

Sintonia do Controlador Evolutivo Granular Robusto por Realimentação Linearizante via Algoritmo de Evolução Diferencial^{*}

Lincoln C. Santos^{*} Lucas S. Oliveira^{*} Valter J. S. Leite^{*}
Fernando Gomide^{**}

^{*} Departamento de Engenharia Mecatrônica, Centro Federal de Educação Tecnológica de Minas Gerais, MG, (e-mail: lincolncaiue@gmail.com, lqsoliveira@cefetmg.br, valter@ieee.org).

^{**} Faculdade de Engenharia Elétrica e da Computação, Universidade Estadual de Campinas, SP (e-mail: gomide@dca.fee.unicamp.br)

Abstract: Robust evolving granular feedback linearization – ReGFL controller is a method that improves robustness to the exact feedback linearization approach. The main feature of this controller, it does not rely upon previews knowledge about the nonlinear functions that make up the system. However, to apply the ReGFL controller, the user needs to set up a set with several parameters that involve directly, the controller dynamics, and, consequently, their performance. This specificity becomes the application in the real systems in a complex proceeding. In this paper, is proposed the use of the differential evolution algorithm in an offline process to tuning the controller parameters. We apply the proposed approach to a level control problem in a nonlinear surge tank. Classical quantitative indexes are used to give an idea of the performance improvement yielded by our approach. The achievements suggest that the proposed tuning method of the ReGFL controller significantly improves the closed-loop performance.

Resumo: O controlador evolutivo granular robusto por realimentação linearizante – ReGFL consiste de um método online que agrega robustez a técnica de linearização por realimentação exata. Como principal característica, essa abordagem não exige o conhecimento prévio das funções não lineares que compõem o sistema. Contudo, seu uso está sujeito a escolha de diversos parâmetros que afetam diretamente a dinâmica do controlador, e consequentemente, seu desempenho. Essa característica torna sua aplicação em sistemas reais um procedimento complexo. Neste trabalho, é proposto o uso do algoritmo de evolução diferencial em uma operação offline para sintonia dos parâmetros do controlador. A abordagem proposta é aplicada a um sistema de controle de nível em um tanque não linear. Os resultados são quantificados com o auxílio de índices de desempenho clássicos que indicam a melhora de desempenho proporcionada por nossa abordagem. A análise desses critérios comparada ao desempenho do controlador ReGFL sintonizado via metodologia padrão e controle por realimentação linearizante. Os resultados alcançados indicam que o método de sintonia proposto para o controlador ReGFL proporciona significativa melhora de desempenho à malha fechada.

Keywords: Feedback linearization; granular control; adaptive control; differential evolution; evolving systems.

Palavras-chaves: Linearização por realimentação; controle granular; controle adaptativo; evolução diferencial; sistemas evolutivos.

1. INTRODUÇÃO

Sistemas evolutivos fuzzy (*evolving fuzzy systems* – EFS) possuem a habilidade de adaptação simultânea tanto da estrutura dos modelos quanto do ajuste dos parâmetros que os compõem. Essa adaptação ocorre a partir da análise do fluxo dos dados de entrada do algoritmo (Silva et al., 2013). Esses sistemas têm sido aplicados em diferentes abordagens, como por exemplo: modelagem de sistemas

dinâmicos, previsão de séries temporais, agrupamento de dados, controle de processos em malha fechada, entre outros (Oliveira et al., 2019a, 2020). Há diversos mecanismos de aprendizagem, agrupamento, e evolução dos modelos ou regras para os EFS (Leite et al., 2020).

Nessa perspectiva, Angelov (2004) faz uso das características do EFS no projeto de controladores fuzzy Takagi-Sugeno (TS), para agregar ao controlador TS a capacidade de evoluir continuamente sua base de regras. Essa abordagem resultou no popular algoritmo evolutivo TS (eTS). A partir do uso do algoritmo de aprendizagem participativa (*evolving participatory learning algorithm* –

^{*} Agradecemos ao CEFET-MG pelo apoio financeiro através do Programa Bolsa de Complementação Educacional – BCE, edital nº092/2018 e CNPq (311208/2019 – 3 e 305906/2014 – 3).

ePL) desenvolvido em (Lima et al., 2006), Silva et al. (2018) apresentam uma topologia de controle similar à utilizada no controle adaptativo por modelo de referência, em que o algoritmo ePL agrega robustez e adaptabilidade a malha de controle por realimentação linearizante (Khalil, 2002, Cap. 13). Em (Oliveira et al., 2019b) uma malha de controle por realimentação linearizante é desenvolvida com ajuda do algoritmo ePL. Nessa abordagem, assume-se que o modelo que descreve o sistema seja incerto com relação aos parâmetros. Assim, o algoritmo ePL encarrega-se da estimação do erro de modelagem da mitigação dos efeitos negativos decorrentes da diferença de dinâmica do modelo em relação ao sistema real. Motivado pela estrutura de controle exposta em (Oliveira et al., 2019b), Oliveira et al. (2019a) propõem uma nova malha de controle com habilidade para estimar as funções não lineares do sistema. Nessa abordagem, desenvolvem uma nova lei de controle linearizante usando *certainty equivalence principle* (Wang, 1996). Desse modo, a linearização por realimentação passa a ser regida por uma lei de controle em que as funções não lineares são estimadas em tempo real. Uma dificuldade nos métodos baseados no algoritmo ePL está no ajuste de seus parâmetros que regem a criação, fusão, ou eliminação de grupos de dados (ou clusters). Em geral, há 8 parâmetros a serem sintonizados, com consequências sobre a capacidade e velocidade de aprendizagem do método. Tais parâmetros, embora possuam faixas típicas de escolha e algumas regras empíricas para escolha, demandam simulações prévias à implementação para verificação do comportamento do ePL em malha fechada.

