

Desenvolvimento de um controlador $PI^\lambda D^\mu$ discreto otimizado e embarcado em incubadora neonatal

Igor R. Sousa

Pós-Graduação em Eng. de Teleinformática(PPGETI)
Universidade Federal do Ceará (UFC)
 Campus do Pici, Centro de Tecnologia
 igor.sousa@alu.ufc.br

Guilherme A. Barreto

Pós-Graduação em Eng. de Teleinformática(PPGETI)
Universidade Federal do Ceará (UFC)
 Campus do Pici, Centro de Tecnologia
 gbarreto@ufc.br

Resumo—Neste trabalho, é apresentado o projeto e implementação de um controlador PID de ordem fracionária, também conhecido como $PI^\lambda D^\mu$, em um sistema embarcado com microcontrolador de 8-bits para controle de temperatura de uma incubadora neonatal. A partir de dados reais, é levantada a função de transferência da planta. Os parâmetros k_p , k_i , k_d , λ e μ do controlador fracionário foram obtidos através da metaheurística de otimização PSO. Para evitar a perda de otimalidade que acontece ao se obter os ganhos do controlador analógico e posteriormente convertendo para uma versão discreta e aproximada, o sistema de controle utilizado no processo de otimização possui as limitações reais existentes em um microcontrolador. São aplicadas diferentes funções objetivo, e os resultados dos controladores obtidos com essas funções são convergentes. O controlador de ordem fracionária otimizado obteve 6.468s de tempo de acomodação e sobressinal de 1,18%, enquanto que no sistema real, este controlador obteve 13.668qs de tempo de acomodação e sobressinal de 3,32%.

Index Terms—identificação recursiva, otimização de enxame, controlador de ordem fracionária

I. INTRODUÇÃO

Os controladores PID de ordem fracionária (FOPID, acrônimo do termo em inglês) também conhecidos pela sigla $PI^\lambda D^\mu$, se baseiam no cálculo fracionário. A função de transferência do controlador $PI^\lambda D^\mu$ é definida por

$$C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu, \quad (1)$$

em que λ e μ são as ordens fracionárias de integração e derivação, respectivamente, enquanto que $U(s)$ e $E(s)$ são a saída do controlador e o sinal de erro do sistema de controle, respectivamente. Nota-se que os controladores PID convencionais são casos particulares do controlador $PI^\lambda D^\mu$, em que $\lambda = \mu = 1$. Assim, além de parametrizar os ganhos k_p , k_i e k_d , é necessário parametrizar a ordem de integração λ e de derivação μ , conferindo graus de liberdade adicionais ao modelo do controlador, o que pode vir a melhorar seu desempenho no controle de sistemas dinâmicos complexos [1].

Este trabalho conta com o apoio financeiro da CAPES (finance code 001), CNPq (No. 309379/2019-9) e do Centro de Referência em Inteligência Artificial - CEREA, convênios 23067.055633/2021-19 (UFC) / 2020/09706-7 (FAPESP).

De acordo com Jamil *et al.* [2], atualmente, controladores $PI^\lambda D^\mu$ vêm sendo utilizados com sucesso em um número crescente de aplicações, uma vez que podem alcançar melhores desempenhos que o controladores PID clássicos. Em controle de temperatura, pode-se citar o trabalho de Feliu-Batlle *et al.* [3], que realizam o controle de um forno de reaquecimento com um controlador $PI^\lambda D^\mu$, apresentando melhores resultados que o PID convencional. Já Petráš e Vinagre [4] realizaram o controle de um bloco de aquecimento por controlador $PI^\lambda D^\mu$ e PID, em que o $PI^\lambda D^\mu$ também demonstrou melhor desempenho.

Segundo Shah e Agashe [1], as técnicas de sintoniza de controladores $PI^\lambda D^\mu$ podem ser divididas em três tipos: métodos baseados em regras, como as de Ziegler-Nichols para $PI^\lambda D^\mu$ proposta por Barbosa *et al.* [5], os métodos analíticos, como a aplicação de funções ortogonais por partes utilizada em Bouafoura e Braiek [6] e os métodos de otimização numérica.

Entre os métodos numéricos de otimização utilizados em sintonia de controladores $PI^\lambda D^\mu$, pode-se citar a aplicação de algoritmos genéticos em Cao *et al.* [7], evolução diferencial em Abraham *et al.* [8], colônia de abelhas em Rajasekhar *et al.* [9] e enxame de partículas em Karimi-Ghartemani *et al.* [10].

A otimização por enxame de partículas (*particle swarm optimization* - PSO) tornou-se um dos métodos de inteligência computacional mais eficazes em resolver problemas de otimização global. Este método provou ser uma poderosa ferramenta na resolução de problemas com não-linearidade e não-diferenciabilidade, múltiplos ótimos e alta dimensionalidade. As vantagens do PSO são sua relativa simplicidade e convergência estável com boa eficiência computacional [11].

Um dos maiores problemas que surge na implementação de controladores $PI^\lambda D^\mu$ reside justamente em sua teoria matemática subjacente, ou seja, o cálculo fracionário, que é bem mais complexo que o cálculo diferencial e integral de ordem inteira [12]. No entanto, com o passar do tempo, vários pesquisadores encontraram diferentes soluções para superar tais problemas, identificando e aproximando interpretações geométricas e físicas [12]. É importante mencionar que os controladores de ordem fracionária são filtros lineares de dimensão infinita e todos os métodos de implementação existentes são aproximações de dimensão finita [13].

Entre as formas existentes de implementação, destaca-se o controlador $PI^\lambda D^\mu$ discreto de memória longa (LD- $PI^\lambda D^\mu$, na sigla em inglês), proposta por Merrikh-Bayat *et al.* [14], que consiste na discretização da variável contínua s em (1) para z utilizando a transformação de Tustin e então realizando uma expansão em série de potência, que é truncada devido à limitação de memória.

É bastante comum que primeiro seja projetado o controlador $PI^\lambda D^\mu$ contínuo em (1) e posteriormente seja sintetizada uma versão discreta aproximada. Porém, a otimalidade alcançada durante a fase de projeto é perdida após essa aproximação, uma vez que, na prática, qualquer sistema discreto possui limitações de memória. De fato, um controlador ótimo projetado no tempo contínuo (i.e., o controlador analógico) não será mais ótimo após a aproximação no tempo discreto [14].

