

## Abordagens meta-heurísticas na sintonia de PID de ordem fracionária

Yan Lieven Souza Lúcio\* Luiza Scapinello Aquino\*  
Viviana Cocco Mariani\*,\*\* Leandro dos Santos Coelho\*,\*\*

\* *Programa de Pós-graduação em Engenharia Elétrica (PPGEE),  
Universidade Federal do Paraná (UFPR), PR, (e-mails:  
yanlieven@ufpr.br, luiza.scapinello@ufpr.br).*

\*\* *Programa de Pós-Graduação em Engenharia Mecânica (PPGEM),  
Pontifícia Universidade Católica do Paraná (PUCPR), PR, (e-mail:  
viviana.mariani@pucpr.br)*

\*\*\* *Programa de Pós-Graduação em Engenharia de Produção e  
Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná  
(PUCPR), PR, (e-mail: leandro.coelho@pucpr.br)*

---

**Abstract:** In most cases, careful tuning for conventional Proportional-Integral-Derivative (PID) controllers is used to get the highest performance for the widest range of applications. However, in more complex cases, such as multiple input and multiple output (MIMO) systems, fine-tuning methods do not extract the best behavior from the system. This performance can be improved using more advanced techniques, such as Fractional Order PID (FOPID), which is based on fractional calculus. Metaheuristic optimization techniques can be used to solve the problem of optimizing a PID controller. The goal of this paper is to assess the effectiveness of FOPID when it is tweaked using metaheuristics, including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Harris Hawks Optimization (HHO), Marine Predators Algorithm (MPA), and Manta Ray Foraging Optimization (MRO). The ball mill pulverizing system was utilized as a test technique to evaluate tuning performance. Over 50 different runs and statistical metrics of the fitness function were applied to assess the metaheuristics' performance. At the end of this process, it became evident that the techniques are efficient in general, and the MPA was the metaheuristic with the smallest objective function values for structured tuning in the ball mill pulverizing system, making it the most efficient technique, while both classical techniques GA and PSO were the least efficient ones.

**Resumo:** Na maioria dos casos, o ajuste cuidadoso de controladores Proporcional-Integral-Derivativos (PID) convencional é usado para obter o mais alto desempenho para a mais ampla gama de aplicações. No entanto, em casos mais complexos, como sistemas com múltiplas entradas e múltiplas saídas (MIMO), os métodos de sintonia não extraem o melhor comportamento do sistema. Esse desempenho pode ser melhorado usando técnicas mais avançadas, como o PID de Ordem Fracionária (FOPID), que é baseado em cálculo fracionária. Meta-heurísticas de otimização podem ser utilizadas para resolver o problema de otimização de um controlador PID. O objetivo deste artigo é avaliar a eficácia do FOPID quando ele é ajustado usando meta-heurísticas, tais como Algoritmo Genético (GA), Otimização de Enxame de Partículas (PSO), Otimização de *Harris Hawks* (HHO), Algoritmo de Predadores Marinhos (MPA) e Otimização de Manta Ray (MRO). O *ball mill pulverizing system* (BMPS) foi utilizado como técnica de teste para avaliar o desempenho da sintonia. Mais de 50 rodadas diferentes e métricas estatísticas da função *fitness* foram aplicadas para avaliar o desempenho das meta-heurísticas. Ao final deste processo, ficou evidente que as técnicas são eficientes em geral, e o MPA foi a meta-heurística com menores valores de função objetivo na sintonia estruturada no sistema de pulverização do moinho de bolas, o que a torna a mais eficiente para o problema, enquanto que as duas técnicas clássicas GA e PSO foram as menos eficientes.

*Keywords:* Proportional-Integral-Derivative Controller; Systems Optimization; Multivariable Process; Optimization Metaheuristics.

*Palavras-chaves:* Controlador Proporcional-Integral-Derivativo; Otimização de Sistemas; Processo Multivariável; Meta-heurísticas de Otimização.

