLMS	LMS	NLMS	FVSS-NLMS	LMS	NLMS	FVSS-NLMS
VAF(%)	94,2020	96,2011	97,9612	94,1223	96,0019	98,1251	94,7812	97,0104	98,8125
MSE	0,2351	0,2041	0,0501	0,2376	0,2078	0,0502	0,2261	0,2019	0,0311
NRMSE	0,0917	0,0721	0,0112	0,0913	0,0721	0,0126	0,0761	0,0511	0,0071
NDEI	0,0813	0,0672	0,0099	0,0815	0,0698	0,0101	0,0789	0,0547	0,0084
FIT(%)	94,9156	96,3415	98,9912	94,9812	96,4516	98,9914	95,4589	96,9814	99,1355

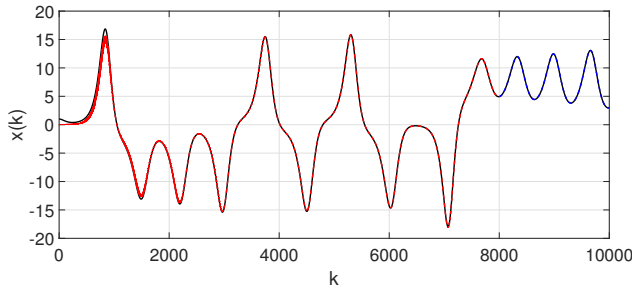


Figura 1: Estimação da série temporal de Lorenz $x(k)$ referente à etapa de treinamento (cor vermelha) e à etapa de predição (cor azul).

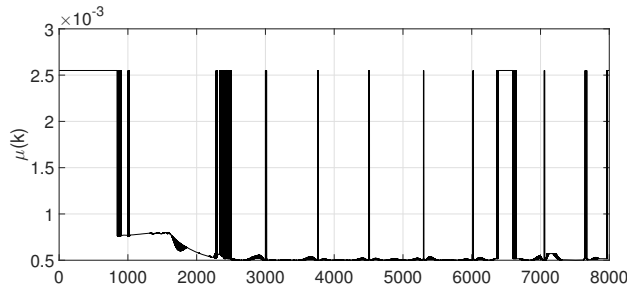


Figura 2: Tamanho de passo variável para a série temporal de Lorenz $x(k)$.

com o tamanho de passo fixo, o tamanho de passo para os algoritmos LMS e NLMS foi definido igual a $\mu = 0,003$. Para realização das etapas de treinamento e de predição, foram obtidas 10000 amostras da série temporal de Lorenz em (21), em que as 8000 primeiras amostras foram utilizadas para obtenção do modelo ARMA(p, q) de treinamento e as 2000 amostras restantes foram utilizadas para avaliar a capacidade de predição do modelo ARMA(p, q). A etapa de predição foi realizada para $t = 2$ passos a frente. Os parâmetros (p, q) foram definidos por tentativa e erro com $p = 2$ e $q = 0$. Nas Figuras 1, 3 e 5 são apresentados a estimação das séries temporais de Lorenz $x(k)$, $y(k)$ e $z(k)$ pelo modelo

ARMA(2,0) para as etapas de treinamento e de predição.

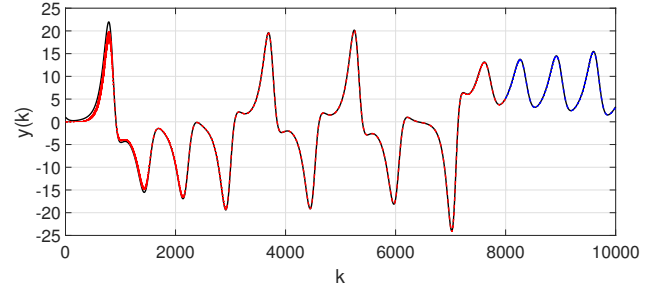


Figura 3: Estimação da série temporal de Lorenz $y(k)$ referente à etapa de treinamento (cor vermelha) e à etapa de predição (cor azul).