Do exposto, neste trabalho é realizado o controle PID de ordem fracionária de temperatura de uma incubadora neonatal. Para isso, é feita a identificação da incubadora pelo algoritmo de mínimos recursivos e, posteriormente, é projetado um controlador $PI^\lambda D^\mu$ utilizando PSO para otimização dos seus cinco parâmetros, a saber $\{k_p, k_i, k_d, \lambda, \mu\}$. Porém, para não perder a otimalidade alcançada via PSO, o sistema de controle otimizado consiste já na forma aproximada do controlador $PI^\lambda D^\mu$ para tempo discreto, e não pelo controlador $PI^\lambda D^\mu$ no tempo contínuo. Posteriormente, a aproximação do controlador obtido é implementada em um microcontrolador de 8 *bits* para realizar o controle real da incubadora.

O artigo está dividido da seguinte forma: na Seção II é apresentado o algoritmo utilizado para identificação da planta. Na Seção III é apresentada a teoria de cálculo fracionário aplicado em controladores PID. Já na Seção IV são apresentados os fundamentos do algoritmo de otimização PSO, enquanto que a planta utilizada é apresentada na Seção V. A metodologia utilizada neste trabalho é apresentada na Seção VI. Por fim, os resultados e conclusões estão dispostos nas Seções VII e VIII, respectivamente.

II. IDENTIFICAÇÃO VIA MÍNIMOS QUADRADOS RECURSIVOS

Neste trabalho, a incubadora neonatal é identificada por um modelo de tempo discreto ARX (*autoregressive with exogenous input*) de segunda ordem, com base em Sousa *et al.* [15], que realizaram a identificação LPV (*linear parameter-varying*) desta incubadora neonatal e obtiveram um modelo de segunda ordem como mais representativo. Um modelo ARX de segunda ordem é definido por

$$y[k] = b_0 u[k] + b_1 u[k-1] + b_2 u[k-2] - a_1 y[k-1] - a_2 y[k-2], \quad (2)$$

em que $u[k]$ e $y[k]$ denotam, respectivamente, a entrada e a saída do modelo no instante k . Os coeficientes $b_0, b_1, b_2, a_1, a_2 \in \mathbb{R}$ são os parâmetros do modelo a serem estimados. A Equação (2) pode ser escrita na forma de produto escalar como

$$y[k] = \boldsymbol{\theta}^T \mathbf{x}[k], \quad (3)$$

em que

$$\mathbf{x}[k] = [u[k] \ u[k-1] \ u[k-2] \ -y[k-1] \ -y[k-2]]^T \quad (4)$$

é o vetor de regressores e

$$\boldsymbol{\theta} = [b_0 \ b_1 \ b_2 \ a_1 \ a_2]^T \quad (5)$$

o vetor de coeficientes do modelo.

O vetor de parâmetros $\boldsymbol{\theta}$ pode ser estimado pelo método dos mínimos quadrados recursivos (*Recursive Least Square - RLS*) [16], que baseia-se na minimização da seguinte função custo:

$$J_{RLS}[k] = \sum_{k=0}^n \phi^{n-k} \epsilon^2[k] = \sum_{k=0}^n \phi^{n-k} \left\{ y[k] - \hat{\boldsymbol{\theta}}^T[k] \mathbf{x}[k] \right\}^2, \quad (6)$$

em que $0 \leq \phi \leq 1$ é o fator de esquecimento e $\epsilon[k]$ é o erro cometido entre o valor real e o estimado no instante k . Esta função custo é minimizada a cada iteração e o número de parcelas do somatório cresce em função de k . A regra de ajuste recursivo é escrita como

$$\hat{\boldsymbol{\theta}}[k+1] = \hat{\boldsymbol{\theta}}[k] + \mathbf{K} \epsilon[k], \quad (7)$$

em que $\mathbf{K}[k] \in \mathbb{R}^5$ é o ganho de Kalman, definido por

$$\mathbf{K}[k] = \frac{\mathbf{P}[k] \mathbf{x}[k]}{\phi + \mathbf{x}[k]^T \mathbf{P}[k] \mathbf{x}[k]}, \quad (8)$$

e $\mathbf{P}[k] \in \mathbb{S}_{++}^5$ é matriz definida por

$$\mathbf{P}[k+1] = \frac{1}{\phi} \left\{ \mathbf{P}[k] - \mathbf{K}[k] \mathbf{x}^T[k] \mathbf{P}[k] \right\}. \quad (9)$$

III. DERIVADAS E INTEGRAIS FRACIONÁRIAS

A definição de Grunwald-Letnikov para a operação integro-diferencial fracionária é

$${}_{t_0} \mathcal{D}_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[t-t_0/h]} (-1)^j \binom{\alpha}{j} f(t-jh), \quad (10)$$

em que

$$\binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j! \Gamma(\alpha-j+1)}. \quad (11)$$

O valor de α determina se a operação é uma derivada fracionária ($\alpha > 0$) ou uma integral fracionária ($\alpha < 0$). Os limites inferior e superior da operação são definidos respectivamente por t_0 e t , enquanto que $\Gamma(\cdot)$ é a função Gamma.

De acordo com Xue [17], se o passo de cálculo h for pequeno o suficiente, o limite em (10) pode ser removido e pode ser calculado aproximadamente como

$${}_{t_0} \mathcal{D}_t^\alpha f(t) \approx \frac{1}{h^\alpha} \sum_{j=0}^{[t-t_0/h]} (-1)^j \binom{\alpha}{j} f(t-jh). \quad (12)$$

A. Estabilidade de sistemas de ordem fracionária

Seja $Q(s)$ a equação característica de uma função de transferência de ordem fracionária, dada por

$$Q(s) = \sum_{i=1}^n \delta_i s^{\beta_i}, \quad (13)$$

em que $\delta_i, \beta_i \in \mathbb{R}_+$. Fazendo $\sigma = s^q$, (13) pode ser reescrita como

$$Q(\sigma) = \sum_{i=1}^n \delta_i \sigma^{\gamma_i}, \quad (14)$$

em que $\gamma_i \in \mathbb{Z}_+$. Assim, este expediente matemático permite criar uma nova função característica (no plano σ) de ordem inteira. Uma análise das raízes de (14) pode ser feita para avaliar a estabilidade do sistema. De acordo com o teorema de estabilidade de Matignon [18], uma função de transferência de ordem fracionária com equação característica $Q(\sigma)$ é estável se e somente se

$$|\arg(\sigma_i)| > q \frac{\pi}{2}. \quad (15)$$

A estabilidade no plano σ é ilustrada na Fig. 1. É interessante notar que para $q = 1$ em (15), a região de instabilidade é a mesma de sistemas de ordem inteira. É importante salientar que o teorema de estabilidade de Matignon, na expressão em (15) e Figura 1 referem-se ao plano $\sigma = s^q$, e não ao plano s . A região de estabilidade no plano s continua sendo o semi-plano esquerdo.