## 1. INTRODUÇÃO

O controlador proporcional-integral-derivativo (PID) tem sido amplamente utilizado no setor industrial há mais de um século (Zarei (2020)). Isto é em razão de sua estrutura simples, flexibilidade e eficiência, bem como utilização de recursos tais como a realimentação e capacidade de antecipar amostras de ações futuras, podendo efetivamente eliminar o erro de estado estacionário (Iruthayarajan and Baskar (2009)). O objetivo principal de um controlador PID é remover ou reduzir o erro entre um ponto de ajuste especificado e uma variável de processo medida em tempo real, tanto quanto possível. Por ser menos sensível a mudanças nos ganhos do controlador, o controlador PID de ordem fracionária (FOPID) é uma modificação do controlador PID original (Podlubny (1994)). Sendo baseado em cálculo fracionário, o que permite expandir a estrutura do controlador, dando mais configurações possíveis, tornando-o menos sensível a alterações nos ganhos do controlador.

Um processo de ajuste é necessário para se obter a melhor resposta do controlador, predefinindo seus parâmetros de entrada (Fister et al. (2016)). Diversos procedimentos de sintonia são documentados na literatura, sendo os métodos de Ziegler-Nichols (Åström and Hägglund (2004)) e Cohen-Coon (Cohen (1953)) os mais clássicos. Nas últimas décadas, contudo, outras técnicas começaram a ter mais popularidade, tais como redes neurais artificiais (Rivas-Echeverria et al. (2001)), abordagens baseadas em sistemas *fuzzy* (He et al. (1993)) e meta-heurísticas de otimização (Pareek et al. (2014)). Este último caso refere-se a algoritmos baseados em inteligência evolutiva e de enxame, que são capazes de encontrar uma solução quase ótima mais rapidamente do que outras abordagens devido à sua simplicidade e facilidade de implementação.

As meta-heurísticas de otimização são flexíveis, podendo ser aplicadas a vários tipos de problemas de otimização, incluindo problemas não-lineares, não-diferenciáveis, ou com ótimos locais. Essas otimizações, como mencionado, podem ser divididas em duas categorias: algoritmos evolutivos e algoritmos baseados em inteligência de enxame (Du et al. (2016)). O primeiro tipo são inspiradas em aspectos da evolução na natureza, como a sobrevivência do mais apto, reprodução e mutação genética, sendo o mais conhecido o algoritmo genético (GA) (Mirjalili (2019)). Já os algoritmos de inteligência de enxame reproduzem as interações de organismos em vivência coletiva, como por exemplo a otimização por enxame de partículas (PSO) (Kennedy and Eberhart (1995)), a otimização de gaviões-asa-de-telha (HHO) (Heidari et al. (2019)), o algoritmo de predadores marinhos (MPA) (Faramarzi et al. (2020)) e a otimização das raias manta (MRO) (Zhao et al. (2020)).

\* Esse estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de financiamento 001.

A contribuição deste estudo é comparar o desempenho de cinco meta-heurísticas de otimização: GA, PSO, HHO, MPA e MRO, no processo de sintonia de um controlador FOPID para um sistema de pulverização de moinho de bolas com um desacoplador em regime permanente. Essa aplicação é definido como um processo de múltiplas entradas e saídas múltiplas (do inglês *Multiple Input Multiple Output*, MIMO).

O restante desta pesquisa está organizada da seguinte maneira: os princípios dos controladores PID e FOPID, bem como o processo de sintonia, são descritos na Seção 2, que também inclui uma discussão sobre algoritmos de meta-heurística utilizados e uma descrição do estudo de caso. A Seção 3 explica a abordagem utilizada neste artigo. Os resultados são discutidos na Seção 4. As considerações finais são abordadas na Seção 5.

## 2. FUNDAMENTOS

Os fundamentos sobre os controladores PID, FOPID, e as meta-heurísticas são apresentados a seguir.