A contribuição principal deste trabalho é a proposta de uma metodologia para sintonia dos parâmetros do controlador evolutivo granular robusto por realimentação linearizante (ReGFL) introduzido por Oliveira et al. (2019a). Nesta abordagem, faz-se uso do algoritmo de evolução diferencial (DE) para prover a solução de um problema de otimização dos parâmetros do ePL que ditam a dinâmica de evolução do controlador ReGFL. A metodologia proposta, é avaliada em um sistema de controle de nível de um tanque não-linear, comparando o desempenho com a sintonia apresentada para o controlador ReGFL em (Oliveira et al., 2019a) e com o controle linearizante por realimentação clássico apresentado em (Khalil, 2002). O desempenho é quantificado por índices clássicos de desempenho, os quais indicam que a abordagem proposta leva a melhor significativamente no desempenho da malha fechada do processo controlado, especialmente na presença de erros de modelagem.

O trabalho encontra-se organizado como segue: na Seção 2 é apresentada uma breve revisão dos algoritmos ePL, DE e do controlador ReGFL. O problema de otimização que busca sintonizar os parâmetros do controlador é desenvolvido na Seção 3. Uma aplicação usada em diferentes trabalhos da literatura é apresentada na Seção 4, juntamente com a análise de desempenho das estratégias de controle testadas. Por fim, na Seção 5, são apresentadas as conclusões e perspectivas do trabalho.

2. PRELIMINARES

Nesta seção são apresentadas uma breve revisão do algoritmo de aprendizagem participativa, seguida pelo contro-

lador evolutivo granular robusto por realimentação linearizante e, por fim, do algoritmo diferencial evolutivo.

2.1 Algoritmo de Aprendizagem Participativa (ePL)

O algoritmo ePL proposto em (Lima et al., 2006) possui mecanismos de aprendizagem baseados no processo de agrupamento/clusterização dos dados de entrada. Nesse sentido, esse algoritmo possui a capacidade de adaptar os parâmetros dos antecedentes e consequentes simultaneamente, modificando a cada iteração modelos locais presentes em regras fuzzy funcionais da forma (Silva et al., 2018):

$$\text{SE } \mathbf{z}^k \text{ é } A^i \text{ ENTÃO } y_i^k = \gamma^k [\mathbf{z}^k \ 1], \quad (1)$$

em que $\mathbf{z}^k \in [0,1]$ são os dados de entrada, A^i são os antecedentes da i -ésima regra fuzzy, y_i^k e γ^k são a saída e o vetor dos consequentes da regra fuzzy no instante k , respectivamente. Note que cada grupo de dados corresponde a uma regra fuzzy. Além disso, diferentemente de outros algoritmos evolutivos, como por exemplo, o algoritmo evolutivo Takagi-Sugeno (*evolving Takagi-Sugeno algorithm* – eTS), o ePL admite que o centro do grupo não seja necessariamente correspondente a um dado de entrada, sendo portanto, o mais representativo do grupo (Pedrycz and Gomide, 2007). As regras fuzzy são descritas por funções Gaussianas do tipo:

$$\mu_i^k(\mathbf{z}^k) = e^{-r\|\mathbf{z}^k - \mathbf{v}_i^k\|^2}, \quad (2)$$

em que μ_i^k é o grau de pertinência da i -ésima regra fuzzy, $r \in [0,1]$ a dispersão ou zona de influência da função Gaussiana, \mathbf{v}_i^k o centro do cluster e $\|\cdot\|$ a norma euclidiana.

Em função da natureza de aprendizagem não supervisionada e autoadaptativa, o ePL pode, de forma autônoma, criar novos grupos, atualizar os centros das regras existentes ou eliminar grupos redundantes a cada passo k . Neste caso, o índice de alerta, a_i , monitora a compatibilidade dos dados de entrada em relação ao conjunto de regras. Desse modo, se o índice de alerta torna-se superior ao limiar $\tau \in [0,1]$, ou seja, se

$$\operatorname{argmin}_{j=1, \dots, c^k} a_i^{k+1} \geq \tau \quad (3)$$

com c^k igual ao número de grupos, então uma nova regra é criada. Com o cômputo de a_i^{k+1} dado por:

$$\rho_i^k = 1 - \frac{\|\mathbf{z}^k - \mathbf{v}_i^k\|}{\sqrt{p}} \quad (4)$$

$$a_i^{k+1} = a_i^k + \beta (1 - \rho_i^k - a_i^k)$$

em que ρ_i^k é a compatibilidade do dado de entrada \mathbf{z}^k em relação ao centro do grupo \mathbf{v}_i^k , p é a dimensão do espaço dos dados de entrada, e $\beta \in [0,1]$ a taxa do índice de alerta. Caso contrário, o grupo com maior compatibilidade é atualizado com auxílio da taxa de aprendizagem $\alpha \in [0,1]$, como segue,

$$\mathbf{v}_s^{k+1} = \mathbf{v}_s^k + \alpha \left(\rho_s^{k+1-a_s^k} [\mathbf{z}^k - \mathbf{v}_s^k] \right) \quad (5)$$

$$s = \operatorname{argmax}_{j=1, \dots, c^k} \rho_j^k,$$

e portanto, os parâmetros do vetor dos consequentes dos modelos locais são atualizados pelo uso de algum método recursivo, como por exemplo, os mínimos quadrados recursivos (Aguirre, 2007, pp.323). Em seguida, o ePL verifica

