

PSO APLICADO À SINTONIA DO CONTROLADOR PI/PID DA MALHA DE NÍVEL DE UMA PLANTA DIDÁTICA INDUSTRIAL

LUCAS H. S. DE ANDRADE*, BRUNO L. G. COSTA*, BRUNO A. ANGÉLICO*

*Avenida Alberto Carazzai, 1640
Universidade Tecnológica Federal do Paraná (UTFPR)
Cornélio Procópio, Paraná, Brasil

Emails: lhsandrade1@gmail.com, brunolgcosta@gmail.com, bangelico@utfpr.edu.br

Abstract— This work aims to make a comparative analysis of two methods for tuning PID controllers: the classic method of Chien-Hrones-Reswick and Particle Swarm Optimization metaheuristic, applied to a level process in a didactic industrial plant. The system automation and control are implemented using CompactRIO hardware associated with LabVIEW graphic programming language.

Keywords— Metaheuristics and Optimization in Control Systems, PSO, LabVIEW, CompactRIO

Resumo— Este trabalho tem como objetivo fazer uma análise comparativa entre dois métodos de sintonia para controladores PID: o método clássico de Chien-Hrones-Reswick e a metaheurística Otimização por Enxame de Partículas, aplicados em uma malha de controle de nível de uma planta didática industrial. A automação e o controle do sistema são efetuados utilizando o *hardware CompactRIO*, associado à linguagem gráfica de programação *LabVIEW*.

Keywords— Metaheurísticas e Otimização em Sistemas de Controle, PSO, *LabVIEW*, *CompactRIO*

1 Introdução

Os controladores PID, atualmente, são uma das metodologias mais aplicadas em sistemas de controle industriais. Estima-se que mais de 90% das malhas destes processos empregam o controle PID (Knospe, 2006, p.30), devido a sua simplicidade estrutural e potencialidades de aplicação.

O projeto de um controlador PID consiste basicamente da parametrização de seus ganhos, através de métodos de sintonia apropriados. Métodos baseados em tabela, a partir da identificação paramétrica de uma dada planta, são comumente utilizados, tais como: método de Ziegler-Nichols, método de Cohen-Coon, método de Chien-Hrones-Reswick (CHR), entre outros (Astrom and Häggglund, 1995). Estes métodos nem sempre garantem uma sintonia adequada, porém podem resultar em um bom ponto de partida para uma sintonia refinada dos ganhos do controlador.

Se, por outro lado, um índice de desempenho for definido para o sistema, tais parâmetros podem ser obtidos, a partir da minimização desta função. Neste caso, métodos de otimização são aplicados.

Para índices de desempenho (funções custo) mal comportados, métodos analíticos nem sempre são factíveis. Sendo assim, metaheurísticas podem ser empregadas para resolver problemas de otimização relativamente complexos em tempo computacional aceitável (Valerdi, 2011).

Dentro deste contexto, este artigo apresenta uma aplicação prática do algoritmo de Otimização por Enxame de Partículas, PSO (*Particle Swarm Optimization*), na sintonia de um controlador PI/PID da malha de nível de uma planta

didática industrial, sendo comparado ao método clássico de sintonia CHR.

A automação e o controle da planta didática envolvida foram desenvolvidos utilizando o *software LabVIEW* em conjunto com o PAC (*Programmable Automation Controller CompactRIO*).

Resultados experimentais foram obtidos e são apresentados, de modo a validar a estratégia de otimização considerada.

2 Planta Didática de Controle

A planta didática industrial modificada da fabricante FESTO, utilizada neste trabalho, possui quatro malhas de controle: temperatura, pressão, nível e vazão (Almeida, 2012). Dentre elas, a malha de controle de nível foi considerada neste artigo e é comentada a seguir.

2.1 Malha de Controle de Nível

A malha de nível da referida planta didática é apresentada na Figura 1. Os componentes em destaque desta são:

- *CompactRIO*: controlador da planta;
- Bomba centrífuga: consiste no atuador do sistema de controle, realizando o deslocamento de fluidos nos reservatórios;
- Sensor de nível: sensor do tipo ultrassônico;
- Válvula de distúrbio: componente utilizado para a inserção de distúrbios no sistema, onde em condição de operação normal a mesma é mantida aberta.

