

METODOLOGIA DE CONTROLE PREDITIVO BASEADO EM MODELO NEBULOSO T-S EVOLUTIVO

ARNALDO PINHEIRO DE AZEVEDO JÚNIOR*, GINALBER LUIZ DE OLIVEIRA SERRA†

**Av. dos Portugueses, s/n, Bacanga, CEP: 65001-970
Universidade Federal do Maranhão
São Luís, Maranhão, Brasil*

†*Av. Getúlio Vargas, 04, Monte Castelo, CEP: 65030-005
Instituto Federal do Maranhão
São Luís, Maranhão, Brasil*

Emails: arnaldoe@ifma@gmail.com, ginalber@ifma.edu.br

Abstract— This paper proposes a evolving T-S fuzzy model based predictive control methodology for nonlinear uncertain SISO processes. The addition, merging, and exclusion mechanism of rules are used. The control signal is obtained based on the PNMPC algorithm in the presence of constraints on the variables of input, control effort and output, respectively. To evaluate the proposed methodology, a case study with the surge tank is carried out in the following scenarios: reference tracking, disturbance rejection and robustness to parametric uncertainties.

Keywords— Predictive control, evolving T-S fuzzy model, PNMPC, constraints.

Resumo— Neste artigo é proposta uma metodologia de controle preditivo baseado em modelo nebuloso T-S evolutivo para processos SISO com dinâmica não linear e parâmetros incertos. São utilizados os mecanismos de adição, fusão e exclusão de regras. O sinal de controle é obtido com base no algoritmo PNMPC na presença de restrições nas variáveis de entrada, incremento de controle e saída, respectivamente. Para avaliar a metodologia proposta é realizado um estudo de caso com o *surge tank* nos seguintes cenários: seguimento de referência, rejeição de distúrbio e robustez às incertezas paramétricas.

Palavras-chave— Controle preditivo, modelo nebuloso T-S evolutivo, PNMPC, restrições.

1 Introdução

Técnicas de identificação de sistemas para Controle Preditivo Baseado em Modelo (CPBM) desempenham um papel importante na performance do controlador. A principal justificativa disto é que supõe-se necessário um modelo preciso do processo a ser controlado. Tal suposição pode originar problemas, visto que muitos processos complexos são difíceis de serem modelados matematicamente, pois apresentam grandes incertezas e fortes não linearidades. Por esta razão, várias técnicas de modelagem não linear são utilizadas no contexto CPBM (Rastegar et al., 2016; Boulkaibet et al., 2017; Eski and Temürtenk, 2013).

Nesse interim, a modelagem nebulosa, em particular a estrutura *Takagi-Sugeno* (T-S), tornou-se um ativo campo de pesquisa graças a sua capacidade de lidar com incertezas, bem como na descrição de processos com dinâmica não linear (Lughofer, 2011). No entanto, quando aplicada de modo *off-line*, pode apresentar um desempenho insatisfatório principalmente quando a dinâmica do processo é variante no tempo. Neste caso, uma estratégia de aprendizagem *online* é mais indicada, o que corresponde a um mecanismo de evolução da estrutura do modelo, bem como de seus parâmetros (Angelov, 2012). Esta última, chamada de modelagem nebulosa evolutiva, resultou no desenvolvimento de diversos algoritmos em anos recentes (Angelov and Filev, 2004; Lughofer and Klement, 2005; Kasabov

and Song, 2002; Dovžan and Škrjanc, 2011).

Em Rastegar et al. (2016) foi proposta uma metodologia de modelagem nebulosa evolutiva baseado no *New Unsupervised Fuzzy Algorithm* (NUFCA) aplicado em um esquema de controle preditivo adaptativo em processos *Single-Input Single-Output* (SISO) não lineares. Em Blazic et al. (2014) é proposta uma metodologia de controle nebuloso adaptativo com antecedente evolutivo aplicada a processos SISO não lineares com dinâmica dominante de primeira ordem. Em Zdešar et al. (2014) é proposto um algoritmo de controle com dois graus de liberdade aplicado a um processo SISO não linear. O controle baseia-se no modelo nebuloso evolutivo T-S e utiliza o algoritmo de agrupamento *recursive Gustafson-Kessel*.

Neste artigo é proposta uma metodologia de controle preditivo baseado em modelo nebuloso T-S evolutivo para processos SISO com dinâmica não linear e parâmetros incertos. São utilizados os mecanismos de adição, fusão e exclusão de regras. O sinal de controle é obtido com base no algoritmo *Practical Nonlinear Predictive Control* (PNMPC) proposto por Plucenio (2010) na presença de restrições nas variáveis de entrada, incremento de controle e saída, respectivamente. Para avaliar a metodologia proposta é realizado um estudo de caso com o *surge tank* (Boulkaibet et al., 2017) nos seguintes cenários: seguimento de referência, rejeição de distúrbio e robustez às incertezas paramétricas.