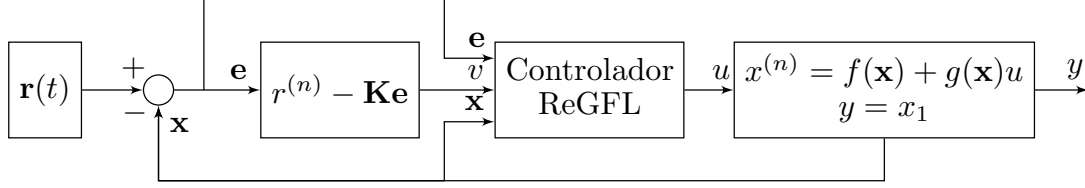


Figura 1. Malha fechada para o controlador evolutivo granular robusto por realimentação linearizante.

se há redundância entre as regras, por meio do cálculo de compatibilidade entre os grupos, ou seja,

$$\rho_{ij}^k = 1 - \frac{\|\mathbf{v}_i^k - \mathbf{v}_j^k\|}{\sqrt{p}}, \quad (6)$$

com $i = 1, \dots, c^k - 1$ e $j = i + 1, \dots, c^k$. Nesse caso, se a compatibilidade entre os grupos for maior que o limiar de compatibilidade $\lambda \in [0, 1]$, ou seja, $\rho_{ij}^k \geq \lambda$ então o grupo \mathbf{v}_j^k é declarado redundante e é removido da base de regras. Caso contrário, a base de regras é mantida. Por fim, a saída do algoritmo é calculada pela média ponderada dos modelos locais, tal que:

$$y^k = \sum_{i=1}^{c^k} \frac{\mu_i^k y_i^k}{\sum_{i=1}^{c^k} \mu_i^k}. \quad (7)$$

2.2 Controle Evolutivo Granular Robusto

Considere a classe de sistemas não lineares

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + B(f(\mathbf{x}) + g(\mathbf{x})u) \\ y &= C\mathbf{x} \end{aligned} \quad (8)$$

em que $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T \in D \subseteq \mathbb{R}^n$, $u \in \mathbb{R}$ e $y \in \mathbb{R}$, são respectivamente os estados, a entrada e saída do sistema, $f(\mathbf{x})$ e $g(\mathbf{x}) \in D \subseteq \mathbb{R}^n$ são funções não lineares infinitamente diferenciáveis pelos estados (Khalil, 2002), $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$ e $C \in \mathbb{R}^{1 \times n}$ são as matrizes canônicas do sistema, tal que:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \text{ e } C = [1 \ 0 \ \dots \ 0].$$

Assumindo que $g(\mathbf{x}) \neq 0 \ \forall \mathbf{x} \in D \subseteq \mathbb{R}^n$, então o sistema (8) pode ser linearizado por realimentação via a lei de controle (Khalil, 2002):

$$u = \frac{1}{g(\mathbf{x})} [v - f(\mathbf{x})] \quad (9)$$

em que v é uma lei de controle linear. Aplicando (9) em (8), obtém-se a dinâmica em malha fechada para o sistema

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + Bv. \quad (10)$$

Contudo, em sistemas físicos é comum que as funções $f(\mathbf{x})$ e $g(\mathbf{x})$ sejam parcialmente ou completamente desconhecidas, impossibilitando a obtenção da dinâmica desejada para a malha fechada (10), levando o sistema em malha fechada à perda de desempenho ou, até mesmo, à instabilidade (Oliveira et al., 2020). Recentemente, Oliveira et al. (2019a) apresentaram o controle evolutivo granular robusto por realimentação linearizante (*robust evolving granular feedback linearization* – ReGFL). Nessa abordagem, usam o algoritmo ePL apresentado na Seção 2.1 para

a estimação em tempo real dos valores das funções não lineares do sistema $\hat{f}(\mathbf{x})$ e $\hat{g}(\mathbf{x})$. Para tal, o controlador ReGFL faz uso de regras fuzzy semelhantes a (1), tal que:

SE \mathbf{z}^k **É** A^i **ENTÃO** $\hat{f}_i(\mathbf{z}^k) = \gamma_f^{i,k} \xi^k$ **E** $\hat{g}_i(\mathbf{z}^k) = \gamma_g^{i,k} \xi^k$ em que $\mathbf{z}^k = [\mathbf{x}^k \ e_1^k]$ é composto pelo vetor de estado do sistema, \mathbf{x}^k , e pelo erro de seguimento (ou regulação), e_1^k , no instante k . A^i representa os antecedentes da i -ésima regra fuzzy, $\gamma_f^{i,k}$ e $\gamma_g^{i,k}$ são os vetores dos consequentes dos modelos locais de cada regra fuzzy e $\xi^k = [\mathbf{z}^k \ 1]$. Nessa abordagem os consequentes dos modelos são atualizados pelo método dos mínimos quadrados recursivos com fator de esquecimento, tal que:

$$\begin{aligned} \Upsilon_{f_i}^k &= \frac{\Phi_{f_i}^{k-1} \xi^k}{\xi^{kT} \Phi_{f_i}^{k-1} \xi^k + \zeta} \\ \gamma_{f_i}^k &= \gamma_{f_i}^{k-1} + \Upsilon_{f_i}^k \left[(K_p e_1^k + K_i \int_{j=0}^k e_1^j) - \xi^{kT} \gamma_{f_i}^{k-1} \right] \\ \Phi_{f_i}^k &= \frac{1}{\zeta} \left[\Phi_{f_i}^{k-1} - \frac{\Phi_{f_i}^{k-1} \xi^k \xi^{kT} (k) \Phi_{f_i}^{k-1}}{\xi^{kT} \Phi_{f_i}^{k-1} \xi^k + \zeta} \right] \end{aligned} \quad (11)$$