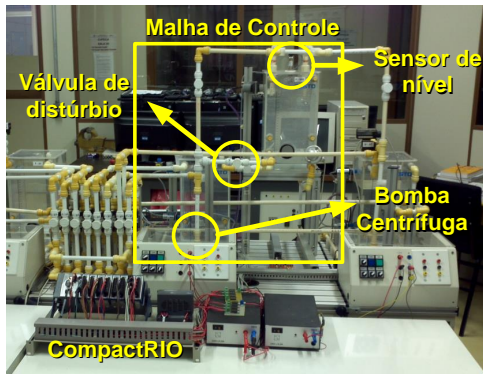


Figura 1: Malha de controle de nível

A seguir, alguns conceitos sobre o controlador *CompactRIO* e seu *software* de programação gráfica, *LabVIEW*, são comentados.

3 *CompactRIO*

O *CompactRIO* consiste em um PAC fabricado pela *National Instruments*, formado por um chassi reconfigurável baseado na tecnologia FPGA (*Field-Programmable Gate Array*).

Cada chassi aloja um processador de ponto flutuante embarcado com operação em tempo real, um FPGA de alta performance e módulos de E/S (entradas e saídas). Cada módulo de E/S é conectado diretamente ao FPGA, promovendo um rápido processamento de sinais. A conexão entre o FPGA e o processador embarcado é realizada através de uma rede PCI de alta velocidade (Bilik et al., 2008).

A programação do dispositivo, tanto do processador quanto do FPGA, é realizada por meio do *software LabVIEW* através de dois modos: *LabVIEW FPGA* e *Scan mode*, sendo ambos alocados no módulo *Real-Time* do programa.

3.1 *LabVIEW*

O *LabVIEW* é um *software* de programação gráfica, fabricado pela *National Instruments*, que emprega ícones ao invés de textos para criar aplicações.

A programação é efetuada na forma de um diagrama de blocos, que a torna ideal para desempenhar funções de testes e medições, análise de aplicação de dados e controle de instrumentos (Bitter et al., 2001). A Figura 2 ilustra o formato deste tipo de programação.

Todo programa elaborado no *LabVIEW* é chamado de Instrumento Virtual, VI (*Virtual Instrument*), devido a sua aparência e operação assemelharem-se às de instrumentos de medição e controle reais. Um VI é constituído de duas partes principais:

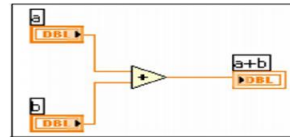


Figura 2: Programação em diagrama de blocos

- Diagrama de blocos: é a estrutura onde a programação é realizada, ou seja, é parte que contém o código fonte construído de forma gráfica pelo usuário;
- Painel frontal: é a interface com o usuário, onde são alocados todos os controles, gráficos e indicadores, o que resulta em uma tela que simula o painel físico de um instrumento.

Este *software* pode ser utilizado para adquirir, analisar e processar dados de instrumentos/equipamentos, e para a automação e controle destes (Larsen, 2011). Por meio do módulo *LabVIEW Real-Time*, as operações podem ser realizadas dentro de um tempo real e determinado, que são características da presente aplicação.

4 Sintonia de Controladores PID

Controladores PID podem ser vistos como componentes fundamentais para sofisticadas estratégias controle, sendo implementados quando conceitos básicos de controle não são suficientes para a obtenção de um desempenho aceitável (Visioli, 2006, p.1).

Devido ao seu enorme emprego, muitas pesquisas têm sido desenvolvidas, visando a busca por novas metodologias de identificação de sistemas e o aprimoramento de regras de sintonia (Knospe, 2006, p.30).

Em projetos de controle, inicialmente um método de identificação deve ser empregado para a correta modelagem matemática do sistema. Dentre vários presentes na literatura podem ser citados os métodos de Ziegler-Nichols (Z-N), Sundaresan/Krishnaswamy e Smith (Coelho and Coelho, 2004).

Após este procedimento é necessário efetuar a sintonia do controlador (ou seja, parametrizar seus respectivos ganhos). Métodos clássicos de sintonia, baseados em tabela, podem ser utilizados para tal finalidade, tais como os de Z-N, Chien-Hrones-Reswick (CHR) e Cohen-Coon (Astrom and Hägglund, 1995).

No entanto, estas técnicas clássicas nem sempre garantem uma boa sintonia dos controladores PID, resultando em alguns casos ações indesejáveis (tais como sobressinais e longos tempos de assentamento).