2 Modelagem Nebulosa Evolutiva

Neste artigo, o modelo nebuloso evolutivo T-S apresenta a i -ésima regra, dada por:

$$R^i : \mathbf{SE} (\mathbf{z}_k \sim \mathbf{z}^{i*}) \text{ ENTÃO } y_k^i = \mathbf{r}_{k-1}^T \theta_k^i \quad (1)$$

onde $i = 1, 2, \dots, L$ é o número da regra, $\mathbf{z}_k = [z_k^1 \ z_k^2 \ \dots \ z_k^p]^T$ são as variáveis do antecedente, $\mathbf{z}^{i*} = [z_1^{i*} \ z_2^{i*} \ \dots \ z_p^{i*}]$ é o centro da i -ésima regra, $\mathbf{r}_{k-1}^T = [y_{k-1} \ \dots \ y_{k-n_y} \ u_{k-1} \ \dots \ u_{k-n_u}] \in \mathbb{R}^{1 \times (n_u + n_y)}$ é o vetor de regressores e $\theta_k^i \in \mathbb{R}^{(n_u + n_y) \times 1}$ é vetor de parâmetros de cada regra obtido no instante de tempo atual k .

O grau de ativação de um ponto na i -ésima regra é representado por uma gaussiana com centro z_j^{i*} , dada por:

$$\mu_j^i(z_k^j) = e^{-\frac{(z_k^j - z_j^{i*})^2}{2(\sigma_{k,j}^i)^2}} \quad (2)$$

onde $\sigma_{k,j}^i$ é variância da j -ésima variável do antecedente da i -ésima regra obtida no instante k .

O grau de ativação normalizado da i -ésima regra, é calculado como segue:

$$\gamma^i(\mathbf{z}_k) = \frac{\prod_{j=1}^p \mu_j^i(z_k^j)}{\sum_{i=1}^{N_c} \prod_{j=1}^p \mu_j^i(z_k^j)} \quad (3)$$

A saída do modelo nebuloso TS, é dada por:

$$y_k = \sum_{i=1}^L \gamma^i(\mathbf{z}_k) y_k^i \quad (4)$$

Substituindo (1) em (4), encontra-se:

$$y_k = \mathbf{r}_{k-1}^T \sum_{i=1}^L \gamma^i(\mathbf{z}_k) \theta_k^i = f(\mathbf{r}_{k-1}^T) \quad (5)$$

onde f é uma função não linear que representa todo o modelo nebuloso.

2.1 Estimação dos Parâmetros do Antecedente: Abordagem Evolutiva

O algoritmo utilizado para estimar as regras nebulosas emprega o conceito da Estimação Recursiva da Densidade (Angelov and Zhou, 2008). A densidade da amostra \mathbf{z}_k é dada por:

$$D(\mathbf{z}_k) = \frac{1}{1 + \frac{1}{k-1} \sum_{i=1}^k |\mathbf{z}_k - \mathbf{z}^{i*}|} \quad (6)$$

Recursivamente, a equação (6) pode ser formulada como (Angelov and Zhou, 2008):

$$D_k(\mathbf{z}_k) = \frac{k-1}{(k-1) \left(\sum_{i=1}^p (z_k^i)^2 + 1 \right) + b_k - \Lambda_k} \quad (7)$$

¹Neste artigo utilizou-se $\mathbf{z}_k = [u_{k-1} \ y_{k-1}]$, logo $p = 2$.

onde $\Lambda_k = 2 \sum_{i=1}^p z_k^i c_k^i$, $D_1(\mathbf{z}_1) = 1$, $b_k = b_{k-1} + \sum_{j=1}^p (z_{k-1}^j)^2$, $b_1 = 0$ e $c_k^j = c_{k-1}^j + z_{k-1}^j$.

A condição utilizada para selecionar uma amostra como um novo centro, chamada **Condição A**, é dada por (Angelov, 2012):

$$\mathbf{SE} D_k(\mathbf{z}_k) > \max_{i=1}^L D_k(\mathbf{z}^{i*}) \text{ ou } D_k(\mathbf{z}_k) > \min_{i=1}^L D_k(\mathbf{z}^{i*}) \text{ ENTÃO } L = L + 1 \quad (8)$$

Quando \mathbf{z}_k é eleito como novo centro, uma nova regra é criada e sua densidade precisa ser atualizada a cada nova amostra. Isso é feito, recursivamente, como:

$$D_k(\mathbf{z}^{i*}) = \frac{k-1}{k-1 + (k-2) \left(\frac{1}{D_{k-1}(\mathbf{z}^{i*})} - 1 \right) + \Psi_k} \quad (9)$$

onde $\Psi_k = \sum_{j=1}^p (z_k^j - z_{k-1}^j)^2$.

A fim de evitar redundância de regras, a **Condição B** é usada (Angelov and Kordon, 2010):

$$\mathbf{SE} \mu_i^j(z_k^j) > \epsilon \ (\forall j = 1, \dots, p) \text{ ENTÃO } L = L - 1 \quad (10)$$

com $i = 1, \dots, L - 1$. Desse modo, assim que uma nova regra é criada, todas as regras pré-existent com grau de ativação (2) acima de certo valor de tolerância ϵ sofrem fusão, evitando redundância de regras coexistentes.

Para evitar que regras com baixa utilidade sejam mantidas na base de regras, minimizando o efeito de *outliers*, a **Condição C** é usada, como segue (Angelov and Kordon, 2010):

$$\mathbf{SE} U_k^i < \eta \text{ ENTÃO } L = L - 1 \quad (11)$$

com $\eta \in [0, 01 \ 0, 3]$, $i = 1, \dots, L$ e

$$U_k^i = \frac{\sum_{j=1}^k \gamma_j^i(\mathbf{z}_k)}{k - I^{i*}} \quad (12)$$

é a utilidade da regra, que calcula o valor médio do grau de ativação normalizado (3) a partir do instante de tempo I^{i*} (idade da regra) em que a regra foi criada até o instante k .