B. Aproximação discreta de controladores $PI^\lambda D^\mu$

A aproximação proposta por Merrikh-Bayat *et al.* [14] é baseada no mapeamento modificado de Tustin, que aplicado à derivada fracionária s^μ , resulta em

$$s^\mu = \left(\frac{\omega_c}{\tan(\omega_c T/2)} \cdot \frac{1-z^{-1}}{1+z^{-1}} \right)^\mu = \alpha^\mu \left(\frac{1-z^{-1}}{1+z^{-1}} \right)^\mu, \quad (16)$$

em que

$$\alpha \triangleq \omega_c / \tan(\omega_c T/2), \quad (17)$$

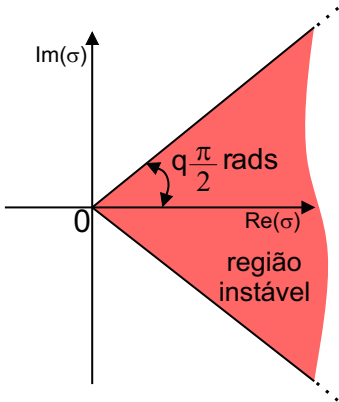


Fig. 1: Região de estabilidade de sistemas fracionários.

ω_c é o ganho de frequência cruzada da função de transferência de malha aberta e $\tan(\cdot)$ é a função trigonométrica tangente. A expansão em série de potência de (16) é dada por

$$\alpha^\mu \left(\frac{1-z^{-1}}{1+z^{-1}} \right)^\mu = \alpha^\mu \sum_{k=0}^{\infty} f_k(\mu) (z^{-1})^k, \quad (18)$$

em que

$$f_k(\mu) = \frac{1}{k!} \cdot \frac{d^k}{d(z^{-1})^k} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \Big|_{z^{-1}=0}. \quad (19)$$

Inserindo (16) e (18) em (1), com $s^{-\lambda} = (1/s)s^{1-\lambda}$, a expressão de aproximação do controlador $PI^\lambda D^\mu$ resulta em

$$C(z) = K_p + K_i \frac{1+z^{-1}}{1-z^{-1}} \sum_{k=0}^M f_k(1-\lambda) z^{-k} + K_d \sum_{k=0}^M f_k(\mu) z^{-k}, \quad (20)$$

em que

$$K_p = k_p, \quad K_i = k_i \alpha^{-\lambda}, \quad K_d = k_d \alpha^\mu, \quad (21)$$

e M é o tamanho da memória considerada na aproximação. Se (20) for desenvolvida, pode-se reescrevê-la como

$$C(z) = \frac{w_0 + w_1 z^{-1} + w_2 z^{-2} + \dots + w_{M+1} z^{-(M+1)}}{1 - z^{-1}}, \quad (22)$$

em que $w_0 \dots w_6$ são constantes que serão definidas na Seção VI que dependem dos ganhos do controlador (k_p, k_i, k_d), dos expoentes de derivação/integração (λ, μ) e de M .

Comentário 1. É importante ressaltar que k_p, k_i e k_d são os ganhos do controlador no domínio do tempo contínuo, ou seja, o controlador analógico apresentado em (1). Já K_p, K_i e K_d são as versões destes ganhos na aproximação discreta proposta por Merrikh-Bayat *et al.* [14].

IV. FUNDAMENTOS DO ALGORITMO PSO

A metaheurística de otimização PSO é um algoritmo baseado em população e sua busca pela solução ótima se dá em múltiplas direções pelas várias partículas, ao invés de uma única solução. O algoritmo é inspirado no comportamento social e na auto-organização de grupos de pássaros migratórios e de cardume de peixes [19]. Entre suas vantagens, o algoritmo PSO possui memória que retém as boas soluções alcançadas por todas as partículas. Além disso, o algoritmo tem cooperação entre as partículas do enxame, compartilhando informações [20].

Para a otimização em um espaço de busca D -dimensional, a posição da j -ésima partícula do enxame é definida como $\mathbf{p}_j = [p_{j1}, p_{j1}, \dots, p_{jD}]$. Na iteração k , a melhor posição encontrada pela partícula \mathbf{p}_j é definida como $\mathbf{p}_j^{best}[k]$, enquanto que a melhor posição encontrada pelo enxame como um todo é definida como $\mathbf{g}^{best}[k]$. Estas posições são utilizadas

na atualização das velocidades v_j e das posições de cada partícula, conforme

$$\zeta_1[k] = c_1 \psi_1[k] (\mathbf{p}_j^{best}[k] - \mathbf{p}_j[k])$$

$$\zeta_2[k] = c_2 \psi_2[k] (\mathbf{g}^{best}[k] - \mathbf{p}_j[k])$$

$$\mathbf{v}_j[k+1] = \varrho \{w[k] \mathbf{v}_j[k] + \zeta_1[k] + \zeta_2[k]\} \quad (23)$$

$$\mathbf{p}_j[k+1] = \mathbf{p}_j[k] + \mathbf{v}_j[k+1], \quad (24)$$

em que $c_1, c_2 \in \mathbb{R}_{>0}$ são os coeficientes constantes de aprendizagem cognitiva e social, respectivamente. Já ψ_1 e ψ_2 são variáveis aleatórias uniformemente distribuídas no intervalo $(0,1)$. O coeficiente ϱ é um fator de constrição. Por fim, $w[k]$ é o fator de peso inercial decrescente no instante k e visa evitar problemas relacionados à velocidade das partículas. Neste trabalho, este fator é definido como

$$w[k] = w_{final} + (w_{inicial} - w_{final}) \left(\frac{N_{it} - k}{N_{it} - 1} \right), \quad (25)$$

em que $w_{inicial}$ e w_{final} são, respectivamente, o valor inicial e final de w , enquanto que N_{it} é o número de iterações do algoritmo PSO.