e

$$\begin{aligned} \Upsilon_{g_i}^k &= \frac{\Phi_{g_i}^{k-1} \xi^k}{\xi^{kT} \Phi_{g_i}^{k-1} \xi^k + \zeta} \\ \gamma_{g_i}^k &= \gamma_{g_i}^{k-1} + \Upsilon_{g_i}^k \left[(K_i \int_{j=0}^k e_1^j) - \xi^{kT} \gamma_{g_i}^{k-1} \right] \\ \Phi_{g_i}^k &= \frac{1}{\zeta} \left[\Phi_{g_i}^{k-1} - \frac{\Phi_{g_i}^{k-1} \xi^k \xi^{kT} (k) \Phi_{g_i}^{k-1}}{\xi^{kT} \Phi_{g_i}^{k-1} \xi^k + \zeta} \right] \end{aligned} \quad (12)$$

em que $\Upsilon_{q_i}^k$ é o ganho de Kalman para a i -ésima regra no passo k , $q = f$ ou g , $\Phi_{q_i}^k$ é a matriz de covariância e ζ o fator de esquecimento.

Por fim, a saída do algoritmo é dada pela média ponderada dos modelos locais, como se segue,

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{c^k} \frac{\mu_i^k \hat{f}_i(\mathbf{z}^k)}{\sum_{i=1}^{c^k} \mu_i^k} \text{ e } \hat{g}(\mathbf{x}) = \sum_{i=1}^{c^k} \frac{\mu_i^k \hat{g}_i(\mathbf{z}^k)}{\sum_{i=1}^{c^k} \mu_i^k}. \quad (13)$$

Baseado no princípio da certeza equivalente (Wang, 1996), a lei de controle (9) pode ser reescrita como:

$$u = \frac{1}{\hat{g}(\mathbf{x})} [v - \hat{f}(\mathbf{x})]. \quad (14)$$

Aplicando (14) em (8), obtém-se

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + B(v + \delta_w) \quad (15)$$

em que $\delta_w = [f(\mathbf{x}) - \hat{f}(\mathbf{x})] + [g(\mathbf{x}) - \hat{g}(\mathbf{x})]u$ é uma perturbação na malha fechada do sistema, oriunda do erro de estimação pelo algoritmo ePL. A topologia do controlador ReGFL associado a lei de controle linear por realimentação de estados, $v = r^{(n)} - \mathbf{K}\mathbf{e}$ para seguimento de trajetória é apresentada na Figura 1, e suas etapas descritas pelo pseudo-código apresentado no Algoritmo 1.

Algoritmo 1: Pseudo-Código do controlador ReGFL.

Entrada: $\mathbf{z}(k) = [\mathbf{x}(k) \ e_1(k)] \in [0,1]^p$, $k = 1, \dots$

Saída: $\hat{f}(\mathbf{x})$, $\hat{g}(\mathbf{x})$

Escolha os parâmetros: $\alpha, \beta, \tau, \lambda, r$ e $\zeta \in [0,1]$;

Escolha os ganhos: K_p, K_{i_f}, K_{i_g} ;

enquanto $k \leq \infty$ **faça**

 Leia novo dado: $\mathbf{z}^k \leftarrow [\mathbf{x}^k \ e_1^k]$;

 Atualize: $\xi^k = [\mathbf{z}^k \ 1]$;

para $i = 1$ **até** c^k **faça**

 Calcule a_i^{k+1} e ρ_i^k usando (4);

fim

se $\text{argmin}_{j=1, \dots, c^k} a_i^{k+1} \geq \tau$ **então**

 Cria-se um novo grupo: $\mathbf{v}_i^{c^k+1} \leftarrow \mathbf{z}^k$;

fim

senão

 Atualize a regra mais compatível \mathbf{v}_s^{k+1} com (5);

 Atualize os parâmetros dos modelos locais $\gamma_f^{i,k}$
 e $\gamma_g^{i,k}$ usando (11) e (12);

fim

para $i = 1$ **até** $c^k - 1$ **e** $j = i + 1$ **até** c^k **faça**

 Calcule a compatibilidade entre as regras ρ_{ij}^k
 usando (6);

fim

se $\rho_{ij}^k \geq \lambda$ **então**

 Exclua o grupo \mathbf{v}_j^k ;

fim

para $i = 1$ **até** c^k **faça**

 Calcule o grau de ativação da regra usando (2);

fim

 Calcule $\hat{f}(\mathbf{x})$ e $\hat{g}(\mathbf{x})$ usando (13);

 Calcule o sinal de controle, u^k , usando (14);

retorna u^k ;

fim

2.3 Algoritmo de Evolução Diferencial

O algoritmo de evolução diferencial (*differential evolution algorithm* – DE) é um método estocástico de busca direta da classe de algoritmos evolucionários, introduzido no final da década de 1990 por Storn and Price (1997), que apresenta as seguintes etapas de execução:

- (1) *Iniciação da População:* essa etapa consiste em realizar-se uma distribuição uniforme randômica por todo o espaço de busca, tal que:

$$x_{i,j,0} = \text{rand}_j(0,1)(b_{j,U} - b_{j,L}) + b_{j,L} \quad (16)$$

em que $x_{i,j,0}$ são os indivíduos da população inicial (geração zero) $X_{i,0} = [x_{0,0} \ x_{1,0} \ \dots \ x_{N_p-1,0}]$, $i = 0, 1, \dots, N_p - 1$ e $j = 0, 1, \dots, D - 1$ são os índices do processo de evolução, N_p denota o tamanho da população e D a dimensão do espaço de busca. As notações $b_{j,U}$ e $b_{j,L}$ indicam os limites superior e inferior do espaço de busca, respectivamente.