Por fim, a variância $\sigma_{k,j}^i$, também conhecida como zona de influência da regra, é atualizada recursivamente, como segue (Angelov, 2012):

$$\sigma_{k,j}^i = \sqrt{\zeta (\sigma_{k-1,j}^i)^2 + (1 - \zeta) \frac{1}{S_k^i} (z_k^j - z_j^{i*})^2} \quad (13)$$

onde ζ é a constante de aprendizagem e S_k^i é igual ao número de amostras associadas à i -ésima regra no instante k .

2.2 Estimação dos Parâmetros do Consequente

Os parâmetros do consequente θ_k^i são obtidos pela solução recursiva, via mínimos quadrados

(Lughofer, 2011), do seguinte conjunto de equações:

$$\begin{aligned} g^i &= \mathbf{P}_{k-1}^i \mathbf{r}_{k-1} (\lambda_i / \gamma(\mathbf{z}_k) + \mathbf{r}_{k-1}^T \mathbf{P}_{k-1}^i \mathbf{r}_{k-1})^{-1} \\ \theta_k^i &= \theta_{k-1}^i + g^i * (y_k - \mathbf{r}_{k-1}^T \theta_{k-1}^i) \\ \mathbf{P}_k^i &= (\lambda_i^{-1} \mathbf{I} - g^i \mathbf{r}_{k-1}^T) \mathbf{P}_{k-1}^i \end{aligned} \quad (14)$$

onde $i = 1, 2, \dots, L$, $\lambda_i \in [0.95, 1]$ é o fator de esquecimento e \mathbf{P}_k^i é matriz de covariância do erro da i -ésima regra.

Desse modo, todo o processo de modelagem descrito nesta seção é expresso no Algoritmo 1.

Algoritmo 1: Modelagem Nebulosa

Evolutiva

início

Inicialização;

$k = 1$;

Leia a amostra \mathbf{z}_k ;

$L = 1$; $\mathbf{z}^{i*} = \mathbf{z}_k$; $D_k(\mathbf{z}_k) = 1$; $I^{i*} = k$;

repita

$k = k + 1$;

Calcular a densidade $D_k(\mathbf{z}_k)$ (7);

Atualizar a densidade dos centros

$D_k(\mathbf{z}^{i*})$ (9);

se *Condição A* (8) **então**

$L = L + 1$; $\mathbf{z}^{i*} = \mathbf{z}_k$;

$D_k(\mathbf{z}^{i*}) = D_k(\mathbf{z}_k)$; $I^{i*} = k$;

$\mathbf{P}_k^L = \alpha \mathbf{I}$; $\theta_k^L = \mathbf{0}$

se *Condição B* (10) **então**

Remover todas as regras
que satisfazem essa
condição;

fim

fim

Atualizar a utilidade das regras

U_k^i (12);

se *Condição C* (11) **então**

Remover todas as regras que
satisfazem essa condição;

fim

Atualizar a variância σ_i^j ; **para**

$i = 1$ até L **faça**

Atualizar parâmetros do
consequente θ_k^i (14)

fim

até $\mathbf{z}_{k+1} = \emptyset$;

fim

3 Controle Preditivo Baseado em Modelo Nebuloso Evolutivo

Em geral, os diversos algoritmos de CPBM utilizam diferentes funções custo para calcular a lei de controle, porém praticamente todos eles consideram como objetivo minimizar um índice J que considera o erro entre as predições da saída ($\hat{y}_{k+j|k}$) e a referência futura (ω_{k+j}) e penaliza,

em alguns casos, o sinal de controle (u_k) ou seus incrementos (Δu_k):

$$J = \sum_{j=1}^N [\hat{y}_{k+j|k} - \omega_{k+j}]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u_{k+j-1}]^2 \quad (15)$$

onde N é o horizonte de predição, N_u é o horizonte de controle e λ é a ponderação do erro.

Para o cálculo das predições utiliza-se uma estrutura que combina o modelo nebuloso evolutivo (5) com o modelo de perturbação (distúrbio). Sendo assim, adicionando-se o termo $e(k)/\Delta$ em (5) tem-se:

$$y_k = \mathbf{r}_{k-1}^T \sum_{i=1}^L \gamma^i(\mathbf{z}_k) \theta_k^i + \frac{e(k)}{\Delta} \quad (16)$$

onde $\Delta = 1 - q^{-1}$ é o operador de diferenças e $e(k)$ é um ruído branco com média nula.

Fazendo manipulações algébricas em (16), obtém-se:

$$y_k = y_{k-1} + \Delta \mathbf{r}_{k-1}^T \sum_{i=1}^L \gamma^i(\mathbf{z}_k) \theta_k^i + e(k) \quad (17)$$

onde

$$\Delta \mathbf{r}_{k-1}^T = [\Delta y_{k-1} \dots \Delta y_{k-n_y} \Delta u_{k-1} \dots \Delta u_{k-n_u}] \quad (18)$$

Para calcular a predição ótima da saída do sistema em um instante de tempo $k+j$ com informação conhecida em k utiliza-se o valor esperado da variável $y_{k+j|k} = E(y_{k+j})$, que possui melhor valor, no sentido estocástico, quando o ruído futuro ($e(k+j)$) é nulo, resultando em:

$$\hat{y}_{k+j|k} = \hat{y}_{k+j-1|k} + \Delta \hat{\mathbf{r}}_{k+j-1|k}^T \sum_{i=1}^L \gamma^i(\mathbf{z}_{k+j}) \theta_k^i \quad (19)$$

