

# PID Controller Tuning with Two Degrees of Freedom by Particle Swarm Optimization

## Sintonia de controlador PID com Dois Graus de Liberdade por Otimização de Enxame de Partículas

Vitor C. Chagas<sup>1</sup>, João Pedro M. Araujo<sup>1</sup>, Daniel M. Cruz<sup>1</sup>, João Gustavo C. Pena<sup>1</sup>

<sup>1</sup> Faculdade de Engenharia - FAENG  
Universidade Federal de Mato Grosso – Várzea Grande, MT – Brazil

vitor.chagas@sou.ufmt.br, joao.araujo@sou.ufmt.br,

daniel.cruz@ufmt.br, joao.pena@ufmt.br

**Abstract.** While PID controllers are the most widely used in the industry and have a vast array of analytical and empirical tuning methods, the same cannot be said for two-degree-of-freedom PID controllers. Therefore, this article aims to assess the use of the PSO in tuning the parameters of this type of controller. In this study, the method employed seeks to find the optimal controller gain values applied to a set of first-order systems under the same simulation operating conditions, with the goal of improving performance criteria such as robustness and control loop stability, in other words, minimizing the cost function. The results obtained by the algorithm are compared with those obtained using classical empirical methods, specifically the CHR, IAE, and ITAE methods.

**Keywords:** 2DOF Controller; PID Tuning; PSO Algorithm; IAE; ITAE.

**Resumo.** Embora os controladores PID sejam os mais usados na indústria e para os quais existe um grande número de métodos analíticos e empíricos de sintonia, o mesmo não pode ser dito acerca dos controladores PID com dois graus de liberdade. Por isso, este artigo tem como objetivo avaliar o uso do PSO no ajuste dos parâmetros desse tipo de controlador. Neste estudo, o método utilizado busca encontrar os valores ótimos dos ganhos do controlador aplicado a um conjunto de sistemas de primeira ordem nas mesmas condições operacionais de simulação, a fim de melhorar os critérios de desempenho como robustez e estabilidade da malha de controle, isto é, minimizar a função custo. Os resultados obtidos pelo algoritmo são comparados com os resultados obtidos com o uso dos métodos empíricos clássicos, no caso, os métodos CHR, IAE e ITAE.

**Palavras-chave:** Controlador 2DOF; Sintonia PID; Algoritmo PSO; IAE; ITAE.

### 1. Introdução

A técnica de Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO), desenvolvido por [Kennedy and Eberhart 1995], é uma técnica de otimização inspirada no comportamento de enxames na natureza. Essa abordagem tem a capacidade de explorar eficientemente espaços de solução em busca de convergência para soluções ótimas

ou quase ótimas, como ilustrado em [Xu 2022], que a utiliza para ajustar controladores Proporcional-Integral-Derivativo (PID) em sistemas de controle de veículo elétrico DC. Estudos conduzidos por [Kennedy and Mendes 2002] e [Mirjalili et al. 2012] demonstraram que métodos baseados em PSO superam técnicas tradicionais de ajuste em termos de velocidade de convergência e desempenho. Por essa razão, o PSO tem sido empregado para ajustar parâmetros de controladores PID industriais, incluindo a versão mais complexa do PID com 2 Graus de Liberdade (2DOF PID), que permite uma solução entre a trajetória de referência (*setpoint*) e a trajetória de perturbação, tornando possível controlar sistemas onde as perturbações têm um impacto significativo [Rovira et al. 1969]. Nesse contexto, métodos de ajuste de parâmetros de controladores PID são cruciais para garantir o desempenho ideal do sistema de controle, tanto na configuração simples quanto na 2DOF PID. No entanto, o ajuste manual de parâmetros pode ser um processo complexo, e métodos clássicos como Ziegler-Nichols e Cohen-Coon frequentemente requerem conhecimento especializado e experimentação repetitiva para alcançar resultados que satisfaçam as especificações do processo.