- (2) *Mutação:* nessa etapa uma nova população $V_{i,g} = [v_{0,g} \ v_{1,g} \ \dots \ v_{N_p-1,g}]$ é criada baseando-se na diferença vetorial, como segue (Mezura-Montes et al., 2006),

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (17)$$

em que os vetores $x_{rt,g}$ para $t = 0, 1, 2$ são sorteados entre os membros da população e $F \in [0, 2]$ é o fator de escalonamento.

- (3) *Diversificação da População:* a recombinação ou diversificação da população é dada por

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{se } \text{rand}_j(0,1) \leq C_r \\ x_{j,i,g}, & \text{caso contrário.} \end{cases} \quad (18)$$

Nesse procedimento os indivíduos da população $X_{i,g}$ são recombinaados com indivíduos da população mutante $V_{i,g}$, o que resulta em um nova população $U_{i,g}$ de acordo com a probabilidade de combinação C_r .

- (4) *Seleção:* nessa etapa utiliza-se como critério de seleção a busca do tipo *one-to-one*. Desse modo, os indivíduos da população $U_{i,g}$ são comparados com os indivíduos da população $X_{i,g}$ por meio da função custo $h(\cdot)$, tal que:

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{se } h(u_{i,g}) \leq h(x_{i,g}) \\ x_{i,g}, & \text{caso contrário.} \end{cases} \quad (19)$$

Por fim, o ciclo de busca pelo algoritmo DE é finalizado quando satisfeito um dos critérios de parada: ter alcançado um erro de busca menor que determinada tolerância especificada ou atingido o número máximo de iterações, n_{max} .

3. FORMULAÇÃO DO PROBLEMA

O controlador ReGFL possui como principal característica a inclusão de robustez (Oliveira et al., 2020) e adaptabilidade (Oliveira et al., 2019a,b) ao método de linearização por realimentação. Contudo, conforme introduzido nas seções 2.1 e 2.2, essa abordagem apresenta parâmetros de operação passíveis de definição pelo usuário, como por exemplo, a taxa de aprendizagem (α), a taxa do índice de alerta (β), os limiares de alerta e compatibilidade (τ, λ), a zona de influência da Gaussiana (r), os ganhos K_p, K_{i_f}, K_{i_g} de atualização dos consequentes dos modelos locais em (11) e (12). É importante ressaltar que há na literatura (Pedrycz and Gomide, 2007; Lima et al., 2010; Lughofer, 2011) indicações das regiões para busca dos parâmetros do ePL. Porém, essas regiões são amplas e com múltiplas combinações, o que pode tornar trabalhosa a sintonia do controlador ReGFL em aplicações reais. A principal contribuição deste trabalho consiste na utilização do DE para a busca desses parâmetros.

Os parâmetros de sintonia do ePL podem ser divididos em dois grupos distintos, o primeiro grupo composto pelos limiares de alerta e compatibilidade (τ, λ), juntamente com a dispersão da Gaussiana (r). Esses parâmetros estão diretamente relacionados entre si e estão relacionados a ocupação do hiper cubo que contém as coordenadas dos centros dos grupos, $\mathbf{v}_i^k \in [0,1] \subseteq \mathbb{R}^p$ para $i = 1, \dots, c^k$. Desse modo, podem ser facilmente obtidos a partir da análise ocupacional do hiper cubo conforme apresentado em (Lima et al., 2010; Lughofer, 2011). Os demais parâmetros compõem o segundo grupo de variáveis e definem a dinâmica de agrupamento e atualização dos modelos locais do controlador ReGFL, que devem estar relacionados à dinâmica do processo a ser controlado.

Neste trabalho é proposto uma metodologia para sintonia dos parâmetros ($\alpha, \beta, K_p, K_{i_f}, K_{i_g}$) por meio do uso do algoritmo DE. Nessa abordagem, busca-se otimizar o

índice de desempenho do sistema dado pela integral do módulo do erro (IAE),

$$\text{IAE} = \int_0^t |e(t)| dt \quad (20)$$

que resulta em uma malha fechada com desempenho otimizado. Nesse caso, é proposta a solução do seguinte procedimento de otimização:

$$\begin{aligned} \text{Minimizar: } & \text{IAE} \\ \text{Sujeito a: } & \alpha, \beta, K_p, K_{i_f}, K_{i_g} \\ & \text{e } \alpha, \beta \in [0,1] \text{ e } K_p, K_{i_f}, K_{i_g} \geq 0 \end{aligned} \quad (21)$$

O Algoritmo 2 apresenta o pseudo-código do algoritmo DE desenvolvido para solução do problema de otimização (21).

Algoritmo 2: Pseudo-código para busca dos parâmetros ótimos do controlador ReGFL.

Entrada: População inicial: $X_{i,0}$;
Saída: Parâmetros ótimos: $\alpha, \beta, K_p, K_{i_f}, K_{i_g}$;
 Escolha os parâmetros do ReGFL: τ, λ, ζ e $r \in [0,1]$;
 Escolha os parâmetros do DE: D, F, C_r, N_p, b_U, b_L ,
 tolerância e estratégia;
 Inicie a população inicial, $X_{i,0}$, usando (16);
 Inicie os critérios de parada: $g = 0$ e $\text{erro} = 0$;

para $i = 1$ até N_p **faça**

Simule o sistema não linear (8) com o controlador ReGFL usando o Algoritmo 1 e lei de controle (14) para os indivíduos da população $X_{i,0}$;
 Calcule o índice IAE para cada indivíduo da população $X_{i,0}$ usando (20);

fim

enquanto $g \leq n_{max}$ ou $\text{erro} \leq \text{tolerância}$ **faça**

Crie uma população mutante, $V_{i,g}$, usando (17);
 Crie a população diversificada, $U_{i,g}$ usando (18);

para $i = 1$ até N_p **faça**

Simule o sistema não linear (8) com o controlador ReGFL usando o Algoritmo 1 e lei de controle (14) para os indivíduos da população $U_{i,g}$;
 Calcule o índice IAE para cada indivíduo da população $U_{i,g}$ usando (20);

fim

Faça a seleção dos indivíduos da próxima geração usando (19);

se $g = n_{max}$ ou $\text{erro} = \text{tolerância}$ **então**

Pare a busca e retorne os parâmetros ótimos.