### 2.1 Controlador PID

O controlador PID convencional é composto de três ações de controle: proporcional, integral, e derivativa, as quais são combinadas para formar a seguinte estrutura:

$$K(s) = K_P + \frac{K_I}{s} + K_D \cdot s, \quad (1)$$

onde  $K(s)$  é a função de transferência do PID,  $s$  é o parâmetro do domínio da frequência, e  $K_P$ ,  $K_I$ , e  $K_D$  são o ganhos proporcional, integral, e derivativo, respectivamente. A parte proporcional fornece uma ação de controle proporcional ao erro da resposta do sistema, enquanto que a parte integrativa é usada para diminuir o erro de regime permanente, e a parte derivativa melhora a resposta em regime transitório (Ogata et al. (2010)).

Em um processo multivariável com  $n$  dimensões, com representação em matriz  $n$  por  $n$ , é necessária uma matriz de controle  $K(s)$  na qual cada elemento  $K_{ij}$  representa um controlador PID na forma de (1). Com isso, para um processo multivariável a matriz de controle com estrutura descentralizada é dada por

$$K(s) = \begin{bmatrix} K_{11}(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & K_{nn}(s) \end{bmatrix}. \quad (2)$$

### 2.2 Controlador FOPID

O controlador FOPID é uma extensão do controlador PID convencional, a qual acrescenta cálculo fracionário à estrutura do PID convencional, isto é, as ordens de

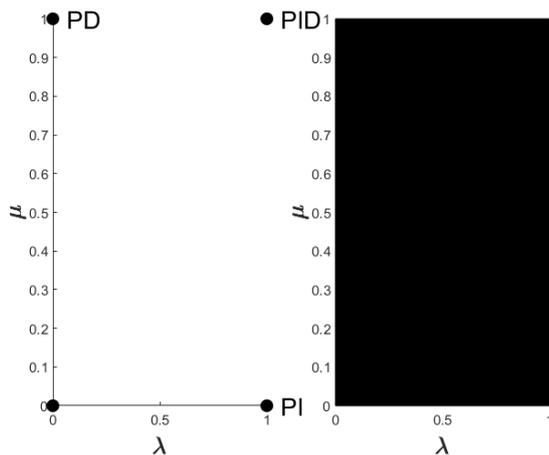


Figura 1. Regiões de atuação de um controlador PID convencional (esquerda) e um FOPID (direita).

integração e derivação passam a ter ordens fracionárias. Isso permite ações de controle mais suaves pois o FOPID é menos sensível à alterações nos ganhos do controlador (Podlubny (1994)). A estrutura do FOPID é dada por:

$$K(s) = K_P + \frac{K_I}{s^\lambda} + K_D \cdot s^\mu, \quad (3)$$

onde  $\lambda$  e  $\mu$  podem assumir valores menores ou iguais a 1, sendo esse último caso a estrutura convencional de um PID. Isso é ilustrado na Figura 1.

### 2.3 Meta-heurísticas

Foram escolhidas duas técnicas de otimização bastante usadas no contexto de sintonia de controladores PID (GA e PSO) e três técnicas mais recentemente desenvolvidas: HHO, MPA, e MRO. Para efeito de comparação, um ponto a ser observado é como as duas técnicas mais clássicas se comparam às mais recentemente propostas para solucionar o problema de otimização aqui demonstrado. Com isso, pode ser realizada a comparação não apenas entre meta-heurísticas mas também entre meta-heurísticas mais novas e as clássicas.

**Algoritmo Genético** O algoritmo genético (GA) é um algoritmo evolucionário inspirado pela teoria da evolução natural (Wald (2002)) e se baseia numa sequência de passos baseada em três fases: seleção, *crossover*, e mutação.

Na seleção são escolhidos 2 pares de cromossomos (soluções) em ponto flutuante para se reproduzirem, baseado no valor de *fitness*. O *crossover* gera novos cromossomos através daqueles escolhidos na seleção por meio de alguma estratégia. Por exemplo, a regra para gerar novos cromossomos pode ser juntar a primeira metade dos genes (parâmetros) de um pai com a segunda metade dos genes do outro. Nesse trabalho o *crossover* foi feito num cruzamento das melhores soluções com as piores soluções, de forma que os genes dos filhos fossem uma fração e seu complemento dos genes dos dois pais. A mutação modifica um ou mais genes das soluções geradas anteriormente com a intenção de criar diversidade, e o algoritmo repete os três passos por um certo número de gerações.