Portanto, este artigo avalia o uso do método PSO para ajustar os parâmetros dos controladores PID e 2DOF PID, e faz uma comparação do desempenho do sistema de controle proposto em relação à seu desempenho quando sintonizado com métodos clássicos, como Chien, Hrones e Reswick (CHR), Integral do Erro Absoluto (IAE) e Integral Ponderada no Tempo do Valor Absoluto do Erro (ITAE). As simulações são realizadas considerando funções de transferência de primeira ordem com diferentes valores de ganhos e constantes de tempo, mantendo um cenário de simulação fixo, com variações de *setpoint* e perturbação. Esse artigo é assim organizado: na Seção 1 é feita a introdução do tema de estudo; na Seção 2 é feita a revisão bibliográfica dos principais temas e métodos abordados no trabalho; na Seção 3 é apresentada a Metodologia aplicada para a construção da solução proposta, para medição e comparação entre os diferentes métodos; depois, na Seção 4 são apresentados e discutidos os principais resultados do estudo e, finalmente, na Seção 5 é feita a conclusão do trabalho, sendo apresentadas as considerações finais.

## 2. Revisão Bibliográfica

Nesta seção é feita a revisão bibliográfica da técnica de otimização utilizada para escolha dos parâmetros do controlador 2 DOF PID, também é feita a revisão de literatura dos controladores PID simples e com dois graus de liberdade, além de alguns dos métodos de sintonia de controladores, como CHR e métodos da integral do erro.

### 2.1. Otimização por Enxame de Partículas - PSO

O PSO é uma técnica de otimização inspirada no comportamento de bandos de pássaros e cardumes de peixes. Seu objetivo é minimizar ou maximizar uma função objetivo definida pelo usuário, que pode ser complexa ou não linear. Cada partícula representa uma possível solução e é influenciada tanto pela sua própria experiência quanto pela experiência do enxame. As equações fundamentais do PSO definidas por [Jain et al. 2022] são:

$$V(i, j) = W \cdot V(i, j) + C_1 \cdot rand() \cdot (P_{best}(i, j) - X(i, j)) + C_2 \cdot rand() \cdot (G_{best}(j) - X(i, j)) \quad (1)$$

$$X(i, j) = X(i, j) + V(i, j) \quad (2)$$

Nessas Equações 1 e 2,  $V(i,j)$  representa a velocidade da partícula  $i$  na dimensão  $j$ ,  $W$  é o peso inercial,  $C_1$  e  $C_2$  são constantes de aprendizado,  $\text{rand}()$  é uma função de números aleatórios uniformemente distribuídos,  $P_{best}(i,j)$  é a melhor posição já visitada pela partícula  $i$  na dimensão  $j$ , e  $G_{best}(j)$  é a melhor posição já visitada pelo enxame na dimensão  $j$ . O coeficiente de aprendizado cognitivo,  $C1$ , influencia a partícula a retornar à melhor solução que ela explorou, sendo que sua intensidade é diretamente proporcional ao valor de  $C1$ . Quanto maior o valor de  $C1$ , mais importante é a experiência individual da partícula na busca por soluções ótimas.  $C2$ , por sua vez, representa o coeficiente de aprendizado social e desempenha um papel semelhante, mas influencia todas as partículas. A intensidade dessa influência também é proporcional a  $C2$ , onde valores maiores priorizam soluções ótimas.

O PSO tem sido bem-sucedido na otimização dos ganhos de controladores PID em diversos estudos. Por exemplo, em [Gao and Yang 2013], o PSO ajustou os ganhos de um controlador PID em um sistema de controle de nível em um tanque, resultando em melhorias significativas no desempenho com base em critérios da função objetivo. A função objetivo pode variar, incluindo a minimização do erro médio, otimização do tempo de resposta ou redução do erro quadrático integral. Neste artigo, uma técnica de PSO é aplicada para minimizar a função objetivo formulada de modo a levar em conta os seguintes custos: sobressinal (*overshoot*), tempo de estabilização, erro quadrático integral (ISE), erro quadrático integral ponderado pelo tempo (ITSE) e erro em estado estacionário. O PSO foi ajustado empiricamente de modo a reduzir o tempo necessário para encontrar uma boa solução para os parâmetros do controlador 2DOF PID aplicado à algumas malhas de controle de primeira ordem.