Uma alternativa que tem sido considerada em alguns trabalhos é o emprego de metaheurísticas para a otimização dos ganhos do controlador (Chen, 2007; Oi et al., 2008; Costa, 2013). Estes métodos ajudam a resolver problemas de otimização (comumente encontrados em engenharia) por simples operações de julgamento, reduzindo sua complexidade e fornecendo soluções próximas às melhores possíveis (Valerdi, 2011).

Este trabalho utiliza uma metaheurística de otimização, o PSO, com o objetivo de melhorar o ajuste dos ganhos, sendo usado para refinar os parâmetros do controlador, inicialmente obtidos por um método clássico, que neste trabalho é o CHR.

Nas seções a seguir são descritos detalhes com relação aos métodos abordados para a parametrização.

4.1 Método Clássico de Sintonia CHR

A malha de controle de nível foi modelada como um processo de primeira ordem com atraso de transporte, como descrito na equação (1):

$$G(s) = \frac{Ke^{-Ls}}{Ts + 1} \quad (1)$$

onde K , L e T , são, respectivamente, o ganho estático, o atraso de transporte e a constante de tempo do sistema (Coelho and Coelho, 2004).

Após a obtenção destes parâmetros para a malha considerada, o método clássico de sintonia de CHR foi empregado.

Dois critérios de projeto foram propostos por CHR: resposta rápida sem nenhum sobressinal (CHR 0%) e o outro com um máximo de 20% de sobressinal (CHR 20%). Neste trabalho utilizou-se o primeiro critério, sendo o seu projeto de controladores PI/PID descrito na Tabela 1.

Tabela 1: Sintonia PI/PID pelo Método CHR 0%

Controlador	K_p	T_i	T_d
PI	$\frac{0,35T}{KL}$	$1,2T$	0
PID	$\frac{0,6T}{KL}$	T	$0,5L$

4.2 Otimização por Enxame de Partículas

O algoritmo PSO foi desenvolvido com base no comportamento social de animais em busca de alimento, contando com conceitos muito simples de implementação (Kennedy and Eberhart, 1995).

Este algoritmo possui um conjunto de indivíduos (vetores solução) que evoluem no decorrer do processo por cooperação e competição (Shi and Eberhart, 1998, p.69). O método contém cálculos matemáticos primitivos, resultando em aceitável complexidade computacional, e que o torna muito eficaz na solução de problemas de otimização. No

fluxograma da Figura 3 é apresentada a estrutura de implementação do PSO.

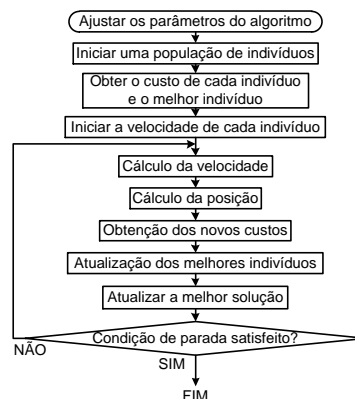


Figura 3: Fluxograma do algoritmo PSO

Inicialmente, o algoritmo inicializa uma população P de indivíduos/partículas. Cada partícula possui uma posição, vista como um vetor solução de N variáveis (geradas de maneira aleatória dentro de um intervalo estabelecido, X_{\max} e X_{\min}), potencial para a solução do problema. Portanto, a posição de uma i -ésima partícula é representada pela equação (2):

$$X_i = [X_{i,1}, X_{i,2}, \dots, X_{i,N}], \quad i = 1, 2, \dots, P \quad (2)$$

Cada indivíduo deste grupo também possui uma dada velocidade, assumida, inicialmente com o valor zero neste trabalho. Em um dado instante/iteração (it) as posições e as velocidades são descritas, respectivamente, pelas equações (3) e (4):

$$X_i(it) = [X_{i,1}(it), X_{i,2}(it), \dots, X_{i,N}(it)] \quad (3)$$

$$V_i(it) = [V_{i,1}(it), V_{i,2}(it), \dots, V_{i,N}(it)] \quad (4)$$

Sendo assim, cada um dos vetores solução são avaliados na função de custo proposta, visando à obtenção do melhor indivíduo do grupo.