A fim de calcular as predições do modelo nebuloso evolutivo (5) utilizou-se a estratégia PNMPC, que tem como principal vantagem a utilização da formulação tradicional de cálculo da predição, dividindo-a em respostas livre e forçada, enquanto se mantém maior proximidade da resposta real do processo ao utilizar um modelo não linear para determinar a resposta livre. Baseado nesta técnica, bem como na apresentada em Oviado et al. (2004), utilizou-se a seguinte aproximação

$$\mathbf{Y} \approx \mathbf{G}_k \Delta \mathbf{U} + \mathbf{Y}_k^{\text{livre}} \quad (20)$$

onde

$$\mathbf{Y} = [\hat{y}_{k+1|k} \dots \hat{y}_{k+N|k}]^T \quad (21)$$

é o vetor de predições da saída,

$$\mathbf{Y}_k^{\text{livre}} = [\hat{y}_{k+1|k}^{\text{livre}} \dots \hat{y}_{k+N|k}^{\text{livre}}]^T \quad (22)$$

é o vetor de resposta livre,

$$\Delta \mathbf{U} = [\Delta u(k|k) \Delta u(k+1|k) \dots \Delta u(k+N_u-1|k)]^T$$

é o vetor de incrementos no controle e \mathbf{G}_k é a matriz de resposta ao salto. Desse modo, as predições são calculadas de modo similar aos algoritmos DMC e GPC (Plucenio, 2010).

No entanto, é importante observar que a representação dada em (20) não é uma representação linear, pois o vetor $\mathbf{Y}_k^{\text{livre}}$ é gerado via simulação usando o modelo nebuloso evolutivo (5) e os elementos da matriz \mathbf{G}_k dependem do ponto de operação e da amplitude do sinal de entrada (Oviedo et al., 2004).

3.1 Resposta Livre

Para calcular o vetor $\mathbf{Y}_k^{\text{livre}}$ é necessário simular o comportamento processo, por meio do modelo nebuloso evolutivo (5) ao longo do horizonte de predição (1 até N) fazendo com que o último sinal de controle aplicado (u_{k-1}) seja mantido constante ao longo do horizonte. Isto pode ser feito por meio do procedimento descrito no Algoritmo 2.

3.2 Resposta Forçada

Para calcular a matriz \mathbf{G}_k utiliza-se a seguinte equação (Plucenio, 2010):

$$\mathbf{G}_k = \frac{\partial \mathbf{Y}}{\partial \Delta \mathbf{U}} \quad (23)$$

que pode ser calculada por uma aproximação. Para isso, é necessário estimar a resposta forçada, expressa por:

$$\mathbf{Y}_k^{\text{forçada}} = [\hat{y}_{k+1|k}^{\text{forçada}} \ \dots \ \hat{y}_{k+N|k}^{\text{forçada}}]^T \quad (24)$$

que, por sua vez, é obtida quando o modelo nebuloso evolutivo (5) recebe um pequeno incremento Δu_{\min} na variável de controle ($u_{k-1} + \Delta u_{\min}$), conforme procedimento descrito no Algoritmo 3.

Assim, a primeira coluna da matriz \mathbf{G}_k é calculada como:

$$\mathbf{G}_k^1 = \frac{\mathbf{Y}_k^{\text{forçada}} - \mathbf{Y}_k^{\text{livre}}}{\Delta u_{\min}} = [\tilde{g}_0 \ \tilde{g}_1 \ \dots \ \tilde{g}_{N-1}]^T \quad (25)$$

que são os coeficientes da resposta ao salto. A matriz \mathbf{G}_k , por sua vez, é expressa como:

$$\mathbf{G}_k = \begin{bmatrix} \tilde{g}_0 & 0 & \dots & 0 \\ \tilde{g}_1 & \tilde{g}_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{g}_{N-1} & \tilde{g}_{N-2} & \dots & \tilde{g}_{N-N_u} \end{bmatrix} \quad (26)$$

Definindo o vetor de referências futuras

$$\mathbf{W} = [\omega_{k+1} \ \omega_{k+2} \ \dots \ \omega_{k+N}]^T \quad (27)$$

a função custo (15) pode ser expressa matricialmente por:

$$J = \frac{1}{2} (\Delta \mathbf{U})^T \mathbf{H} \Delta \mathbf{U} + \mathbf{f}_0^T \Delta \mathbf{U} + \mathbf{b}_0 \quad (28)$$

onde

$$\mathbf{H} = 2(\mathbf{G}_k^T \mathbf{G}_k + \lambda \mathbf{I})$$

$$\mathbf{f}_0^T = 2(\mathbf{Y}_k^{\text{livre}} - \mathbf{W})^T \mathbf{G}_k$$

$$\mathbf{b}_0^T = (\mathbf{Y}_k^{\text{livre}} - \mathbf{W})^T (\mathbf{Y}_k^{\text{livre}} - \mathbf{W})$$

que é resolvido por qualquer algoritmo de programação quadrática (QP), sujeito as seguintes restrições:

$$u_{\min} \leq u_{k+j|k} \leq u_{\max} \quad (29)$$

$$du_{\min} \leq \Delta u_{k+j|k} \leq du_{\max} \quad (30)$$

$$y_{\min} \leq \hat{y}_{k+j|k} \leq y_{\max} \quad (31)$$

3.3 Metodologia de Controle Preditivo Baseada em Modelo Nebuloso Evolutivo Proposta

A metodologia de controle proposta neste artigo consiste em três etapas:

1. Etapa de treinamento: Com base em um conjunto inicial de N_a amostras de entrada-saída do processo, é identificado um modelo nebuloso inicial, utilizando-se o Algoritmo 1. Esta etapa é muito importante, pois caso o modelo nebuloso inicial apresente baixa qualidade, o controlador poderá apresentar performance insatisfatória.
2. Etapa de controle: O sinal de controle deve ser obtido, em tempo hábil, pela minimização da função custo (28). Para isso, segue um procedimento que inicia com a estimação da resposta livre $\mathbf{Y}_k^{\text{livre}}$ (Algoritmo 2), estimação da resposta forçada $\mathbf{Y}_k^{\text{forçada}}$ (Algoritmo 3), cálculo da matriz \mathbf{G}_k (25)-(26), definição do vetor de referências futuras \mathbf{W} (27) e minimização do funcional J (28).
3. Etapa de Identificação: À medida que informações (\mathbf{z}_{k+1}) do processo são adquiridas em tempo real, o modelo nebuloso evolutivo atualiza os parâmetros θ_k^i existentes e altera o número regras, se necessário (Algoritmo 1).

4 Resultados Computacionais

Para avaliar a metodologia, utilizou-se o sistema *surge tank* (Boulkaibet et al., 2017) representado na Figura 1. O modelo matemático deste sistema não linear é dado por:

$$\frac{dh_t}{dt} = \frac{-c_s \sqrt{2g_s h_t}}{A(h_t)} + \frac{1}{A(h_t)} u_t \quad (32)$$

onde u_t é a entrada de controle (entrada de fluxo do sistema), h_t é o nível do líquido e as constantes $c_s = 1$ e $g_s = 9,8$. A área da seção transversal do *surge tank* é definida como: $A(h_t) = a_s(h_t)^2 + b_s$. As constantes: $a_s = 0,01$ e $b_s = 0,2$.

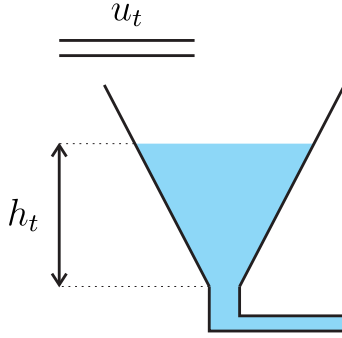
A metodologia proposta, por sua vez, foi implementada com outros dois algoritmos de agrupamento:

Algoritmo 2: Resposta Livre

início $\hat{y}_{k+n}^{\text{livre}} = y_{k+n}$ (para $n < 1$) e $u_{k+m} = u_{k-1}$ (para $m > 0$);**para** $j = 1$ **até** N **faça** $\mathbf{z}_{k+j}^{\text{livre}} = [u_{k-1} \ \hat{y}_{k+j-1}^{\text{livre}}]$; $\Delta \hat{\mathbf{r}}_{k+j-1}^{\text{livre}} = [\Delta \hat{y}_{k+j-1}^{\text{livre}} \ \dots \ \Delta \hat{y}_{k+j-n_y}^{\text{livre}}]^T$; $\hat{y}_{k+j}^{\text{livre}} = \hat{y}_{k+j-1}^{\text{livre}} + (\Delta \hat{\mathbf{r}}_{k+j-1}^{\text{livre}})^T \sum_{i=1}^L \gamma^i (\mathbf{z}_{k+j}^{\text{livre}}) \theta_k^i$;**fim****fim**

Algoritmo 3: Resposta Forçada

início $\hat{y}_{k+n}^{\text{forçada}} = y_{k+n}$ (para $n < 1$) e $u_{k+m} = u_{k-1} + \Delta u_{\min}$ (para $m > 0$);**para** $j = 1$ **até** N **faça** $\mathbf{z}_{k+j}^{\text{forçada}} = [u_{k-1} \ \hat{y}_{k+j-1}^{\text{forçada}}]$; $\Delta \hat{\mathbf{r}}_{k+j-1}^{\text{forçada}} = [\Delta \hat{y}_{k+j-1}^{\text{forçada}} \ \dots \ \Delta \hat{y}_{k+j-n_y}^{\text{forçada}}]^T$; $\hat{y}_{k+j}^{\text{forçada}} = \hat{y}_{k+j-1}^{\text{forçada}} + (\Delta \hat{\mathbf{r}}_{k+j-1}^{\text{forçada}})^T \sum_{i=1}^L \gamma^i (\mathbf{z}_{k+j}^{\text{forçada}}) \theta_k^i$;**fim****fim**

Figura 1: O processo *surge tank*.

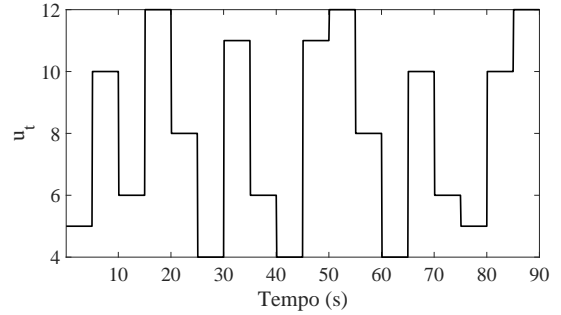
- Agrupamento evolutivo FLEXFIS (Lughofer and Klement, 2005);
- Agrupamento *off-line* Gustafson-Kessel (GK) (Babuška, 1998).