**Particle Swarm Optimization (PSO)** Desenvolvido primeiramente por Kennedy and Eberhart (1995), essa meta-heurística foi motivada pelo comportamento coletivo de enxames de peixes e pássaros, sendo uma técnica de inteligência de enxame baseada em população. A versão do PSO clássica adotada foi a proposta em Sun et al. (2016).

**Harris Hawks Optimization (HHO)** O HHO desenvolvido por Heidari et al. (2019) se baseia no comportamento de caça de aves conhecidas como *Harris Hawks*, ou Gaviões-Asa-de-Telha, como são chamados no Brasil. Presentes nas faunas estadunidense e brasileira, essa ave possui uma estratégia coletiva de caça chamada "pulo surpresa", na qual as aves vão cercando a presa e realizando voos ao seu redor para encurralá-la e cansá-la, até que finalmente é possível fazer sua captura. O algoritmo é baseado em uma fase de *Exploration* e *Exploitation*, sendo que na primeira fase os gaviões se empoleiram em árvores de forma a se preparar para o ataque, e na segunda eles realizam os voos de maneiras diferentes a depender da posição da presa para realizar o pulo surpresa.

**Marine Predators Algorithm (MPA)** O MPA (Faramarzi et al. (2020)) é inspirado pelo movimento de criaturas marítimas, utilizando para isso o voo de *Lévy* e o movimento browniano. Possuindo três fases, o algoritmo começa com uma fase de *Exploration* e termina com uma fase de *Exploitation*, tendo ainda uma fase de transição no meio delas. Na primeira fase as soluções se movimentam usando a forma Browniana, na terceira se movimentam usando voo de *Lévy*, e na fase de transição cada metade da população se movimenta usando um dos movimentos citados.

**Manta Ray Optimization (MRO)** O MRO (Zhao et al. (2020)) é uma meta-heurística bio-inspirada nas raias manta, se utilizando de três técnicas de forrageamento: em cadeia, em ciclone, e por cambalhota. Cada forma de forrageamento corresponde a uma fase do algoritmo. Em cadeia, essas arraia formam uma linha e as arraia macho se aderem nas costas das fêmeas, assim o plâncton (a melhor solução encontrada) que não é pego por uma arraia é pego pela arraia de trás. Em ciclone as raias se movimentam em espiral em direção ao plâncton, e no forrageamento por cambalhota as raias manta se aproximam do plâncton uma após a outra e se afastam fazendo uma cambalhota até uma nova posição. Dessa forma, o algoritmo atualiza as posições de cada solução na população em torno da melhor solução encontrada até o momento (plâncton).

### 2.4 Sistema de Pulverização por Moinho de Bolas

O processo utilizado para verificar a eficácia de cada meta-heurística foi o sistema de pulverização por moinho de bolas (do inglês *Ball Mill Pulverizing System*, BMPS), um processo multivariável e não-linear com leve variação no tempo (Cheng and Wang (2006)). Possui três entradas (velocidade de alimentação de carvão, abertura de ar reciclado, abertura de ar quente) e três saídas (pressão diferencial, pressão negativa de entrada, pressão de saída). Contudo o processo pode ser reduzido a um sistema com duas dimensões de entrada e saída se o sinal de vibração vertical da frente do transporte do moinho representar carregamento de carvão (Qiming et al. (2009)). Sendo a primeira e segunda saídas: temperatura de saída  $T$ , e

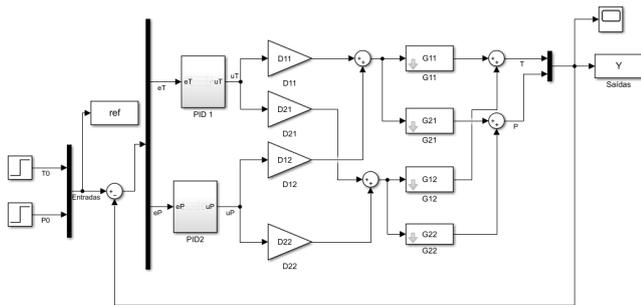


Figura 2. BMPS implementado no Simulink.