Em seguida, o PSO evolui a posição e a velocidade de cada uma de suas partículas de acordo com as equações (5) e (6):

$$V_i(it + 1) = \omega V_i(it) + \phi_1 r_1 (X_i^{best} - X_i(it)) + \phi_2 r_2 (X_g^{best} - X_i(it)) \quad (5)$$

$$X_i(it + 1) = X_i(it) + V_i(it + 1) \quad (6)$$

onde: X_i^{best} e X_g^{best} representam, respectivamente, a melhor posição de um dado indivíduo e a melhor posição dentre todos os indivíduos; ϕ_1 e ϕ_2 são os coeficientes de aceleração individual e global, respectivamente; r_1 e r_2 são os parâmetros cognitivo (associado ao indivíduo) e parâmetro social (associado ao grupo), respectivamente (ambos gerados aleatoriamente entre $[0,1]$); e ω é o peso inercial, que controla a capacidade de exploração do enxame, responsável pela diversificação

de busca e para que o algoritmo não fique preso a um mínimo local.

A fim de evitar a extrapolação da velocidade do bando, uma restrição à velocidade foi aplicada:

$$\text{Se } V_i(it+1) \geq V_{\max} \rightarrow V_i(it+1) = V_{\max} \quad (7)$$

$$\text{Se } V_i(it+1) \leq V_{\min} \rightarrow V_i(it+1) = V_{\min} \quad (8)$$

Um cálculo exatamente como o que foi exposto nas equações (7) e (8) é feito também para restringir as posições das partículas (baseado em X_{\max} e X_{\min}).

Em seguida, os custos para cada uma das novas posições são obtidos. Se o custo atual de uma dada partícula for menor ao observado na iteração anterior, a posição corrente substitui a posição anterior da respectiva partícula. Após a verificação das melhores posições individuais de cada partícula, um mínimo global será identificado. Na sequência, o algoritmo continua com o seu processo iterativo, descrito anteriormente (como exposto na Figura 3), até que uma dada condição de parada seja satisfeita.

Com base no que foi comentado, os parâmetros fundamentais, que devem ser estipulados para a correta execução do algoritmo são: número de variáveis N , população de partículas P , o peso inercial ω , as constantes ϕ_1 e ϕ_2 , as restrições de velocidade (V_{\max} e V_{\min}) e os limites superior e inferior (X_{\max} e X_{\min}) para as variáveis.

5 Resultados Obtidos

Com base no que foi exposto anteriormente, conclui-se que o desafio da sintonia de controladores PID está diretamente relacionado com a obtenção dos ganhos K_p , K_i e K_d (ganhos proporcional, integrativo e derivativo, respectivamente). A estrutura de controlador PID adotada para este trabalho é a paralela, dada pela Figura 4. Com base nesta Figura, os ganhos K_i e K_d podem ser definidos: $K_i = K_p/T_i$ e $K_d = K_p.T_d$, onde T_i e T_d representam, respectivamente, o tempo integrativo e o tempo derivativo.

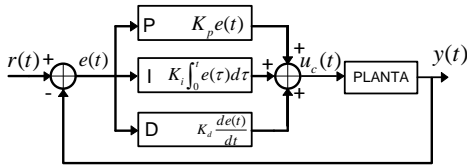


Figura 4: Estrutura PID paralela

A malha de nível empregada para a sintonia PID possui a seguinte função de transferência (identificada por Sundaresan/Krishnaswamy):

$$G(s) = \frac{2,677e^{-9,41s}}{75,29s + 1} \quad (9)$$

Dois tipos de sintonia foram efetuados: uma sintonia PI ($N=2$) e um PID ($N=3$). A priori, o método clássico CHR 0% foi utilizado para a sintonia PI/PID da malha de nível (Tabela 1). Uma vez que os ganhos foram obtidos por essa técnica, um refinamento foi proposto pela metaheurística PSO. Tal metaheurística foi implementada por linhas de código, com o auxílio do *software* MATLAB, sem considerar qualquer *toolbox*.

Um intervalo foi considerado para a geração aleatória dos vetores solução do PSO: de 0 a 50% a mais dos valores inicialmente obtidos pelo método CHR 0% (estes são os limites inferior e superior, X_{\min} e X_{\max} , respectivamente, de cada um dos ganhos PI/PID).

Uma função de custo foi considerada para as simulações do PSO. Tal função (equação (10)) apresenta duas parcelas de minimização: uma baseada no índice de desempenho ITAE (Dorf and Bishop, 2001, p.199-200), e a outra dedicada a minimizar a ação de controle ($u_c(t)$) do sistema, por meio do cálculo da variância do vetor da ação de controle.

$$F_{custo} = W_1 \left(\int_0^T t |e(t)| dt \right) + W_2 \text{var}(u_c(t)) \quad (10)$$

onde W_1 e W_2 são ponderações atribuídas às duas parcelas e $\text{var}(u_c(t))$, a variância do vetor $u_c(t)$.