fim

$g = g + 1$;

fim

Nota-se a partir do pseudocódigo apresentando no Algoritmo 2, que tanto o DE quanto o ePL possuem um elevado número de parâmetros definidos pelo usuário. Contudo, uma escolha subótima dos parâmetros para o DE pode resultar em valores que faça com que o ePL não alcance o valor otimizado para a função objetivo (21). Porém, a estabilidade da malha de controle é assegurada. Por outro lado, uma escolha inadequada dos parâmetros para o ePL pode resultar em perda de desempenho do controlador ou mesmo proporcionar uma condição em que leve a instabilidade da malha de controle. Assim, o procedimento aqui proposto resulta em economia de tempo e preserva a

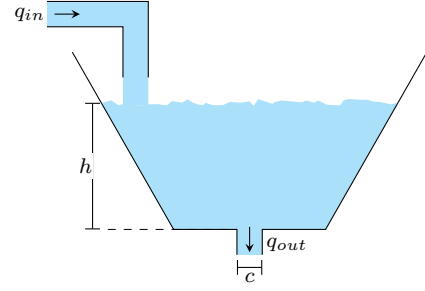


Figura 2. Controle de nível em um sistema de tanque.

estabilidade da malha fechada, reduzindo a sensibilidade da escolha de parâmetros por parte do usuário.

4. RESULTADOS

Nesta seção a metodologia de sintonia dos parâmetros do controlador ReGFL proposta na Seção 3 é avaliada em um sistema de controle de nível em um tanque não linear como o apresentado na Figura 2. O controle de nível em tanques, tem sido um *benchmark* usado para avaliar o desempenho de diversas técnicas de controle (Wang, 1996; Banerjee et al., 2011; Oliveira et al., 2019a,b). O tanque apresentado na Figura 2 possui dinâmica determinada por (Wang, 1996)

$$h = \frac{-c\sqrt{2gh}}{A(h)} + \frac{1}{A(h)}u \quad (22)$$

em que h é o nível da coluna de líquido (m), g é a constante gravitacional (m/s^2), c é a área de seção transversal do tubo de saída (m^2), $u = q_{in}$ é a vazão de entrada do sistema (m^3/s), $A(h)$ é a área da seção transversal do tanque (m^2), dada por: $A(h) = 0,01h + 0,2$. A simulação do sistema é realizada a partir do modelo discretizado a partir da aproximação de Euler:

$$h^{k+1} = h^k + T \left[-\frac{c\sqrt{2gh^k}}{A(h^k)} + \frac{1}{A(h^k)}u^k \right] \quad (23)$$

em que T é o período de amostragem. As condições de simulação, são as mesmas adotadas em (Oliveira et al., 2019a), com $T = 0.1s$, $\tau = 0.0075$, $\lambda = 0.85$, $r = 0.25$ e $\zeta = 0.98$. Note que, o período de amostragem é suficientemente pequeno em relação a constante de tempo do sistema (22) o que nos possibilita a implementação da lei de controle (14) descrita em tempo contínuo. Além disso, os parâmetros do ePL (τ, λ, r) foram definidos conforme metodologia apresentada em (Lima et al., 2010; Lughofer, 2011). Conforme a topologia de controle apresentada na Figura 1, adotou-se a lei de controle linear dada por realimentação de estados, com $\mathbf{K} = 1,25$ e saturação do sinal de controle em $\pm 50 m^3/s$, conforme sugerido em (Oliveira et al., 2019a).

Por outro lado, os parâmetros ($\alpha, \beta, K_p, K_{i_f}, K_{i_g}$) que determinam a dinâmica do controlador ReGFL foram sintonizados com auxílio do algoritmo DE, o qual foi configurado com os seguintes valores: $C_r = 0,80$ e $F = rand(0,5; 1)$, tal que a cada passo o fator de escalonamento é definido randonicamente no intervalo, de modo a proporcionar o aumento do raio de busca e reduzir a probabilidade de ocorrência de uma convergência precoce

(Oliveira et al., 2015; Chakraborty, 2008). Além disso, adotou-se o seguinte número de indivíduos da população $N_p = 50$, estratégia de mutação $DE/best/1$ (Mezura-Montes et al., 2006) e limites superior ($b_U = 1$) e inferior $b_L = 0$ para os todos parâmetros de otimização presente em (21). Por fim, estabeleceu-se como critério de parada o número máximo de iterações, $n_{max} = 50000$, ou tolerância máxima de erro da ordem de 10^{-7} .

A sintonia dos parâmetros do controlador se deu para três formas de onda distintas: quadrada, dente de serra e triangular como sugerido em (Oliveira et al., 2019a). Além disso, a busca com o DE foi repetida 20 vezes para cada forma de onda e os valores médios dos parâmetros encontram-se apresentados na Tabela 1. Esses valores foram obtidos com um número médio de iterações (ω_m) pelo DE de: onda quadrada: $\omega_m = 473$ iterações, dente de serra: $\omega_m = 501$ iterações e triangular de: $\omega_m = 216$ iterações.

Tabela 1. Parâmetros obtidos com o algoritmo DE para o controlador ReGFL.