## 2.2. Controlador PID

Para [Åström and Murray 2008], por várias razões, os controladores PID são os tipos de controladores mais usados na indústria. Em primeiro lugar, o controlador PID é fácil de implementar e seus parâmetros podem ser ajustados rapidamente para diferentes situações. Em segundo lugar, eles possuem uma boa combinação de estabilidade, precisão e resposta, o que os tornam adequados para uma ampla variedade de aplicações. Em terceiro, em situações em que os modelos matemáticos não são conhecidos, ou outros modelos analíticos não podem ser desenvolvidos, os controladores PID ainda podem ser aplicados com sucesso. Definida em [Ogata 2010], a Equação 3 descreve a configuração básica do controlador PID, onde  $K$  é o ganho em malha aberta,  $s$  é o operador de Laplace, sendo o zero da função de transferência a solução do polinômio  $s + a$ .

$$G_c(s) = K \frac{(s + a)^2}{s} \quad (3)$$

Embora o controlador forneça bons resultados, existem algumas questões que podem degradar o desempenho do sistema. Um exemplo disso é que, se a entrada de referência for uma função degrau, devido ao termo derivativo na ação de controle, a variável manipulada resultará em uma função impulso, o que causa um sobressinal na variável controlada. A ação derivativa foi implantada segundo o descrito por [Ogata 2010], onde em vez do termo derivativo puro  $T_d$ , foi utilizado  $T_d s / (\gamma T_d s + 1)$ , com  $\gamma = 0.125$ . Dessa forma, quando a entrada de referência é uma função degrau, a variável manipulada não produzirá uma função impulso, mas sim um salto no valor de referência.

### 2.3. Controlador 2DOF PID

De acordo com [Ogata 2010], os graus de liberdade em um sistema de controle se referem a quantas dessas funções de transferência em malha fechada são independentes. [Alfaro and Vilanova 2013] abordam em seu trabalho a Equação 4 que descreve o 2DOF PID usado neste artigo, onde  $r(s)$  é o ponto de ajuste,  $y(s)$  é a variável controlada,  $d(s)$  é uma perturbação, e  $\alpha$  e  $\beta$  são parâmetros de ponderação do controlador, começando com valores aleatórios no intervalo de 0 a 1, e durante as iterações procuram os melhores valores livremente. Veja sua descrição na Figura 1 [Indhuja and Kamaraj 2016].

$$u(s) = K_p \left\{ e_p(s) + \frac{1}{T_i s} e_i(s) + \frac{T_d s}{\gamma T_d s + 1} e_d(s) \right\} \quad (4)$$

com

$$e_p(s) = \alpha r(s) - y(s) \quad e_i(s) = r(s) - y(s) \quad e_d(s) = \beta r(s) - y(s) \quad (5)$$

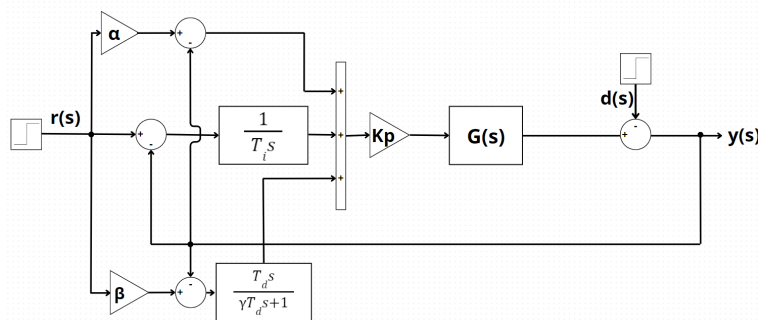


Figura 1. Configuração do controlador PID com 2 Graus de Liberdade (2DoF).

A função de transferência do processo  $G(s)$  é representada por um modelo de primeira ordem cujos parâmetros são o ganho ( $K$ ), a constante de tempo ( $\tau$ ) e tempo morto ( $\theta$ ), que são usados pelos métodos de sintonia para ajuste dos parâmetros do controlador. Os valores de  $G(s)$  são obtidos por meio da curva de resposta ao degrau em malha aberta.

### 2.4. Método CHR

O método CHR foi desenvolvido por CHIEN, HRONES e RESWICK [Chien et al. 1952]. Este método se baseia na análise da resposta mais rápida sem sobressinal (controle servo) e na análise da resposta mais rápida com 20% de sobressinal (controle regulatório). A Tabela 1 apresenta os respectivos valores de  $K_p$ ,  $T_i$  e  $T_d$  de acordo com os critérios [de Campos and Teixeira 2010].