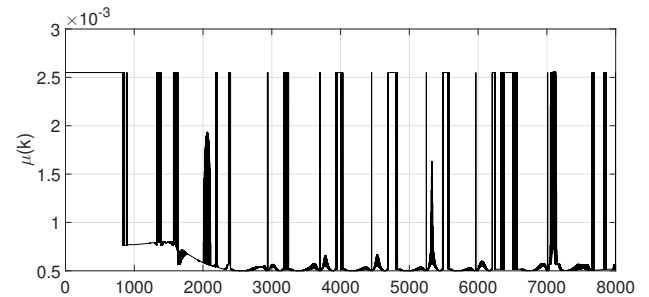


Figura 4: Tamanho de passo variável para a série temporal de Lorenz $y(k)$.

A comparação de resultados obtidos através das métricas estatísticas para as etapas de treinamento e de predição referentes às séries temporais de Lorenz $x(k)$, $y(k)$ e $z(k)$ podem ser vistas nas Tabelas II e III. Com relação às métricas estatísticas VAF(%), MSE, NRMSE, NDEI e FIT(%), é possível notar que o modelo ARMA(2,0) com o vetor de pesos estimado pelo algoritmo FVSS-NLMS obteve os melhores resultados, tanto para a etapa de treinamento como para a etapa de predição. As evoluções temporais dos tamanhos de passos variáveis são apresentadas nas Figuras 2, 4 e 6. Os melhores resultados

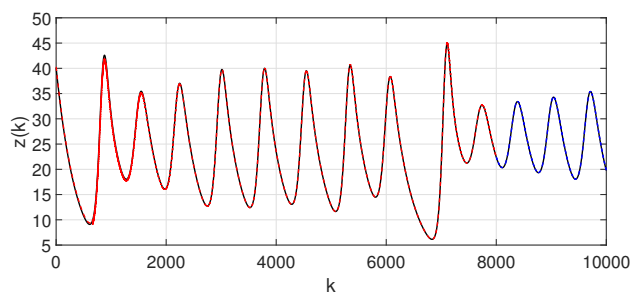


Figura 5: Estimação da série temporal de Lorenz $z(k)$ referente à etapa de treinamento (cor vermelha) e à etapa de predição (cor azul).

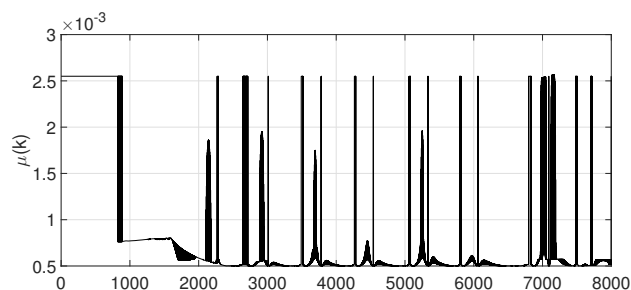


Figura 6: Tamanho de passo variável para a série temporal de Lorenz $z(k)$.

com relação às métricas estatísticas são devidos ao tamanho de passo variável do algoritmo NLMS ser adaptado através de uma descrição linguística baseada no conhecimento do especialista implementada em uma base de regras *fuzzy*, de tal forma que a adaptação é realizada em função do erro quadrático de estimação e do avanço temporal do processo de otimização realizado pelo algoritmo FVSS-NLMS.

VI. CONCLUSÃO

Foi observado que o modelo ARMA com o vetor de pesos estimado pelo algoritmo FVSS-NLMS, tanto para a etapa de treinamento como para a etapa de predição, obteve os melhores resultados com relação às métricas estatísticas utilizadas, quando comparado aos algoritmos LMS e NLMS com o tamanho de passo fixo. Dessa forma, nota-se que a descrição linguística baseada no conhecimento do especialista implementada em uma base de regras permitiu uma satisfatória capacidade de adaptação do tamanho de passo e, consequentemente, uma satisfatória rastreabilidade de uma dinâmica caótica apresentada pela série temporal de Lorenz. Por fim, nota-se que a adaptação do tamanho de passo pelo MFIS independe de medidas estatísticas de alta ordem.