F. Onda	α	β	K_p	Ki_f	Ki_g
Quadrada	0,3721	$2,712e^{-4}$	0,3183	0,0069	0,1323
D. Serra	0,4897	$8,256e^{-4}$	0,3482	0,0128	0,1883
Triangular	0,9043	$6,954e^{-5}$	0,0256	0,0147	0,7546

A partir dos valores apresentados na Tabela 1 o controlador ReGFL foi avaliado para 50 sistemas aleatórios, em que a área de seção transversal do tudo de saída é dada por: $c = 0,25 - \sigma$ (m^2) em que $\sigma = rand(0;0,5)$. O desempenho da malha fechada de controle sintonizada com o algoritmo DE foi comparada com a linearização por realimentação exata — LRE (Khalil, 2002) e com a sintonia apresentada para o controlador ReGFL em (Oliveira et al., 2019a) realizada a partir da região de busca definida em (Lima et al., 2010; Lughofer, 2011). Nessa abordagem, os parâmetros adotados para o controlador foram: $\alpha = 0,005$, $\beta = 1,25e^{-4}$, $K_p = 0,55$, $Ki_f = 0,01$ e $Ki_g = 0,04$. O resultado dessa simulação para a onda quadrada com $\sigma = 0,5$ é apresentada na Figura 4.

Além disso, o desempenho das malhas de controle em estudo foram avaliadas com auxílio dos seguintes índices de desempenho: integral do erro absoluto multiplicado pelo tempo — ITAE = $\int_0^{t_f} t \| e(t) \| dt$, integral da variabilidade ponderada do sinal de controle — IVU = $\sqrt{\frac{1}{t_f} \int_0^{t_f} |u(t) - \bar{u}|^2}$, em que \bar{u} é o sinal de controle médio, e integral da variabilidade ponderado do erro — IVE = $\sqrt{\frac{1}{t_f} \int_0^{t_f} |e(t) - \bar{e}|^2}$, com \bar{e} sendo o erro médio. Os resultados encontram-se apresentados na Tabela 2, em que os valores foram normalizados em relação aos respectivos índices obtido pelo controlador ReGFL com valores de sintonia proposta em (Oliveira et al., 2019a), ou seja, $I_{nx} = \frac{I_x}{I_{ReGFL}}$, em que I_{nx} é o índice normalizado e I_x o índice a ser normalizado.

Conforme pode-se verificar na Tabela 2 o controlador ReGFL apresentou desempenho superior ao método LRE, resultado esse já esperado. Uma vez que o controlador por realimentação de estados não consegue notar o erro de modelagem do sistema, e consequentemente não atua

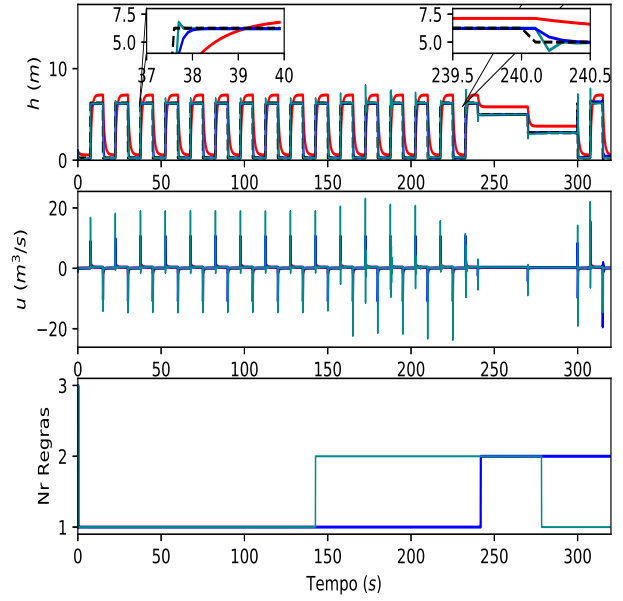


Figura 3. Desempenho do sistema para a referência dada por uma onda quadrada ($\sigma = 0,5$), em que as técnicas de controle estão representadas por: linha contínua vermelha — LRE, azul — ReGFL sintonia (Oliveira et al., 2019a), ciana — ReGFL sintonia DE e linha pontilhada preta a referência.

em ordem de mitigar seus efeitos. É importante observar, que o controlador ReGFL foi iniciado sem nenhum conhecimento prévio do sistema, ou seja, assumiu-se que as funções $f(\mathbf{x})$ e $g(\mathbf{x})$ são completamente desconhecidas. Por outro lado, o controlador LRE é inicializado a partir de um conhecimento prévio da dinâmica do sistema. Além disso, nota-se que o uso do algoritmo DE para sintonia dos parâmetros do controlador ReGFL proporcionou uma significativa melhora no desempenho em malha fechada do sistema controlado. Fato esse, que foi obtido sem exigir aumento no custo computacional pelo controlador, visto que o número de regras manteve-se a mesma para ambos métodos de sintonia. Por exemplo, para a forma de onda triangular o IAE foi reduzido de 46,34 para 16,74, o que representa um aumento no desempenho da ordem de 63,87% em relação a sintonia dos parâmetros realizada por escolha do usuário com sugerido em (Lima et al., 2010; Lughofer, 2011). Pretende-se continuar este trabalho com a realização de testes experimentais e também na otimização das simulações realizadas com o Algoritmo 2.

5. CONCLUSÃO

Neste trabalho foi apresentado uma nova abordagem para sintonia dos parâmetros do controlador ReGFL por meio do uso do algoritmo DE. Na metodologia proposta, o algoritmo DE busca otimizar o índice de desempenho IAE para a malha fechada de controle, através da busca dos parâmetros ótimos para controlador ReGFL. O controle de nível em um tanque não linear, foi empregado na avaliação da abordagem proposta. O resultado foi quantificado com auxílio dos índices de desempenho IAE, ITAE, IVE e IVU

Tabela 2. Resultado dos índices de desempenho da malha fechada normalizados pelos valores obtidos pelo ReGFL que foi considerado unitário em cada critério.