Tabela 1. Valores das Constantes PID.

Controlador	Critério	$K_p$	$T_i$	$T_d$
PID	Servo	$\frac{0.6\tau}{K}$	$\tau$	$\frac{\theta}{2}$
PID	Regulatório	$\frac{0.95\tau}{K\theta}$	$2.375\tau$	$0.421\theta$

## 2.5. Método da Integral do Erro - IAE e ITAE

[Lopez et al. 1967] propõem um método para minimizar os índices (IAE ou ITAE) em problemas de controle dos tipos regulatório e servo. No caso, o método IAE representa a integral do erro absoluto e consiste no ajuste dos parâmetros PID minimizando o erro absoluto do sinal em relação à referência. Já o método ITAE é uma integral ponderada no tempo do valor absoluto do erro (ITAE), que penaliza erros mais pronunciados que persistem por um longo período de tempo, após a aplicação de uma perturbação ou de uma mudança no ponto de ajuste.

$$IAE = \int_0^{\infty} |e(t)| dt \quad ITAE = \int_0^{\infty} t|e(t)| dt \quad (6)$$

Para o sistema regulatório, são utilizadas as equações apresentadas na Equação 7, e as constantes A, B, C, D, E e F estão na Tabela 2, segundo [de Campos and Teixeira 2010], para o controlador e critérios analisados.

$$K_p = \frac{1}{K} \cdot (A \cdot (\frac{\theta}{\tau})^B) \quad T_i = \frac{\tau}{(C \cdot (\frac{\theta}{\tau})^D)} \quad T_d = \tau \cdot (E \cdot (\frac{\theta}{\tau})^F) \quad (7)$$

**Tabela 2. Valores das Constantes PID (regulatório).**

Critério	A	B	C	D	E	F
IAE	1.435	-0.921	0.878	-0.749	0.482	1.137
ITAE	1.357	-0.947	0.842	-0.738	0.381	0.995

[Rovira et al. 1969] descreve um método que minimiza o IAE e o ITAE para um problema de servo (perturbação no ponto de ajuste - mudança na referência do sistema). As equações apresentadas na Equação 8 são usadas para os parâmetros do PID no caso servo e as constantes das fórmulas estão na Tabela 3 para os critérios analisados conforme [de Campos and Teixeira 2010] indica em seu trabalho.

$$K_p = \frac{1}{K} \cdot (A \cdot (\frac{\theta}{\tau})^B) \quad T_i = \frac{\tau}{(C + D \cdot (\frac{\theta}{\tau}))} \quad T_d = \tau \cdot (E \cdot (\frac{\theta}{\tau})^F) \quad (8)$$

**Tabela 3. Valores das Constantes PID (servo).**

Critério	A	B	C	D	E	F
IAE	1.086	-0.869	0.740	-0.130	0.348	0.914
ITAE	0.965	-0.850	0.796	-0.147	0.308	0.929

## 3. Metodologia

Neste trabalho, a combinação entre a linguagem de programação Python e o *software* Matlab é usada para identificar e aplicar sucessivamente o sistema de controle. Para os métodos aplicados, PSO, CHR, IAE e ITAE, o formato da função de transferência (ver

Equação 9) é utilizado com os valores de  $K$  e  $\tau$  variando pelos valores do vetor [0.1, 0.5, 1.0, 2.0, 10.0] e o valor de  $\theta$  é fixado em 0.001.

$$G(s) = \frac{K \cdot e^{-\theta s}}{\tau + 1} \quad (9)$$

A Figura 2 mostra o diagrama de fluxo do processo deste estudo, destacando as etapas em que o sistema de controle foi simulado enquanto o otimizador ajustava e avaliava a resposta com base nos critérios de desempenho. A inicialização do otimizador ocorre em Python, onde os dados de entrada do sistema são definidos, incluindo função de transferência, ponto de ajuste, fatores de inércia, cognitivos e sociais, critério de parada e número de partículas. O PSO começa com uma distribuição aleatória das partículas e velocidades no sistema, e os erros e o cálculo da função de custo (FO) são determinados na primeira iteração. Após a definição dos dados de entrada, usando a biblioteca *matlab.engine*, a configuração do 2DOF PID (Figura 1) é inicializada no *software* Matlab/Simulink, e os valores dos ganhos a serem explorados pelo PSO são definidos. Além dos ganhos, vetores de saída e erros ISE, ITAE e ITSE são gerados para utilização durante as iterações até que o critério de parada seja alcançado.