Como resultado, as estratégias de controle baseadas nestes agrupamentos foram chamadas, neste artigo, de GPC-FLEXFIS e GPC-GK. De modo que, comparações foram realizadas nas seguintes situações: qualidade do modelo nebuloso inicial, seguimento de referência, rejeição de distúrbio e variação paramétrica.

4.1 Etapa de treinamento

Para obter o modelo nebuloso inicial do *surge tank*, o modelo não linear (32) foi usado para gerar um conjunto de $N_a = 900$ amostras de entrada e saída a um intervalo de amostragem $T_s = 0,1$ s e com $\mathbf{r}_{k-1}^T = [y_{k-1} \ y_{k-2} \ y_{k-3} \ u_{k-1}]$. O sinal

de entrada aplicado, bem como a saída obtida são apresentados nas Figuras 2 e 3, respectivamente. Nesta etapa foram utilizados, para o Algoritmo 1, os parâmetros: $\zeta = 0,98$, $\eta = 0,01$, $\epsilon = 0,8$, $\lambda_i = 0,99$ e $\alpha = 1000$ com $\mathbf{r}_{k-1}^T = [y_{k-1} \ y_{k-2} \ y_{k-3} \ u_{k-1}]$.

Figura 2: Sinal de entrada (u_t) aplicado ao *surge tank*.

Para o algoritmo de agrupamento evolutivo FLEXFIS foram utilizados os seguintes parâmetros: $fac = 0,2$ e $init_{gain} = 0,99$. Para o GK foram utilizados: $L = 11$ e $m = 2$ (expoente de ponderação). A validação dos modelos nebulosos é apresentada na Figura 4. A Tabela 1 apresenta, através do *Root Mean Squared Error* (RMSE), uma comparação do desempenho de cada técnica.

A Figura 5 mostra a evolução do número de regras (L) na etapa de treinamento para os algoritmos de agrupamento evolutivo FLEXFIS e proposto (Algoritmo 1).

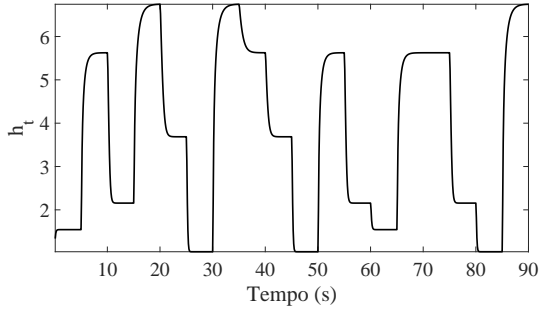


Figura 3: Sinal de saída (h_t) do surge tank usado na etapa de treinamento.

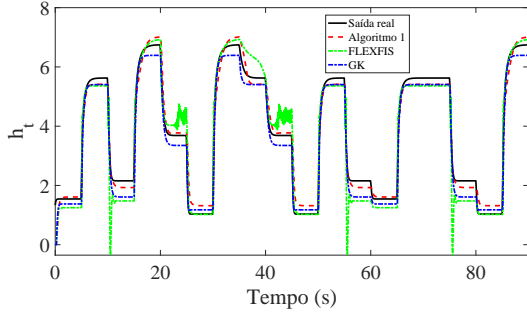


Figura 4: Validação dos modelos nebulosos obtidos na etapa de treinamento.

Método	RMSE	Número de regras (L)
Algoritmo 1	0,2991	7
GK	0,3326	11
FLEXFIS	0,5082	10

Tabela 1: Comparação entre diferentes técnicas de agrupamento nebuloso na etapa de treinamento.

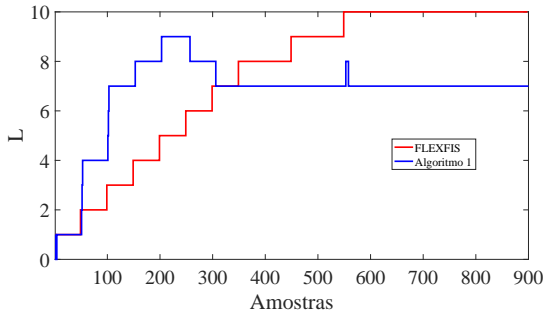


Figura 5: Evolução do número de regras (L) dos algoritmos de agrupamento evolutivo na etapa de treinamento.

4.2 Etapa de controle

A metodologia proposta, bem como as estratégias GPC-FLEXFIS e GPC-GK foram implementadas com os seguintes parâmetros: $N_u = 4$, $N = 10$, $\lambda = 0,01$, $\Delta u_{\min} = -0,3$, $\Delta u_{\max} = 0,3$, $u_{\min} = 4$, $u_{\max} = 15$, $y_{\min} = 1$, $y_{\max} = 12$. Dessa forma, tais estratégias foram submetidas aos seguintes cenários:

rios:

- *Cenário 1:* A metodologia proposta foi submetida ao seguimento de uma trajetória de referência dada por:

$$\omega_k = \begin{cases} 5, & 0 \leq k \leq 40 \text{ s}, \\ 7, & 40 \leq k \leq 80 \text{ s}, \\ 6, & 80 \leq k \leq 120 \text{ s}, \\ 4, & 120 \leq k \leq 180 \text{ s}, \\ 7, & 180 \leq k \leq 200 \text{ s}. \end{cases} \quad (33)$$

As Figuras 6 e 7 apresentam, respectivamente, as respostas de saída (h_t) e controle (u_t), para as estratégias implementadas durante o rastreo de ω_k (33).