REFERÊNCIAS

[1] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.
 [2] A. Gharebaghi and M. Lindén, "A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 9, pp. 4102–4115, 2017.

[3] N. Nonejad, "An overview of dynamic model averaging techniques in time-series econometrics," *Journal of Economic Surveys*, vol. 35, no. 2, pp. 566–614, 2021.
 [4] J. Lee, P. Zhang, L. K. Gan, D. A. Howey, M. A. Osborne, A. Tosi, and S. Duncan, "Optimal operation of an energy management system using model predictive control and gaussian process time-series modeling," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 6, no. 4, pp. 1783–1795, 2018.
 [5] M. Lorenz and M. Brunke, "Trends of nutrients and metals in precipitation in northern germany: the role of emissions and meteorology," *Environmental Monitoring and Assessment*, vol. 193, no. 6, pp. 1–20, 2021.
 [6] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Physical review letters*, vol. 59, no. 8, p. 845, 1987.
 [7] C. Kocak, "Arma (p, q) type high order fuzzy time series forecast method based on fuzzy logic relations," *Applied Soft Computing*, vol. 58, pp. 92–103, 2017.
 [8] K. Ansari, K.-D. Park, and N. Kubo, "Linear time-series modeling of the gnss based tec variations over southwest japan during 2011–2018 and comparison against arma and gim models," *Acta Astronautica*, vol. 165, pp. 248–258, 2019.
 [9] S. Rahman, M. M. Rashid, and M. Z. Alam, "A unified analysis of proposed wavelet transform domain lms-algorithm for arma process," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE, 2019, pp. 195–200.
 [10] R. G. Garroppo and C. Callegari, "Prediction of mobile networks traffic: enhancement of the nmls technique," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2020, pp. 1–6.
 [11] A. Pandey, L. Malviya, and V. Sharma, "Comparative study of lms and nmls algorithms in adaptive equalizer," *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, no. 3, pp. 1584–1587, 2012.
 [12] M. H. Golshan and H. Samet, "Updating stochastic model coefficients for prediction of arc furnace reactive power," *Electric Power Systems Research*, vol. 79, no. 7, pp. 1114–1120, 2009.
 [13] H. Samet, A. Mojallal, and T. Ghanbari, "Employing grey system model for prediction of electric arc furnace reactive power to improve compensator performance," *Przeglad Elektrotechniczny*, vol. 89, no. 12, pp. 110–115, 2013.
 [14] M. S. Aslam, P. Shi, and C.-C. Lim, "Self-adapting variable step size strategies for active noise control systems with acoustic feedback," *Automatica*, vol. 123, p. 109354, 2021.
 [15] H.-C. Shin, A. H. Sayed, and W.-J. Song, "Variable step-size nmls and affine projection algorithms," *IEEE signal processing letters*, vol. 11, no. 2, pp. 132–135, 2004.
 [16] T. Strutz, "Estimation of measurement-noise variance for variable-step-size nmls filters," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
 [17] Y. Sun, M. Wang, Y. Han, and C. Zhang, "An improved vss nmls algorithm for active noise cancellation," in *AIP Conference Proceedings*, vol. 1864, no. 1. AIP Publishing LLC, 2017, p. 020158.
 [18] F. Casco-Sanchez, M. Lopez-Guerrero, S. Javier-Alvarez, and R. C. Medina-Ramirez, "A variable-step size nmls algorithm based on the cross-correlation between the squared output error and the near-end input signal," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 8, pp. 1197–1202, 2019.
 [19] X. Wang, S. Li, J. Ye, X. Fan, M. Qin, X. Jia, G. Wang, and Q. Liu, "Research on digital predistortion technology based on improved nmls algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, p. e22661, 2021.
 [20] Y. Ng, H. Mohamad, and T. Chuah, "Block-based fuzzy step size nmls algorithms for subband adaptive channel equalisation," *IET Signal Processing*, vol. 3, no. 1, pp. 23–32, 2009.
 [21] S. Makridakis and M. Hibon, "Arma models and the box-jenkins methodology," *Journal of forecasting*, vol. 16, no. 3, pp. 147–163, 1997.