Por fim, o desempenho de cada partícula $\mathbf{p}_j[k]$ e as posições $\mathbf{p}_j^{best}[k]$ e $\mathbf{g}^{best}[k]$ são avaliadas por meio de uma função objetivo. A solução ótima obtida depende da função objetivo estabelecida, que é fundamental no projeto.

Os índices de desempenho baseados em erro são bastante utilizados nas funções objetivo em otimização de parâmetros de sistemas de controle em malha fechada. O erro acumulado ao longo do tempo pode ser utilizado de diferentes formas, gerando diferentes índices de desempenho [21]. Destas, destaca-se a integral do erro quadrático multiplicado pelo tempo (*integral time square error - ITSE*), definida como

$$J_{ITSE} = \sum_{k=0}^N kT e^2[k], \quad (26)$$

em que o erro e é definido como a diferença entre o valor de referência e a resposta do sistema. Uma outra abordagem para índice de desempenho é apresentada em Aghababa [22], que utiliza critérios da resposta transitória (*RT*) do sistema de controle, como tempo de subida T_s , tempo de acomodação T_a , máximo sobressinal M_s e erro de estado permanente E_{ep} ,

$$J_{RT} = \frac{1}{1 + e^{-r}} (T_s + T_a) + \frac{e^{-r}}{1 + e^{-r}} (M_s + E_{ep}), \quad (27)$$

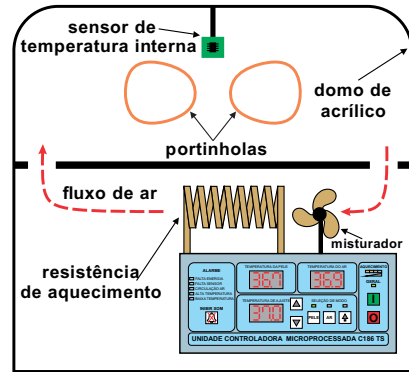
em que r é um fator de ponderação. Otimizações em sistemas de controle que utilizam J_{ITSE} e J_{RT} buscam minimizar tais índices.

V. DESCRIÇÃO DA PLANTA CONTROLADA

A planta utilizada neste trabalho é uma incubadora neonatal, modelo C186-TS da FANEM[®] ilustrada na Fig. 2. A incubadora utiliza uma resistência elétrica de aquecimento, que por sua vez é acionada por um TRIAC via ângulo de disparo. A saber, a variável controlada é a temperatura



(a) Bancada de testes.



(b) Desenho esquemático da planta térmica.

Fig. 2: Incubadora neonatal C186 TS, FANEM[®].

interna da incubadora, enquanto que a variável de controle é um valor numérico relacionado ao registrador do *timer* do microcontrolador que acarreta em um ângulo de disparo para acionamento do TRIAC.

Foram realizadas atualizações no *hardware* e *software* (i.e., *retrofit*) da planta para que fossem possível realizar ensaios de identificação e controle por meio de um microcontrolador de 8 *bits* externo. O microcontrolador utilizado é o ATMEGA328p da Microchip[®], que possui 2 KB de SRAM e 32 KB de memória de programa.

Mais detalhes sobre esse *retrofit*, realizado pelo mesmo grupo de pesquisa deste trabalho, podem ser encontrados em Sousa *et al.* [15]. Este trabalho trata apenas do controle de temperatura da incubadora neonatal.

VI. METODOLOGIA

Primeiramente, é realizada a identificação da planta para um modelo ARX de segunda ordem utilizando o algoritmo de estimação RLS apresentado na Seção II. O sinal de entrada utilizado é do tipo PRBS (*pseudorandom binary sequence*) [23]. Por fim, a entrada e saída da planta são adquiridos a uma taxa de 1.000s devido às constantes de tempo elevadas presentes na planta, com base em [15].

Como citado anteriormente, a otimização do controlador $PI^\lambda D^\mu$ é realizada com a utilização da aproximação discreta em (20). Desenvolvendo (20) para $M = 5$, obtêm-se (22) com

$$\begin{aligned}
 w_0 &= K_p + K_i + K_d \\
 w_1 &= -\{K_p + K_d(2\mu + 1) + K_i[2(1-\lambda) - 1]\} \\
 w_2 &= 2\{K_d(\mu^2 + \mu) + K_i[(1-\lambda)^2 - (1-\lambda)]\} \\
 w_3 &= -2\left\{K_d\left(\frac{2}{3}\mu^3 + \mu^2 + \frac{1}{3}\mu\right) + \right. \\
 &\quad \left. + K_i\left[\frac{2}{3}(1-\lambda)^3 - (1-\lambda)^2 + \frac{1}{3}(1-\lambda)\right]\right\} \\
 w_4 &= 2\left\{K_d\left(\frac{1}{3}\mu^4 + \frac{2}{3}\mu^3 + \frac{2}{3}\mu^2 + \frac{1}{3}\mu\right) + \right. \\
 &\quad \left. + K_i\left[\frac{1}{3}(1-\lambda)^4 - \frac{2}{3}(1-\lambda)^3 + \frac{2}{3}(1-\lambda)^2 - \frac{1}{3}(1-\lambda)\right]\right\} \\
 w_5 &= -2\left\{K_d\left(\frac{2}{15}\mu^5 + \frac{1}{3}\mu^4 + \frac{2}{3}\mu^3 + \frac{2}{3}\mu^2 + \frac{1}{5}\mu\right) + \right. \\
 &\quad \left. + K_i\left[\frac{2}{15}(1-\lambda)^5 - \frac{1}{3}(1-\lambda)^4 + \frac{2}{3}(1-\lambda)^3 + \right. \right. \\
 &\quad \left. \left. - \frac{2}{3}(1-\lambda)^2 + \frac{1}{5}(1-\lambda)\right]\right\} \\
 w_6 &= 2\left\{K_d\left(\frac{2}{15}\mu^5 + \frac{2}{3}\mu^3 + \frac{1}{5}\mu\right) + \right. \\
 &\quad \left. - K_i\left[\frac{2}{15}(1-\lambda)^5 + \frac{2}{3}(1-\lambda)^3 + \frac{1}{5}(1-\lambda)\right]\right\}. \tag{28}
 \end{aligned}$$

Comentário 2. As equações apresentadas em (28) foram desenvolvidas de próprio punho e posteriormente, para fins de conferência, foram confirmadas com o uso da *toolbox* de matemática simbólica no *software* MATLAB, da MathWorks®. Os autores da proposta de aproximação discreta do controlador $PI^\lambda D^\mu$, Merrikh-Bayat *et al.* [14], não as desenvolvem, apresentando apenas a expansão em série de potência mostrada em (19). Tentativas de desenvolvimento destas equações são descritas em Santos *et al.* [24] e El-Shafei *et al.* [25], porém as equações nestas referências apresentam erros.