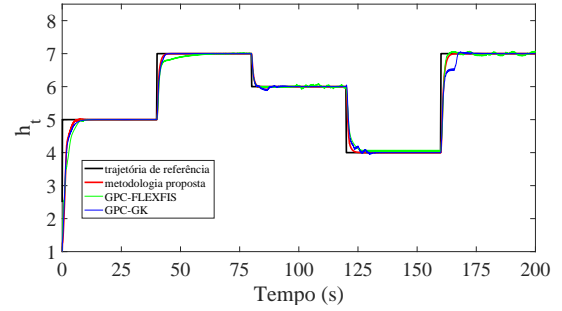


Figura 6: Resposta de saída (h_t) do surge tank para as estratégias de controle implementadas no cenário 1.

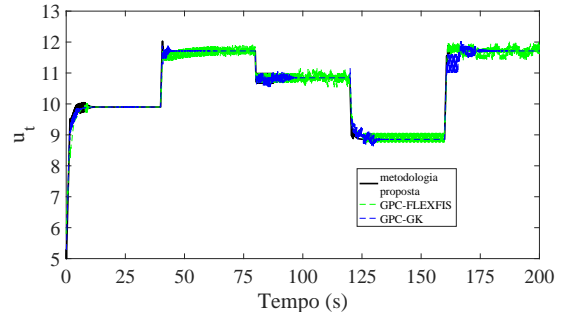


Figura 7: Entrada de controle (u_t) aplicada ao surge tank para as estratégias de controle implementadas no cenário 1.

- *Cenário 2:* A metodologia proposta foi submetida ao seguimento da trajetória de referência ω_k (33) adicionando-se um distúrbio de saída d_k , dado por:

$$d_k = \begin{cases} +1, & 20 \leq k \leq 60 \text{ s}, \\ -1, & 100 \leq k \leq 140 \text{ s}, \\ +1, & 180 \leq k \leq 200 \text{ s}, \end{cases} \quad (34)$$

As respostas da saída (h_t) e controle (u_t), para as estratégias implementadas, no cenário 2 são apresentadas nas Figuras 8 e 9, respectivamente.

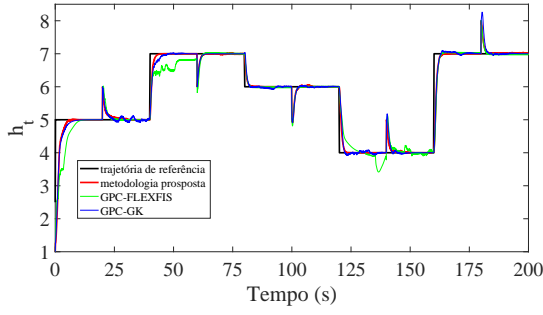


Figura 8: Resposta de saída (h_t) do *surge tank* para as estratégias de controle implementadas no cenário 2.

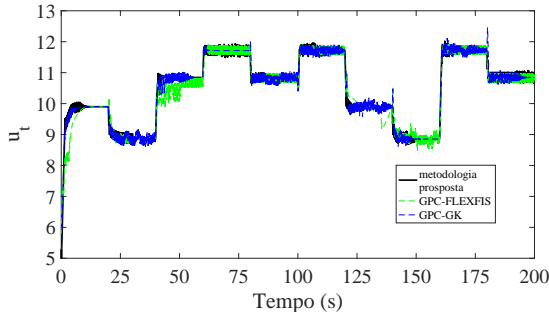


Figura 9: Entrada de controle (u_t) aplicada ao *surge tank* para as estratégias de controle implementadas no cenário 2.

- *Cenário 3*: A metodologia proposta foi submetida ao seguimento da trajetória de referência ω_k (33), quando na variação paramétrica de c_s do modelo não linear (32), segundo a Tabela 2. As Figuras 10 e 11 apresentam, respectivamente, as respostas de saída (h_t) e controle (u_t) neste cenário.

Tempo (s)	c_s
$0 \leq t \leq 20$	0.5
$20 < t \leq 60$	0.6
$60 < t \leq 100$	0.5
$100 < t \leq 140$	1
$140 < t \leq 180$	1.5
$180 < t \leq 200$	2

Tabela 2: Tabela de variações paramétricas de c_s e g_s (32).

É importante observar que o modelo nebuloso T-S evolutivo continua contínuo e diferenciável mesmo na presença de tais incertezas paramétricas.

4.3 Etapa de Identificação

Nesta etapa foram mantidos inalterados os parâmetros do Algoritmo 1 durante a Etapa de Treinamento. A Figura 12 exibe a evolução do número de regras do modelo nebuloso evolutivo, na

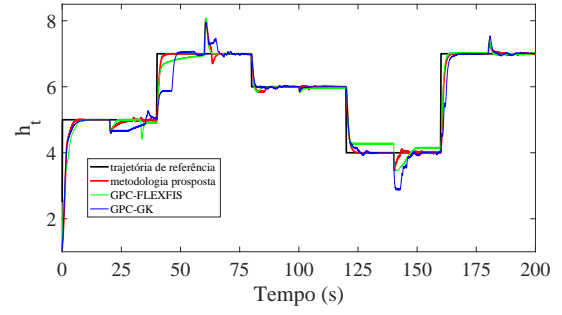


Figura 10: Resposta de saída (h_t) do *surge tank* para as estratégias de controle implementadas no cenário 3.