Entrada	Controlador	σ	IAE	ITAE	IVE	IVU	# Regras
Quadrada	ReGFL(DE)	0,50	0.76	1.13	1.02	5.03	3
		média	0.99	1.02	1.01	2.39	3
	LRE	0,50	6.79	7.57	0.34	1.27	-
		média	5.17	5.59	0.38	0.77	-
Triangular	ReGFL(DE)	0,50	0.36	0.36	0.48	2.34	3
		média	0.60	0.79	2.63	12.4	3
	LRE	0,50	5.06	6.00	3.5	5.07	-
		média	5.16	5.91	4.70	1.64	-
Dente de Serra	ReGFL(DE)	0,50	0.72	0.76	6.34	7.09	3
		média	0.73	0.72	7.10	14.71	3
	LRE	0,50	4.49	4.96	4.83	0.71	-
		média	0.72	0.71	7.01	1.77	-

e comparado com o controlador ReGFL sintonizado por (Oliveira et al., 2019a) e com a linearização por realimentação exata (Khalil, 2002). A análise dos índices de desempenho nos possibilita concluir que o sintonia do controlador via algoritmo DE proporciona um desempenho superior da malha fechada de controle em relação aos demais métodos de controle/sintonia analisados neste trabalho.

AGRADECIMENTOS

Agradecemos ao CEFET-MG pelos recursos disponibilizados para desenvolvimento do presente trabalho.

REFERÊNCIAS

- Aguirre, L.A. (2007). *Introdução à Identificação de Sistemas: Técnica Lineares e Não Lineares Aplicadas a Sistemas Reais*. Editora UFMG, Belo Horizonte, MG, 3 edition.
- Angelov, P. (2004). A fuzzy controller with evolving structure. *Information Sciences*, 161(1), 21 – 35. doi:10.1016/j.ins.2003.03.006.
- Banerjee, S., Chakrabarty, A., Maity, S., and Chatterjee, A. (2011). Feedback linearizing indirect adaptive fuzzy control with foraging based on-line plant model estimation. *Applied Soft Computing*, 11(4), 3441 – 3450. doi:https://doi.org/10.1016/j.asoc.2011.01.016.
- Chakraborty, U.K. (2008). *Advances in Differential Evolution*. Springer-Verlag, Berlin, Heidelberg, 1st edition.
- Khalil, H.K. (2002). *Nonlinear systems*. Prentice-Hall, Upper Saddle River, NJ, 3th edition.
- Leite, D., Škrjanc, I., and Gomide, F. (2020). An overview on evolving systems and learning from stream data. *Evolving Systems*, 11, 181 – 198. doi:10.1007/s12530-020-09334-5.
- Lima, E., Gomide, F., and Ballini, R. (2006). Participatory evolving fuzzy modeling. In *2006 International Symposium on Evolving Fuzzy Systems*, 36–41. doi:10.1109/ISEFS.2006.251135.
- Lima, E., Hell, M., Ballini, R., and Gomide, F. (2010). *Evolving Fuzzy Modeling Using Participatory Learning*, 67–86. Wiley-IEEE Press, 1st edition.
- Lughofer, E. (2011). *Evolving Fuzzy Systems: Methodologies, Advanced Concepts and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition. doi:10.1007/978-3-642-18087-3.
- Mezura-Montes, E., Velázquez-Reyes, J., and Coello Coello, C.A. (2006). A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, 485–492. Association for Computing Machinery, New York, NY, USA. doi:10.1145/1143997.1144086.
- Oliveira, L., Bento, A., Leite, V., and Gomide, F. (2019a). Robust evolving granular feedback linearization. In R.B. Kearfott, I. Batyrshin, M. Reformat, M. Ceberio, and V. Kreinovich (eds.), *Fuzzy Techniques: Theory and Applications*, 442–452. Springer International Publishing, Cham. doi:10.1007/978-3-030-21920-8.
- Oliveira, L., Bento, A., Leite, V.J., and Gomide, F. (2020). Evolving granular feedback linearization: Design, analysis, and applications. *Applied Soft Computing*, 86, 1–12. doi:10.1016/j.asoc.2019.105927.
- Oliveira, L., Franco, A., and Leite, V. (2015). Estratégia para síntese do ganho da malha de controle robusto em sistemas com realimentação linearizante via algoritmo diferencial evolutivo. In *Simpósio Brasileiro de Automação Inteligente*, 1824 – 1829. Natal, RN.
- Oliveira, L., Leite, V., Bento, A., and Gomide, F. (2019b). Robust granular feedback linearization. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–6. doi:10.1109/FUZZ-IEEE.2019.8858974.
- Pedrycz, W. and Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Wiley-IEEE Press, Hoboken, New Jersey, 1st edition.
- Silva, A.M., Caminhas, W.M., Lemos, A.P., and Gomide, F. (2013). Evolving neo-fuzzy neural network with adaptive feature selection. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, 341–349. doi:10.1109/BRICS-CCI-CBIC.2013.64.
- Silva, J., Oliveira, L., Gomide, F., and Leite, V. (2018). Avaliação experimental da linearização por realimentação granular evolutiva. In *Proceedings of the 5th Brazilian Conference on Fuzzy Systems*, 359–370. Fortaleza, CE, Brazil.
- Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341–359. doi:doi.org/10.1023/A:1008202821328.
- Wang, L. (1996). Stable adaptive fuzzy controllers with application to inverted pendulum tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(5), 677–691. doi:10.1109/3477.537311.