Na Tabela 2 estão descritos todos os parâmetros adotados para a metaheurística PSO.

Tabela 2: Parâmetros do PSO

Parâmetros	PI	PID
Tempo de Simulação (s)	[0,100]	
Iterações	30	
P	20	
ω	0,75	
ϕ_1, ϕ_2	2	
V_{\max}, V_{\min}	5, -5	
W_1	0,5	1
W_2	0,5	2

As Tabelas 3 e 4 apresentam os resultados da sintonia realizada.

Tabela 3: Resultados do controle PI

Métodos	K_p	T_i	F_{custo}
CHR 0%	1,0458	90,3481	19,8278
PSO	1,3349	76,8597	10,8905

Tabela 4: Resultados do controle PID

Métodos	K_p	T_i	T_d	F_{custo}
CHR 0%	1,7933	75,2900	4,7050	35,9454
PSO	1,3311	77,6933	0,2603	26,8634

Tendo obtido os ganhos PI/PID por simulação, estes foram configurados na interface de usuário desenvolvida em ambiente *LabVIEW* (Figura 5).

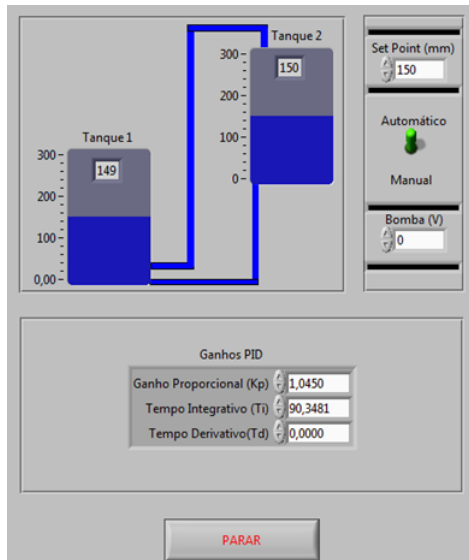


Figura 5: Tela de interface com o usuário

Sendo assim, três situações foram consideradas para analisar o controlador: a primeira é a referência inicial, definida em 150 mm; a segunda refere-se à aplicação de um novo degrau a partir de 60 segs, definido em 220 mm; e a terceira refere-se à abertura da válvula de distúrbio do sistema (em aproximadamente 115 segs). Os gráficos dos resultados dos controladores PI são demonstrados nas Figuras de 6 a 9 e os resultados dos controladores PID nas Figuras de 10 a 13, com as curvas de resposta e da ação de controle do atuador do sistema (no caso, a tensão aplicada na bomba centrífuga).

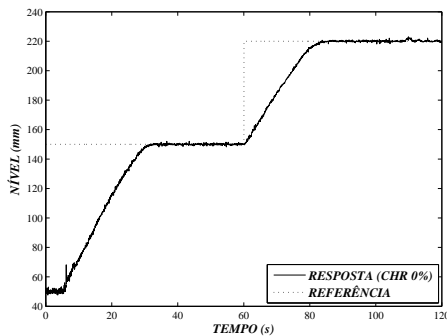


Figura 6: Controle PI: Resposta do sistema - CHR

6 Conclusões

Este trabalho apresentou uma comparação de desempenho entre o método clássico CHR e o algoritmo PSO na sintonia de controladores PI/PID na malha de nível de uma planta didática.

Pode-se verificar que, quando efetuada a sintonia PI, ambas as técnicas obtiveram parâmetros aceitáveis para o controlador, uma vez que solucionaram o problema e tiveram praticamente o