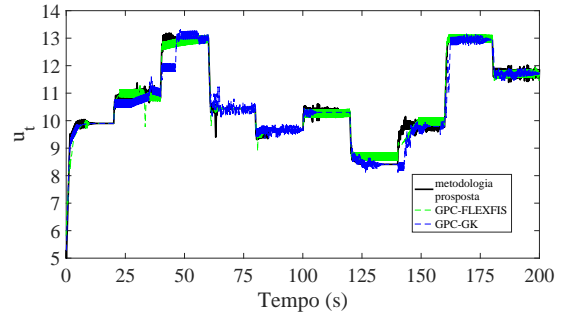


Figura 11: Entrada de controle (u_t) aplicada ao *surge tank* para as estratégias de controle implementadas no cenário 3.

metodologia proposta, para os diferentes cenários apresentados na Etapa de Controle.

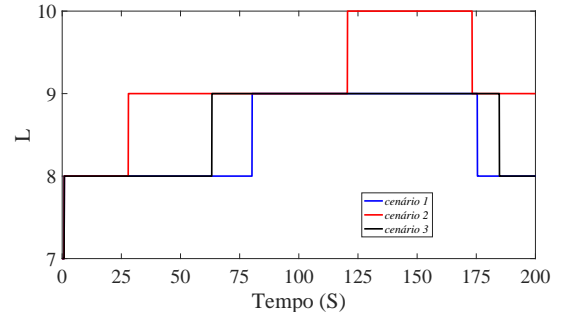


Figura 12: Evolução do número de regras (L) do modelo nebuloso evolutivo proposto na etapa de controle para os cenários 1, 2 e 3.

5 Contribuição deste artigo

As principais contribuições deste artigo são:

- Proposta de metodologia de controle preditivo baseado em modelo nebuloso T-S evolutivo;
- Comparações realizadas entre os agrupamentos nebulosos evolutivo FLEXFIS, *off-line* GK e proposto (Algoritmo 1);

- Estudo de caso com o processo não linear *surge-tank*.

6 Conclusões

Este artigo propôs uma metodologia de controle preditivo baseado em modelo nebuloso T-S evolutivo. A metodologia, por sua vez, foi aplicada ao processo não linear SISO *surge-tank*.

O impacto do agrupamento nebuloso evolutivo nas etapas de treinamento e controle foi avaliada por meio da comparação com os algoritmos de agrupamento evolutivo FLEXFIS e *off-line* GK.

Por fim, comparações foram realizadas em três cenários frequentes em controle de processos: seguimento de referência, rejeição de distúrbio e robustez a variações paramétricas.

Referências

- Angelov, P. (2012). *Autonomous Learning Systems: From Data Streams to Knowledge in Real-time*, Wiley.
- Angelov, P. and Kordon, A. (2010). Adaptive inferential sensors based on evolving fuzzy models, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **40**(2): 529–539.
- Angelov, P. P. and Filev, D. P. (2004). An approach to online identification of takagi-sugeno fuzzy models, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **34**(1): 484–498.
- Angelov, P. P. and Zhou, X. (2008). Evolving fuzzy-rule-based classifiers from data streams, *IEEE Transactions on Fuzzy Systems* **16**(6): 1462–1475.
- Babuška, R. (1998). *Fuzzy Modeling for Control*, International Series in Intelligent Technologies, Springer Netherlands.
- Blazic, S., Dovzan, D. and Škrjanc, I. (2014). Robust evolving fuzzy adaptive control with input-domain clustering, *IFAC Proceedings Volumes* **47**(3): 5387 – 5392. 19th IFAC World Congress.
- Boulkaibet, I., Belarbi, K., Bououden, S., Marwala, T. and Chadli, M. (2017). A new t-s fuzzy model predictive control for nonlinear processes, *Expert Systems with Applications* **88**: 132 – 151.
- Dovzan, D. and Škrjanc, I. (2011). Recursive clustering based on a gustafson-kessel algorithm, *Evolving Systems* **2**(1): 15–24.
- Eski, İ. and Temürlenk, A. (2013). Design of neural network-based control systems for active steering system, *Nonlinear Dynamics* **73**(3): 1443–1454.
- Kasabov, N. K. and Song, Q. (2002). Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Transactions on Fuzzy Systems* **10**(2): 144–154.
- Lughofer, E. (2011). *Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications*, Studies in Fuzziness and Soft Computing, Springer Berlin Heidelberg.
- Lughofer, E. and Klement, E. P. (2005). Flexfis: A variant for incremental learning of takagi-sugeno fuzzy systems, *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pp. 915–920.
- Oviedo, J., Vandewalle, J. and Wertz, V. (2004). *Fuzzy Logic, Identification and Predictive Control*, Advances in Industrial Control, Springer London.
- Plucenio, A. (2010). *Desenvolvimento de Técnicas de Controle Não Linear para Elevação e Fluidos Multifásicos*, PhD thesis, Universidade Federal de Santa Catarina.
- Rastegar, S., Araújo, R. and Mendes, J. (2016). A new approach for online t-s fuzzy identification and model predictive control of nonlinear systems, *Journal of Vibration and Control* **22**(7): 1820–1837.
- Zdešar, A., Dovžan, D. and Škrjanc, I. (2014). Self-tuning of 2 dof control based on evolving fuzzy model, *Applied Soft Computing* **19**: 403 – 418.