$K_{P1}$	$K_{I1}$	$K_{D1}$	$\lambda_1$	$\mu_1$	$K_{P2}$	$K_{I2}$	$K_{D2}$	$\lambda_2$	$\mu_2$
----------	----------	----------	-------------	---------	----------	----------	----------	-------------	---------

Figura 3. Vetor solução.

pressão de entrada negativa do sistema  $P$ . Dessa forma as duas entradas do sistema em malha aberta são a abertura de ar quente para controlar  $T$ , e a abertura de ar reciclado para controlar  $P$ .

Com isso o processo pode ser definido pela matriz

$$G(s) = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} \frac{3.5}{(80s+1)^3} & \frac{-0.14}{(60s+1)^2} \\ \frac{-2}{(8s+1)^2} & \frac{-0.18}{10s+1} \end{bmatrix}, \quad (4)$$

a qual não tem linha diagonal dominante. Um desacoplador é então usado para reduzir interações entre as variáveis (Menhas et al. (2012)):

$$D(s) = G^{-1}(0) = \begin{bmatrix} 0.1378 & -0.1538 \\ -2.1978 & -3.8462 \end{bmatrix}. \quad (5)$$

A estrutura do sistema implementado está na Figura 2. Ela é composta das duas entradas do processo  $T0$  e  $P0$ , representando a temperatura e pressão desejada para o sistema, os dois controladores FOPID, as funções de transferência para as duas matrizes ( $G(s)$  e  $D(s)$ ),  $eT$  e  $eP$  são os erros de temperatura e pressão,  $uT$  e  $uP$  são as ações de controle para os dois controladores, e  $ref$  e  $Y$  são os vetores de referência e da saída, respectivamente.

### 3. METODOLOGIA

Cada possível solução para o problema de otimização pode ser representada como um vetor de 10 posições, como ilustrado na Figura 3. A primeira metade do vetor representa os valores de ganhos e de ordens de integração e derivação para o primeiro controlador, enquanto que a segunda metade corresponde ao segundo controlador. Os limites mínimo e máximo dos valores foram 0 e 50 para os ganhos e 0 e 1 para as ordens de integração e derivação.

A função objetivo utilizada para medir a eficácia de cada controlador foi a soma da integral do tempo multiplicado ao erro (com relação a um degrau unitário) ao quadrado (do inglês *Integral Time Squared Error*, ITSE) das duas saídas do sistema, e é definida como

$$FObj(X) = \sum_{i=1}^2 \left( \int_{t=0}^T t \cdot e_i^2(t) dt \right), \quad (6)$$

onde  $t$  é o tempo com período amostragem 0.1,  $T$  é o tempo máximo de simulação, e  $e_i$  é o erro na entrada  $i$ .

Não há estrutura pré-definida para rodar testes com meta-heurísticas, para este trabalho, decidiu-se utilizar como estrutura para teste 50 como tamanho de população, e 100 como número de gerações. O ambiente computacional Simulink foi usado para rodar as simulações, enquanto que as meta-heurísticas foram implementadas no Matlab. O hardware utilizado foi um computador com processador *Intel(R) Core (TM) i7-7700HQ @ 2.80GHz 2.81GHz*, e 8GB de memória RAM (*Random-Access Memory*). Os algoritmos foram rodados 50 vezes para que fosse possível fazer uma análise estatística da sintonia, e foram computados os valores de função objetivo mínimo, máximo, médio, mediano, e desvio padrão.