Assim, a Equação (22) para $M = 5$ pode ser reescrita em equações a diferenças como

$$u[k] = u[k-1] + \mathbf{w}^T \mathbf{E}[k], \tag{29}$$

em que $\mathbf{w} = [w_0 \dots w_6]^T$ e $\mathbf{E}[k] = [E[k] \dots E[k-6]]^T$.

O sistema de controle completo utilizado por cada partícula é ilustrado na Fig. 3, em que o valor de referência utilizado é de $36,1^\circ\text{C}$. Este sistema de controle leva em consideração as limitações existentes em um microcontrolador e em um sistema de controle real. Entre essas limitações, pode-se citar a necessidade de um saturador na saída de controle para garantir que o sinal esteja dentro dos limites permitidos de

operação. Além disso, há a presença de um quantizador, para reproduzir os resultados da conversão analógico-digital (A/D). O microcontrolador atuará em tempo discreto, e neste trabalho seu tempo de atuação é definido como $T_c = 100\text{s}$. Por fim, as variáveis são trabalhadas com seis casas decimais, pois o controlador $PI^\lambda D^\mu$ será embarcado em um microcontrolador de 8 *bits* em variáveis do tipo *float*. Estas inclusões de limitações visam tornar o sistema de controle utilizado na otimização do PSO o mais próximo possível da realidade.

O funcionamento do algoritmo PSO para o problema apresentado nesse trabalho está ilustrado na Fig. 4, em que o critério de parada é o fim das 20 iterações. Neste trabalho, o vetor de posições da j -ésima partícula é penta-dimensional e é definido como $\mathbf{p}_j = [k_{p_j} \ k_{i_j} \ \lambda_j \ k_{d_j} \ \mu_j]$, em que foram definidos limites para os ganhos, como $k_{p_j}, k_{i_j}, k_{d_j} \in [0, 0,1]$ e $\lambda_j, \mu_j \in [0, 1]$. São utilizadas 10 partículas em 20 iterações do algoritmo, com $w_{inicial} = 0,3$, $w_{final} = 0,01$, $c_1 = 1,2$ e $c_2 = 1$. São executadas um total de 20 realizações do algoritmo. Em cada realização, as posições das partículas são iniciadas de forma aleatória.

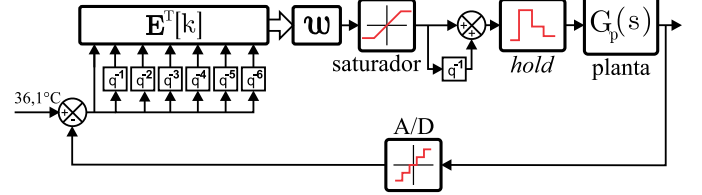


Fig. 3: Sistema de controle em que os parâmetros estimados por cada partícula serão testados.

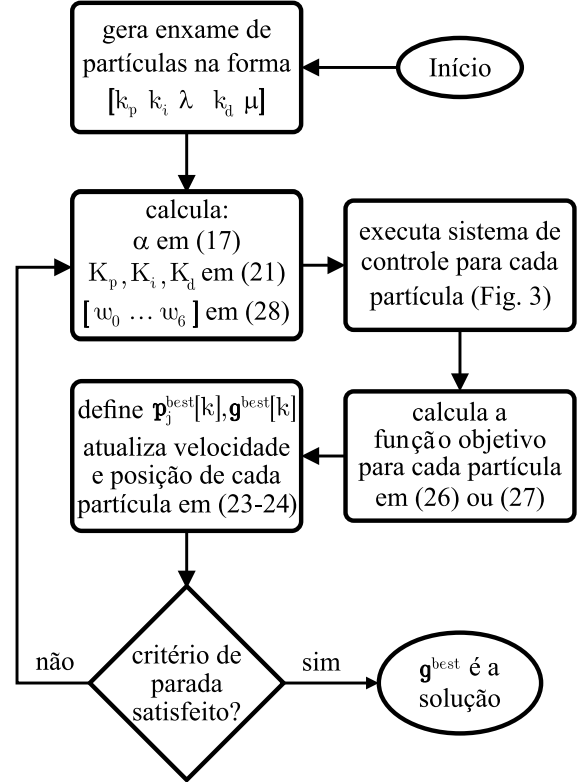


Fig. 4: Fluxograma do algoritmo PSO utilizado neste trabalho.

Comentário 3. Os ganhos K_p , K_i e K_d presentes no fluxo-grama são utilizados apenas para o cálculo de $w_0 \dots w_6$ e são calculados a partir de k_p , k_i e k_d em (21).

Comentário 4. O enxame utilizado na otimização é composto de partículas representando os ganhos k_p , k_i e k_d , além de λ e μ , e não devem ser confundidos com K_p , K_i e K_d .

O controlador $PI^\lambda D^\mu$ obtido a partir da melhor partícula \mathbf{g}^{best} entre todas as realizações é escolhido para ser embarcado no microcontrolador instalado na incubadora neonatal.

VII. RESULTADOS E DISCUSSÕES

O modelo em tempo discreto de segunda ordem resultante da aplicação de estimação RLS apresentado na Seção II, ficou definido como

$$G_p(z) = \frac{Y(z)}{U(z)} = \frac{0,0044 + 0,0455z^{-1} - 0,0474z^{-2}}{1 - 1,5483z^{-1} + 0,5580z^{-2}}, \quad (30)$$

enquanto que o modelo contínuo, obtido a partir de (30) pela aplicação do método *zero-order hold*, é dado por

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{0,0044s^2 + 6,99 \cdot 10^{-5}s + 3,30 \cdot 10^{-9}}{s^2 + 0,0005834 + 1,282 \cdot 10^{-8}}. \quad (31)$$

Após teste preliminares, decidiu-se utilizar o fator de esquecimento do algoritmo RLS como $\phi = 0,96$. A comparação entre os valores reais de aquisição e os estimados pelo modelo são ilustrados na Fig. 5. Observa-se que o modelo possui dinâmica similar ao da incubadora. A raiz do erro quadrático médio (RMSE, na sigla em inglês) entre os dados de aquisição e os estimados pelo modelo é 0,0675.