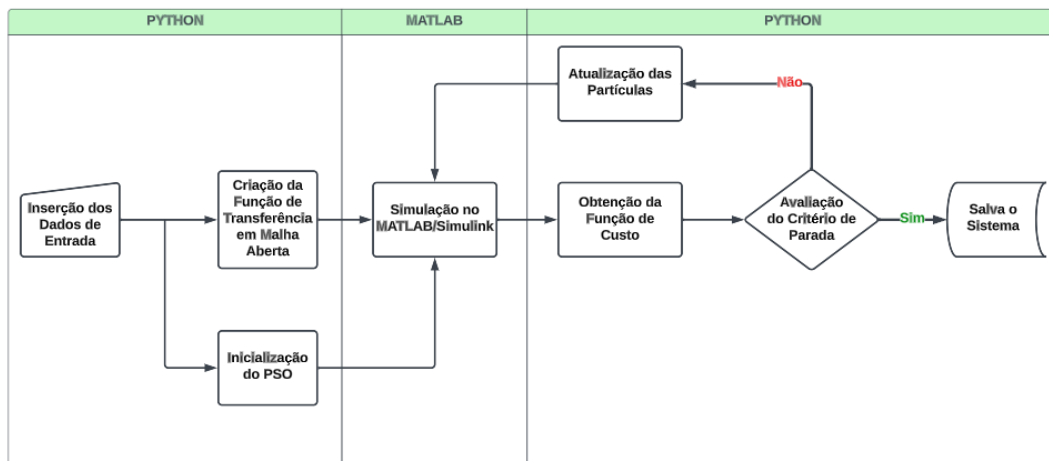


Figura 2. Diagrama de Fluxo do PSO.

Após obter as variáveis e vetores, o código retorna ao Python para a otimização. A escolha dos valores do PSO se dá através de tentativa-e-erro [Poli et al. 2007] através de um certo número de experimentos, pois depende da natureza do problema. O número de partículas usado neste trabalho é 10,  $C_1$  é definido como 1,  $C_2$  como 0.6 e  $W$  como 0.75, escolhidos empiricamente. Segundo [Poli et al. 2007], esses valores podem variar no intervalo [0,4]. Os critérios para criar e explorar as partículas seguem os valores indicados na Tabela 4. Durante as iterações, o tempo de inicialização do algoritmo, o número de melhorias contabilizadas e o cálculo dos erros das partículas aleatórias são armazenados. Além desses valores, os valores de  $P_{best}$  e  $G_{best}$  são identificados. Isso é feito para acompanhar os valores dos ganhos do controlador de cada repetição.

Nas iterações subsequentes, uma nova velocidade é determinada para as partículas com base no tempo de estabilização ( $t_s$ ), erro estacionário ( $e_s$ ) e sobressinal (PO). Essa

**Tabela 4. Intervalo das Partículas.**

Partículas	$K_p$	$\tau_i$	$\tau_d$	$\alpha$	$\beta$
0, 1	0 a 1	0 a 1	0 a 1	0 a 1	0 a 1
2, 3, 4	1 a 10	1 a 10	1 a 10	0 a 1	0 a 1
5, 6	10 a 50	10 a 50	10 a 50	0 a 1	0 a 1
7, 8, 9	50 a 100	50 a 100	50 a 100	0 a 1	0 a 1

velocidade influencia o deslocamento que as partículas realizarão nas próximas iterações. A relação entre  $K_p$ ,  $\tau_i$ ,  $\tau_d$ ,  $\alpha$  e  $\beta$  está ligada à função de custo FO.

$$FO = \frac{(PO - 1) + t_s + (ISE \cdot 100) + (e_s \cdot 100) + ITSE}{5} \quad (10)$$