### 4. RESULTADOS

Os valores obtidos são descritos na Tabela 1, onde os valores em negrito são os melhores valores. É possível ver que o MPA foi a técnica de otimização que melhor otimizou o controlador FOPID, com o valor mínimo de ITSE 304.63. O MPA alcança valores pequenos de ITSE com maior frequência que as outras técnicas, como é possível perceber pelos valores de média e mediana menores. O valor de desvio padrão de 0.03 demonstra que o MPA possui baixa variância nos seus resultados, o que reforça que ele converge mais fácil para o valor mínimo de ITSE, quando comparado com MRO, HHO, GA, e PSO, ordenados pelo valor mínimo de ITSE. As duas meta-heurísticas mais clássicas foram as menos eficientes para o problema proposto.

Tabela 1. Resultados estatísticos da sintonia

	MPA	MRO	HHO	GA	PSO
Mínimo	<b>304.56</b>	304.63	304.65	319.96	322.44
Máximo	<b>304.67</b>	705.79	525.58	341.48	424.70
Média	<b>304.59</b>	377.22	323.95	328.13	362.02
Mediana	<b>304.59</b>	306.48	305.54	328.32	355.82
Desvio Padrão	<b>0.03</b>	150.78	60.13	6.13	24.45

As Figuras 4 e 5 apresentam as respostas das duas saídas do processo quando sintonizadas pelas 5 meta-heurísticas usadas nesse estudo. Para a primeira saída o comportamento das curvas foi bem semelhante, com nenhuma técnica conseguindo convergir para o objetivo, enquanto que o PSO teve o maior *overshoot* e o MRO o menor. Na segunda saída foi onde o comportamento das curvas de resposta apresentaram as maiores diferenças, e é possível observar que o MPA se destacou pois teve um dos menores *overshoots* e foi o que convergiu mais rápido para a resposta, pouco depois dos 2 segundos de execução. Os desempenhos de MRO e GA também tiveram essa característica, mas sem a mesma eficácia, com um tempo de acomodação mais longo que o do MPA. Já HHO e PSO oscilaram bastante, além de terem tipo *overshoots* muito altos, o que definiu sua baixa eficácia perante as outras meta-heurísticas.

Os parâmetros dos melhores controladores encontrados por cada meta-heurística estão presentes na Tabela 2, e

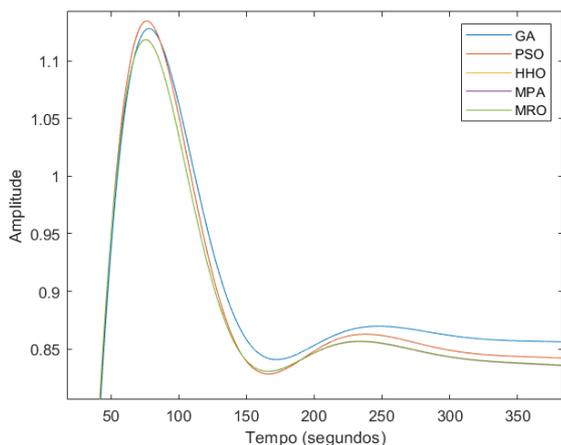


Figura 4. Primeira saída do processo quando controlado pelos melhores FOPID encontrados.

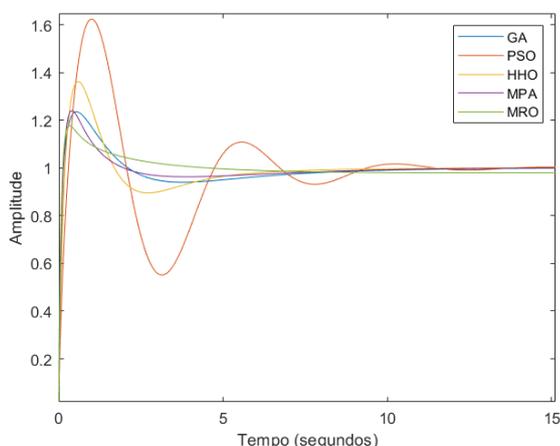


Figura 5. Segunda saída do processo quando controlado pelos melhores FOPID encontrados.