O resultado da execução do algoritmo PSO presente na Fig. 4 para 20 realizações é ilustrado nas Figs. 6 e 7. As respostas temporais do sistema de controle da Fig. 3 para as 20 partículas \mathbf{g}^{best} das 20 realizações para as funções objetivo J_{ITSE} e J_{RT} são ilustradas nas Figs. 6a e 7a, respectivamente. Já as respectivas curvas de convergência das funções objetivos estão ilustradas nas Figs. 6b e 7b. As curvas que obtiveram menor função objetivo estão ilustradas em destaque. É utilizado $r = -0,2$ em J_{RT} após uma análise prévia.

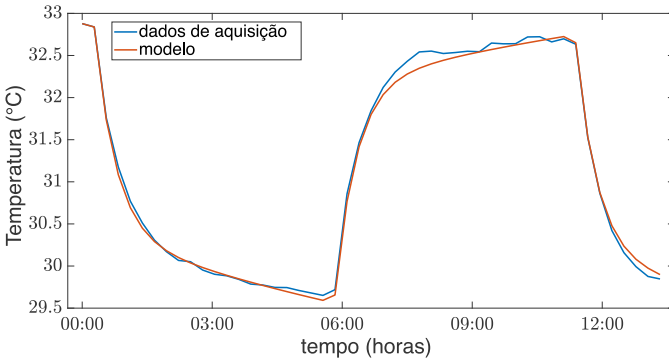
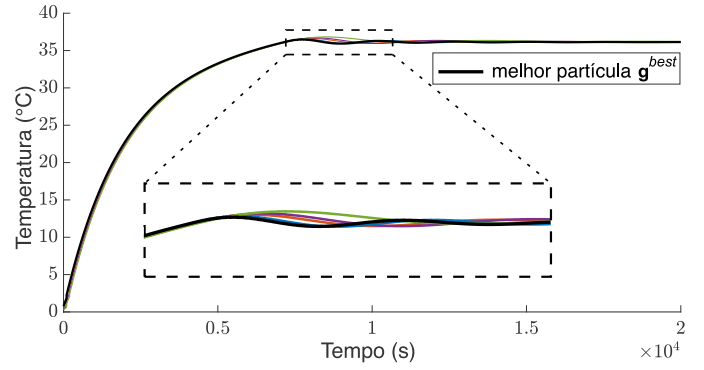
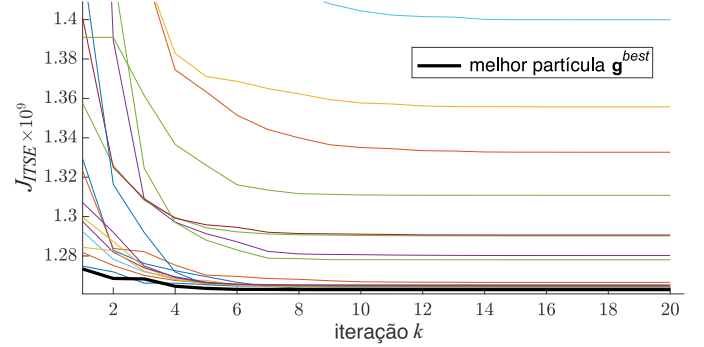


Fig. 5: Comparação entre dados de aquisição e modelo ARX.

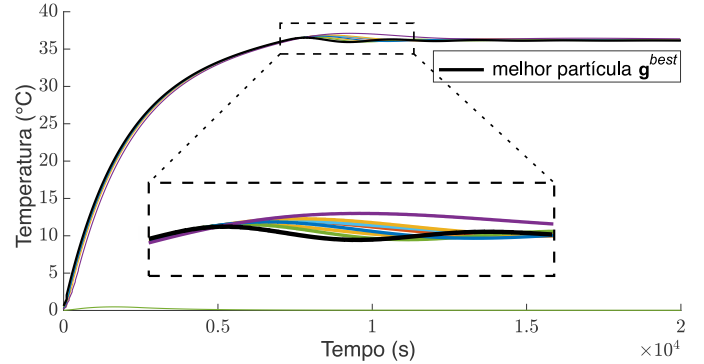


(a) Respostas temporais do sistema de controle.

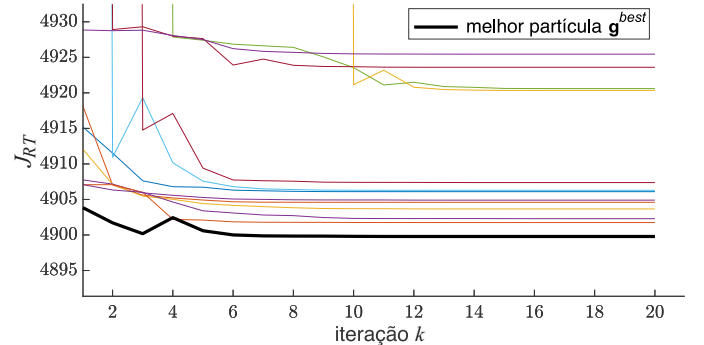


(b) Convergência da função objetivo.

Fig. 6: Desempenho das 20 partículas \mathbf{g}^{best} com J_{ITSE} .



(a) Respostas temporais do sistema de controle.



(b) Convergência da função objetivo.

Fig. 7: Desempenho das 20 partículas \mathbf{g}^{best} com função J_{RT} .

A comparação da resposta temporal entre o melhor g^{best} alcançado pela J_{ITSE} e o melhor g^{best} alcançado pela J_{RT} está ilustrada na Fig. 8 e na Tabela I. Observa-se que as duas funções objetivos alcançaram resultados convergentes, com diferenças mínimas de desempenho. Por possuir resultados ligeiramente melhores, o controlador $PI^\lambda D^\mu$ obtido com a função objetivo J_{RT} é escolhido como o controlador otimizado deste trabalho. Os parâmetros do controlador $PI^\lambda D^\mu$ resultante da otimização são apresentados na Tabela II.

A. Análise de estabilidade

Embora não seja comum a análise de estabilidade em artigos com otimização de controladores $PI^\lambda D^\mu$, essa é uma etapa importantíssima de projeto de sistemas de controle, sobretudo quando se deseja embarcar o controlador.