Esses parâmetros foram selecionados para a função de custo devido ao seu impacto direto na resposta do sistema. Por exemplo, o sobressinal em sistemas de controle refere-se à quantidade pela qual a resposta de um sistema excede o valor desejado antes de convergir para a posição desejada [Åström and Murray 2008]. Esse comportamento pode prejudicar a resposta do modelo, causando respostas instáveis, tempo de estabilização prolongado, vibrações, danos mecânicos, inconsistência e imprecisão. O tempo de estabilização é uma medida que descreve quão rapidamente um sistema atinge a estabilidade após uma perturbação. O valor desse intervalo é definido como uma porcentagem do valor em estado estacionário, geralmente 5% ou 2% [Ogata 2010]. O ISE e o ITSE são métricas usadas para avaliar o desempenho de um sistema de controle em relação ao erro ao longo do tempo [Ogata 2010]. O erro em estado estacionário é uma medida de quão próximo o sistema de controle atinge o valor de referência desejado após a estabilização, definido como a diferença entre o valor de referência desejado e o valor real do sistema quando ele atinge o estado estacionário. O critério de parada é alcançado quando ocorrem 5 iterações consecutivas sem que nenhuma partícula tenha uma melhora maior que 0.001.

Nos métodos CHR, IAE e ITAE, os algoritmos são executados em Python, determinando os valores de  $K_p$ ,  $\tau_i$  e  $\tau_d$ , com o valor de  $\theta$  fixado em 0.001. Esses valores são, então, aplicados à configuração do controlador PID no *software* Matlab, que utiliza uma função de transferência  $G(s)$  e o formato simples do controlador PID. O tempo de simulação é definido como quatro vezes o tempo de estabilização em malha aberta, com uma perturbação introduzida na metade desse tempo. Os parâmetros obtidos com base nas melhores respostas dos métodos são registrados para comparações subsequentes nos resultados, incluindo sobressinal,  $t_s$ , ISE, ITSE, erro em estado estacionário e função de custo.

$$ISE = \int_0^t e^2 dt \quad ITSE = \int_0^t te^2 dt \quad (11)$$

#### 4. Resultados

Os resultados são apresentados a seguir. A Figura 3 apresenta a resposta dinâmica do sistema S45 ajustado via PSO. Neste exemplo, o sobressinal foi de aproximadamente

3%, o  $t_s$  foi de 0.33 segundos, o ISE foi de 0.02, o ITSE foi de 0.72, o erro em estado estacionário foi de 0 e a FO foi de 0.62. Note na Figura 4 que o sobressinal nos métodos regulatórios é pior, comparado ao PSO que alcança resultados entre 0% a 17%. Essa variação ocorre devido ao tipo de sistema e compensação feita pelo otimizador, com base nos critérios de parada e na função de custo estabelecida (Equação 10). Além disso, observa-se que os sistemas, cujo controlador foi sintonizado pelo PSO, alcançam tempos de estabilização mais rápidos em quase todos os resultados devido à minimização do erro em estado estacionário, como visto no resultado da função de transferência S15.

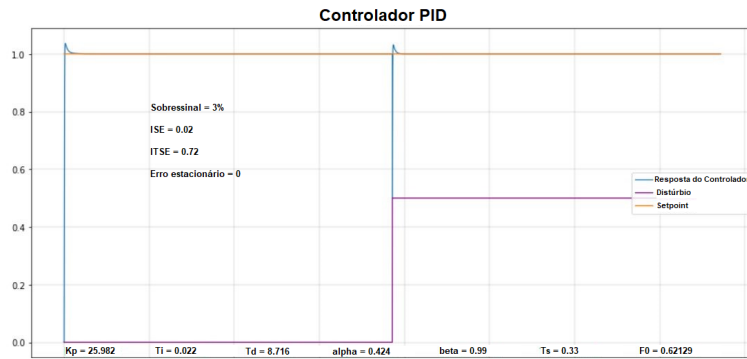


Figura 3. Simulação do sistema S45 ajustado pelo PSO.