por ela é possível ver que nas condições presentes nesse trabalho a solução ótima tende a ter ganhos  $K_{P1}$  e  $K_{I1}$  nulos e os três ganhos do segundo controlador próximos ou iguais ao máximo valor possível. Com relação às ordens de integração e derivação, o consenso entre as técnicas de otimização aqui estudadas é que  $\mu_1$  deve estar próximo ou igual a 0.37, enquanto que  $\mu_2$  deve ser nulo ou próximo a 0. Os graus de integração nos dois controladores não definem se a resposta do sistema será eficiente ou não, sendo possível ver pela tabela que MPA e MRO divergiram de forma incisiva em  $\lambda_1$  e  $\lambda_2$ , mesmo obtendo ao final valores de ITSE próximos.

As respectivas ações de controle dos melhores FOPID em cada meta-heurística estão na Figura 6, onde é possível ver que a primeira ação de controle é bem semelhante entre as 5 técnicas de otimização, e que na segunda ação de controle existem diferenças quanto à amplitude máxima e mínima, com MPA e MRO, as 2 meta-heurísticas mais bem avaliadas possuindo os menores valores mínimos e estabilizando mais rapidamente. As energias dos 2 sinais de controle para cada meta-heurística estão na Tabela 3. O PSO apresentou o menor esforço de controle e o GA o menor.

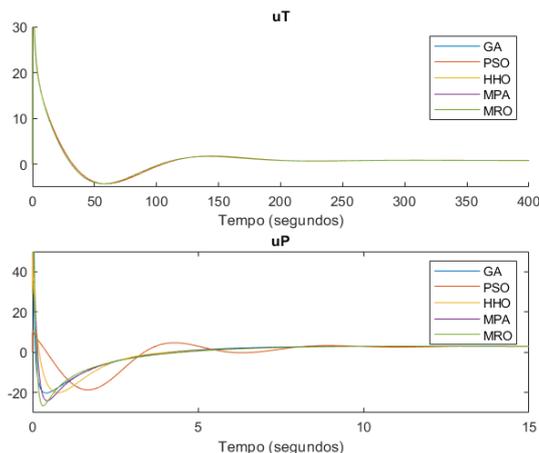


Figura 6. Ações de controle dos melhores FOPID encontrados por cada meta-heurística.

## 5. CONCLUSÃO

Nesse artigo, é apresentada uma comparação de desempenho de meta-heurísticas de otimização aplicadas para a sintonia de um controlador FOPID para controle de um processo de pulverização em um moinho, um processo com duas entradas e duas saídas. Dentro das condições estruturais de solução do problema, foi possível constatar que as técnicas de otimização foram capazes de encontrar desempenhos relativamente eficientes, com exceção da primeira saída, que não convergiu para a saída desejada, ainda que tenha estado próxima. Com relação à comparação entre meta-heurísticas, o MPA foi o otimizador de melhor eficácia, sendo capaz de encontrar não apenas o menor mínimo, como também os menores valores para as métricas estatísticas. As duas técnicas de otimização clássicas (GA e PSO) foram as menos eficientes para o problema tratado pois retornaram os maiores valores de ITSE ao longo das 50 rodadas. Ao observar com atenção os parâmetros dos melhores controladores encontrados é possível perceber que o mínimo global tende a certos valores estabelecidos, e pesquisas futuras que expandam o intervalo de valores possíveis dos parâmetros do controlador aqui demonstrado podem ir além e descobrir melhores configurações de FOPID para o problema apresentado.

Tabela 2. Melhores controladores encontrados.