O mapa de polos do sistema de controle de ordem fracionária é ilustrado na Fig. 9. O plano $\sigma = s^q$ possui $q = 0,01$. O teorema da estabilidade de Matignon [18] aplicado à $Q(\sigma)$ da função de transferência de malha fechada resulta que todas as raízes satisfazem (15), o que significa que o sistema é estável. De fato, o menor valor de $|arg(\sigma_i)|$ é 0,017794, o que é maior que $q\frac{\pi}{2} = 0,015708$. Pode-se observar na Fig. 9b que nenhuma raiz de $Q(\sigma)$ está na região de instabilidade.

O ganho de frequência cruzada para estes ganhos é de $w_c = 0,0022266$. Aplicando os valores mostrados na Tabela II em (21) e (28), tem-se

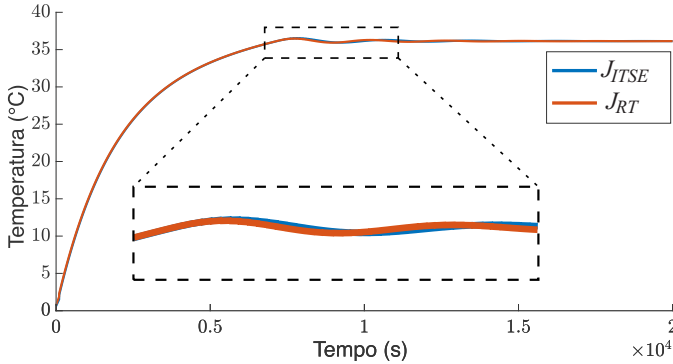


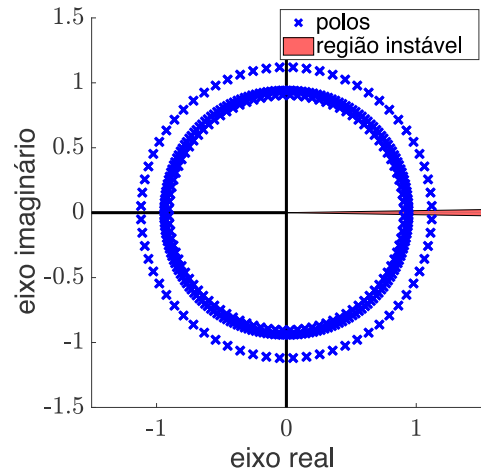
Fig. 8: Comparação entre melhor g^{best} dentre aqueles obtidos com o uso das funções objetivos J_{ITSE} e J_{RT} .

TABELA I: Desempenho das diferentes funções objetivo.

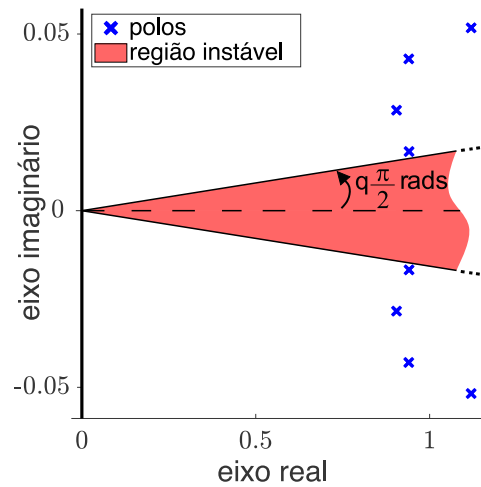
Função objetivo	tempo de subida	tempo de acomodação	sobressinal
J_{ITSE}	4.412,2004s	6.470,79 s	1,198 %
J_{RT}	4.412,1988s	6.468,20 s	1,181 %

TABELA II: Parâmetros resultantes da otimização.

k_p	k_i	k_d	λ	μ
0,0063584	0,084473	0,098577	0,97507	0,6848



(a) Plano s^q , com $q = 0,01$.



(b) Detalhes do plano s^q .

Fig. 9: Mapa de polos do sistema com controlador $PI^\lambda D^\mu$.

$$\begin{aligned}
 w_0 &= 3,859795 & w_1 &= 3,632549 & w_2 &= -0,171446 \\
 w_3 &= -0,071526 & w_4 &= -0,049646 & w_5 &= -0,045465 \\
 w_6 &= -0,033431.
 \end{aligned}$$

A lei de controle aplicada no controlador embarcado é obtida aplicando estes ganhos $w_0 \dots w_6$ em (29). O resultado do controlador $PI^\lambda D^\mu$ embarcado na incubadora neonatal é ilustrado na Fig. (10). Observa-se que na prática, ocorrem oscilações indesejadas na resposta do sistema. É importante ressaltar que essas oscilações existem na resposta simulada obtida com o algoritmo PSO (vide Fig. 8), porém, com menor amplitude e duração. A resposta do sistema de controle obteve sobressinal de 3,32% e tempo de acomodação de 13.668 s.

Embora o modelo ARX obtido tenha RMSE de 0,0675 (vide Fig. 5), uma hipótese que pode ser levantada para a diferença entre os níveis de oscilações é a de que o modelo da planta não é representativo, apresentando um possível problema relacionado ao tempo de discretização e à desconsideração do tempo morto (*dead time*) na dinâmica da planta [26].

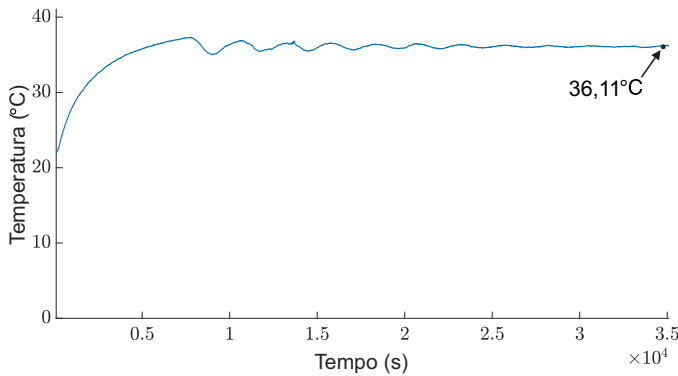


Fig. 10: Resultado real do controlador $PI^{\lambda}D^{\mu}$ embarcado.