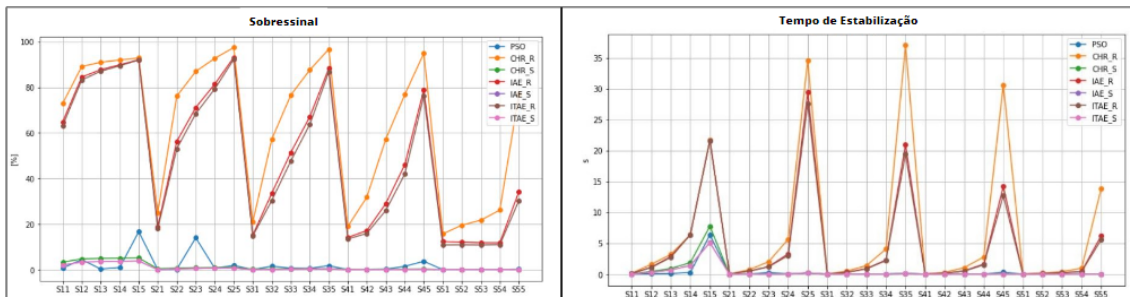


Figura 4. Sobressinal e Tempo de Estabilização.

Na Figura 5 observa-se que a resposta obtida pelo PSO é tão boa quanto a obtida pelo método ITAE (servo) pelo critério do ISE e ligeiramente melhor pelo critério ITSE, sendo esses nesses dois quesitos, muito melhores que os demais métodos. No caso, o PSO produz uma resposta que reduz os erros ISE e ITSE e melhora a estabilidade do sistema.

Os dois gráficos na Figura 6 foram gerados para facilitar a comparação entre os métodos, e revelam que a porcentagem de erro em estado estacionário dos métodos regulatórios clássicos supera significativamente os métodos clássicos de servo e o PSO. A Figura 7 mostra os valores da Função de Custo, que descreve a relação entre as variáveis de controle e os critérios de desempenho do sistema, obtidos por cada método. Ao analisar os resultados, é possível observar que o método proposto fornece respostas melhores em relação aos métodos CHR, IAE e ITAE.

## 5. Conclusão

O PSO obteve, na maioria dos resultados simulados, as melhores respostas, mostrando-se uma técnica eficaz para escolha dos parâmetros do controlador 2DOF PID, em plantas



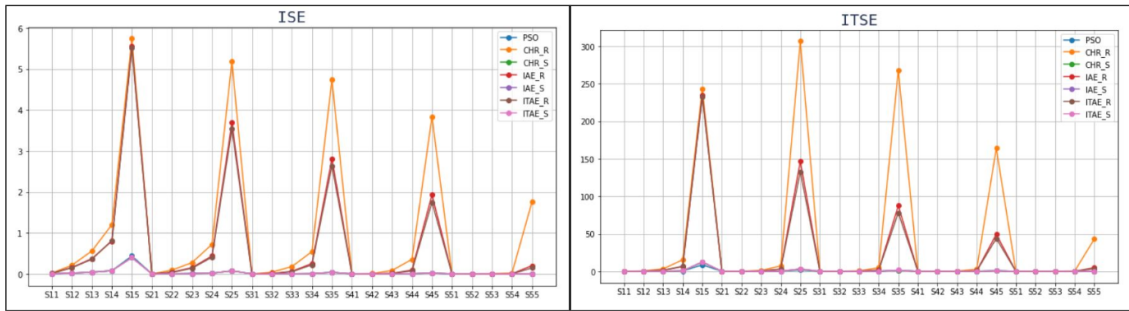


Figura 5. ISE e ITSE.

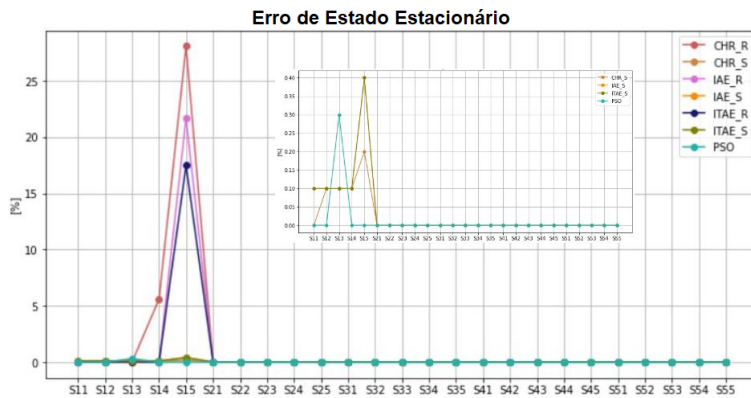


Figura 6. Erro em Estado Estacionário.