	MPA	MRO	HHO	GA	PSO
$K_{P1}$	0	0	0.0005	1.13	0
$K_{I1}$	0	0	0	1.37	0
$K_{D1}$	50	50	50	49.93	46.22
$\lambda_1$	1	0	1	0.02	0
$\mu_1$	0.37	0.37	0.37	0.43	0.35
$K_{P2}$	49.99	50	29.07	42.92	0
$K_{I2}$	50	50	49.78	44.33	28.56
$K_{D2}$	49.99	50	16.18	26.03	11.25
$\lambda_2$	0.99	0	1	0.89	1
$\mu_2$	0.003	0	0.12	0.45	0

Tabela 3. Energia dos sinais de controle

	MPA	MRO	HHO	GA	PSO
uT	9958.7	9967.3	9957	10730	9144.3
uP	1952.1	2164.6	1728.9	2197	1668
Soma	11910.8	12131.9	11685.9	12927	10812.3

## REFERÊNCIAS

- Åström, K.J. and Hägglund, T. (2004). Revisiting the ziegler–nichols step response method for pid control. *Journal of Process Control*, 14(6), 635–650.
- Cheng, Q. and Wang, Y.h. (2006). The overview on the development of control techniques on intermediate storage bunker ball mill pulverizing system of power plant. *Journal of Shanghai University of electric power*, 22(1), 48–54.
- Cohen, G. (1953). Theoretical consideration of retarded control. *Transactions of the ASME*, 75, 827–834.
- Du, K.L., Swamy, M., et al. (2016). Search and optimization by metaheuristics. *Techniques and Algorithms Inspired by Nature*, 1–10.
- Faramarzi, A., Heidarinejad, M., Mirjalili, S., and Gandomi, A.H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377.
- Fister, D., Fister Jr, I., Fister, I., and Šafarič, R. (2016). Parameter tuning of pid controller with reactive nature-inspired algorithms. *Robotics and Autonomous Systems*, 84, 64–75.
- He, S.Z., Tan, S., Xu, F.L., and Wang, P.Z. (1993). Fuzzy self-tuning of pid controllers. *Fuzzy Sets and Systems*, 56(1), 37–46.
- Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872.
- Iruthayarajan, M.W. and Baskar, S. (2009). Evolutionary algorithms based design of multivariable pid controller. *Expert Systems with Applications*, 36(5), 9159–9167.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference On Neural Networks*, volume 4, 1942–1948. IEEE.
- Menhas, M.I., Fei, M., Wang, L., and Qian, L. (2012). Real/binary co-operative and co-evolving swarms based multivariable pid controller design of ball mill pulverizing system. *Energy Conversion and Management*, 54(1), 67–80.
- Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks*, 43–55. Springer.
- Ogata, K. et al. (2010). *Modern control engineering*, volume 5. Prentice hall Upper Saddle River, NJ.
- Pareek, S., Kishnani, M., and Gupta, R. (2014). Optimal tuning of pid controller using meta heuristic algorithms. In *2014 International Conference on Advances in Engineering & Technology Research (ICAE-TR-2014)*, 1–5. IEEE.
- Podlubny, I. (1994). Fractional-order systems and fractional-order controllers. *Institute of Experimental Physics, Slovak Academy of Sciences, Kosice*, 12(3), 1–18.
- Qiming, C., Yinman, C., Ruiqing, G., and Yong, Z. (2009). The forward nn-pid controllers based on chaos pso-bp hybrid optimization algorithms for decoupling control system of ball mill. In *International Conference on Artificial Intelligence and Computational Intelligence*, volume 1, 189–193. IEEE.
- Rivas-Echeverria, F., Rios-Bolivar, A., and Casales-Echeverria, J. (2001). Neural network-based auto-tuning for pid controllers. *Neural Network World*, 11(3), 277–284.
- Sun, J., Lai, C.H., and Wu, X.J. (2016). *Particle swarm optimisation: classical and quantum perspectives*. CRC press.
- Wald, S.E. (2002). The history of science and religion in the western tradition: an encyclopedia.
- Zarei, M. (2020). A physically based pid controller for the power maneuvering of nuclear reactors. *Progress in Nuclear Energy*, 127, 103431.
- Zhao, W., Zhang, Z., and Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, 87, 103300.