VIII. CONCLUSÕES

Neste trabalho, foi realizado o projeto e implementação de um controlador $PI^{\lambda}D^{\mu}$ em um sistema embarcado com microcontrolador de 8-bits para controle de temperatura de uma incubadora neonatal. A identificação da planta por um modelo ARX de segunda ordem pelo algoritmo RLS possui RMSE de 0,0675, embora exista a possibilidade deste modelo não ser representativo o suficiente.

A otimização dos parâmetros do controlador de ordem fracionária pelo algoritmo de enxame de partículas possui limitações que tornam o sistema de controle simulado mais próximo do real. Além disso, é utilizada a aproximação do controlador de ordem fracionária nas simulações de controle durante a otimização, em vez do controlador fracionário de ordem contínua. Este importante passo visa evitar a perda da otimalidade entre os resultados alcançados no tempo contínuo e sua versão aproximada utilizada no tempo discreto.

O otimizador obteve resultados convergentes para diferentes funções objetivos, o que demonstra que o resultado da otimização seja um bom candidato a valor ótimo do problema. Mostrou-se que o sistema de controle com esse controlador é estável, informação essencial para projetos em que se deseja implementar em um sistema embarcado. O tempo de acomodação do sistema de controle em malha fechada é de 6.468 s e com sobressinal de 1,18%, enquanto que no sistema embarcado é de 13.668 s e 3,32%, respectivamente.

REFERÊNCIAS

- [1] P. Shah and S. Agashe, "Review of fractional PID controller," *Mechatronics*, vol. 38, pp. 29–41, 2016.
- [2] A. A. Jamil, W. F. Tu, S. W. Ali, Y. Terriche, and J. M. Guerrero, "Fractional-order PID controllers for temperature control: A review," *Energies*, vol. 15, no. 10, p. 3800, 2022.
- [3] V. Feliu-Battle, R. Rivas-Perez, and F. J. Castillo-Garcia, "Robust fractional-order temperature control of a steel slab reheating furnace with large time delay uncertainty," in *ICFDA'14 International Conference on Fractional Differentiation and Its Applications 2014*. IEEE, 2014, pp. 1–6.
- [4] I. Petráš and B. Vinagre, "Practical application of digital fractional-order controller to temperature control," *Acta Montanistica Slovaca*, vol. 7, no. 2, pp. 131–137, 2002.
- [5] R. S. Barbosa, J. T. Machado, and I. S. Jesus, "Effect of fractional orders in the velocity control of a servo system," *Computers & Mathematics with Applications*, vol. 59, no. 5, pp. 1679–1686, 2010.
- [6] M. K. Bouafoura and N. B. Braiek, " $PI^{\lambda}D^{\mu}$ controller design for integer and fractional plants using piecewise orthogonal functions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 5, pp. 1267–1278, 2010.
- [7] J.-Y. Cao, J. Liang, and B.-G. Cao, "Optimization of fractional order PID controllers based on genetic algorithms," in *2005 international conference on machine learning and cybernetics*, vol. 9. IEEE, 2005, pp. 5686–5689.
- [8] A. Abraham, A. Biswas, S. Das, and S. Dasgupta, "Design of fractional order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 1445–1452.
- [9] A. Rajasekhar, V. Chaitanya, and S. Das, "Fractional-order $PI^{\lambda}D^{\mu}$ controller design using a modified artificial bee colony algorithm," in *Swarm, Evolutionary, and Memetic Computing: Second International Conference, SEMCCO 2011, Visakhapatnam, Andhra Pradesh, India, December 19-21, 2011, Proceedings, Part I 2*. Springer, 2011, pp. 670–678.
- [10] M. Karimi-Ghartemani, M. Zamani, N. Sadati, and M. Parniani, "An optimal fractional order controller for an AVR system using particle swarm optimization algorithm," in *2007 Large Engineering Systems Conference on Power Engineering*. IEEE, 2007, pp. 244–249.
- [11] M. Zamani, M. Karimi-Ghartemani, N. Sadati, and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Engineering Practice*, vol. 17, no. 12, pp. 1380–1387, 2009.
- [12] I. Podlubny, "Geometric and physical interpretation of fractional integration and fractional differentiation," *arXiv preprint math/0110241*, 2001.
- [13] A. J. Calderón, B. M. Vinagre, and V. Feliu, "Fractional order control strategies for power electronic buck converters," *Signal Processing*, vol. 86, no. 10, pp. 2803–2819, 2006.
- [14] F. Merrikh-Bayat, N. Mirebrahimi, and M. R. Khalili, "Discrete-time fractional-order PID controller: Definition, tuning, digital realization and some applications," *International journal of control, automation and systems*, vol. 13, pp. 81–90, 2015.
- [15] I. R. Sousa, F. J. Vasconcelos, F. G. Nogueira, and G. A. Barreto, "Identificação e controle LPV de uma incubadora neonatal," in *XXIV Congresso Brasileiro de Automática - CBA*. SBA, 2022.
- [16] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2002.
- [17] D. Xue, *Fractional-order control systems*. de Gruyter, 2017.
- [18] D. Matignon, "Stability results for fractional differential equations with applications to control processing," in *Computational engineering in systems applications*, vol. 2, no. 1. Lille, France, 1996, pp. 963–968.
- [19] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 3. IEEE, 1999, pp. 1945–1950.
- [20] H.-L. Hung, Y.-F. Huang, C.-M. Yeh, and T.-H. Tan, "Performance of particle swarm optimization techniques on PAPR reduction for OFDM systems," in *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2008, pp. 2390–2395.
- [21] R. C. Bishop and R. H. Dorf, *Modern control systems*. Prentice Hall, 2011, vol. 12.
- [22] M. P. Aghababa, "Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm," *Soft Computing*, vol. 20, no. 10, pp. 4055–4067, 2016.
- [23] L. A. Aguirre, *Introdução à Identificação de Sistemas—Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 2015.
- [24] N. M. Santos, P. R. Martins, G. C. Silva Neto, and F. A. Ayres Junior, "Projeto e implementação de um controlador FOPI em uma bancada de controle de nível," in *XXIV Congresso Brasileiro de Automática - CBA*. SBA, 2022.
- [25] M. A. El-Shafei, M. I. El-Hawary, and H. M. Emara, "Implementation of fractional-order PID controller in an industrial distributed control system," in *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2017, pp. 713–718.
- [26] L. R. da Silva, R. C. C. Flesch, and J. E. Normey-Rico, "Controlling industrial dead-time systems: When to use a PID or an advanced controller," *ISA transactions*, vol. 99, pp. 339–350, 2020.