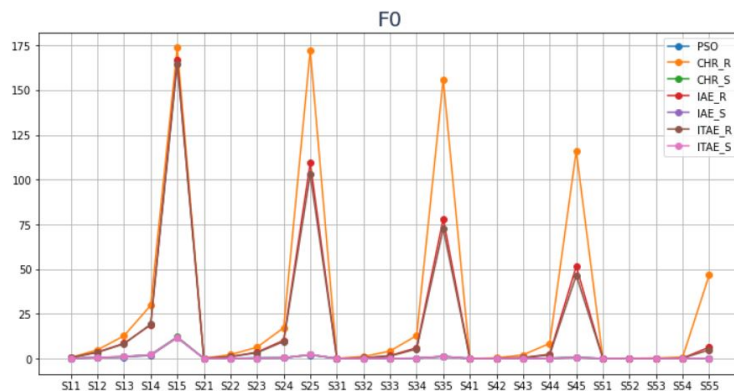


Figura 7. Função de Custo.

cuja função de transferência é de primeira ordem. Esse resultado é interessante, porque é uma técnica conceitualmente simples, de fácil implantação, e aderente a diferentes problemas de otimização. Nesse caso, como não são tão comuns os métodos de sintonia como são no caso dos controladores PID simples, é útil verificar que a otimização foi bem sucedida. Outra vantagem, em relação aos demais métodos de sintonia, é que o PSO não requer informações prévias sobre o sistema, isto é, não é necessário o conhecimento da função de transferência. No entanto, é importante considerar algumas de suas propriedades, como sensibilidade às condições iniciais e a possibilidade de ficar preso em locais excepcionais, que podem afetar a robustez e estabilidade do sistema. Para atenuar esses problemas, foram realizados ensaios para uma pré-sintonia dos parâmetros do PSO.

Trabalhos futuros podem explorar o uso do PSO em sistemas de ordem mais elevada e em aplicações de controle adaptativo em tempo real, incluindo execução em plantas piloto.

## Referências

- Alfaro, V. M. and Vilanova, R. (2013). Model reference based robust tuning of five-parameter 2dof pid controllers for first-order plus dead-time models. In *2013 European Control Conference (ECC)*, pages 3931–3936.
- Chien, I. L., Hrones, J. A., and Reswick, J. B. (1952). On the automatic control of generalized passive systems. *Trans. ASME*, 74:175–185.
- de Campos, M. C. M. M. and Teixeira, H. C. G. (2010). *Controles Típicos de Equipamentos e Processos Industriais*. Blucher, São Paulo, 2nd edition.
- Gao, Z. and Yang, Y. (2013). Tuning of pid controllers based on particle swarm optimization for liquid level control of a spherical tank. *ISA transactions*, 52(1):97–104.
- Indhuja, R. and Kamaraj, N. (2016). Implementation of pso tuned robust 2dof pid controller for avr system. In *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 159–163.
- Jain, M., Saihjal, V., Singh, N., and Singh, S. B. (2022). An overview of variants and advancements of pso algorithm. *Applied Sciences*, 12(17).
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE.
- Kennedy, J. and Mendes, R. (2002). Population structure and particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1671–1676. IEEE.
- Lopez, A. M. et al. (1967). Tuning controllers with error-integral criteria. *Instrumentation Technology*, 14:57–62.
- Mirjalili, S., Gandomi, A. H., and Mirjalili, S. M. (2012). Multi-objective bat algorithm: a novel multi-objective optimization algorithm. *Neural Computing and Applications*, 22(6):1239–1255.
- Ogata, K. (2010). Engenharia de controle moderno. In *Engenharia de controle moderno*, pages 521–544. Pearson Prentice Hall, 5 edition.
- Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimisation: An overview. *Swarm Intelligence Journal*.
- Rovira, A. A., Muril, P. W., and Smith, C. L. (1969). Tuning controllers for set-point changes. *Instruments and Control Systems*.
- Xu, M. (2022). Control of dc adjustable speed electric vehicle based on pso-pid algorithm optimization research. In *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 616–621.
- Åström, K. J. and Murray, R. M. (2008). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.