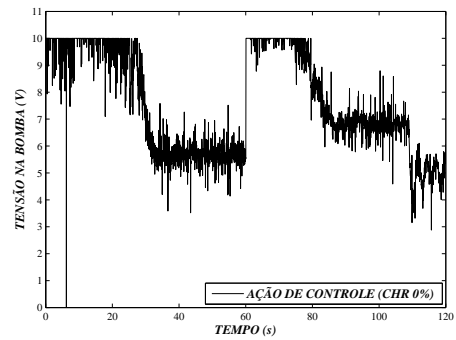


Figura 7: Controle PI: Ação de Controle - CHR

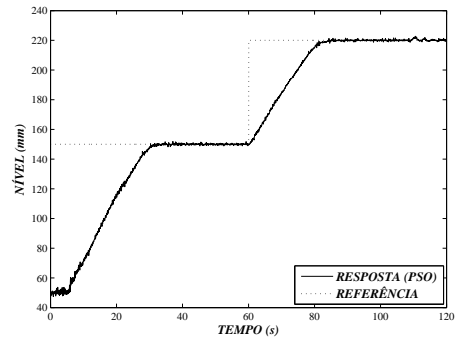


Figura 8: Controle PI: Resposta do sistema - PSO

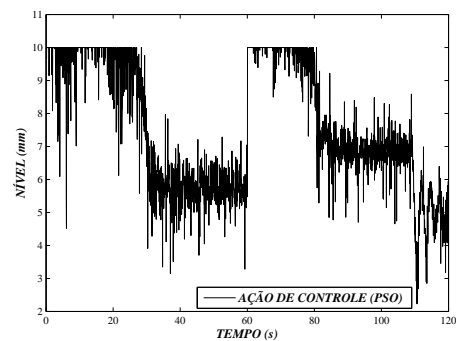


Figura 9: Controle PI: Ação de Controle - PSO

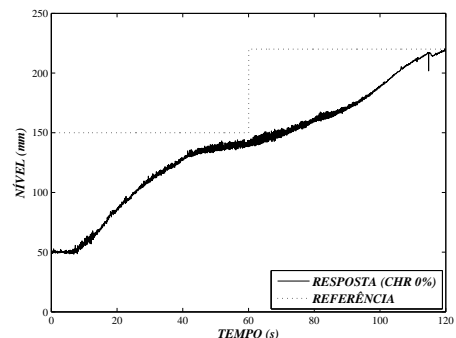


Figura 10: Controle PID: Resposta do sistema - CHR

mesmo desempenho. No entanto, para a sintonia PID, verificou-se que com o método clássico, o controlador do sistema se comportou de forma indesejada, ao passo que com o PSO, a sintonia

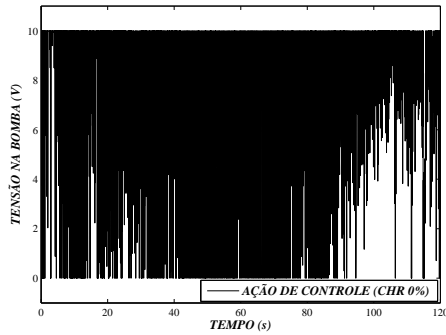


Figura 11: Controle PID: Ação de Controle - CHR

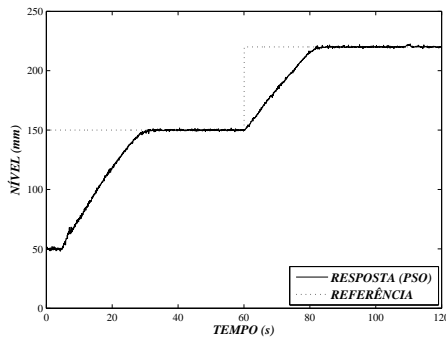


Figura 12: Controle PID: Resposta do sistema - PSO

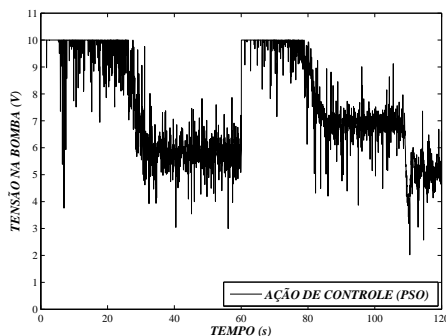


Figura 13: Controle PID: Ação de Controle - PSO

foi efetuada de forma a garantir um bom desempenho.

Com isso, pode-se concluir que a abordagem metaheurística representa uma forma alternativa para a sintonia do controlador PID, quando a sintonia por métodos clássicos não resulta em desempenho adequado.

Referências

Almeida, J. P. L. S. d. (2012). *Automação de uma Planta Didática de Sistemas de Controle*, 122f. Trabalho de Conclusão de Curso (Graduação) — Curso Superior de Tecnologia em Automação Industrial. Universidade Tecnológica Federal do Paraná: Cornélio Procópio.

Astrom, K. J. and Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*, 2 edn, Instrument Society of America, Research Triangle Park.

Bilik, P., Koval, L. and Hajduk, J. (2008). Compactrio embedded system in power quality analysis, *International Multiconference on Computer Science and Information Technology (IMCSIT)*, pp. 577–580.

Bitter, R., Mohiuddin, T. and Nawrocki, M. (2001). *LabVIEW Advanced Programming Techniques*, CRC Press, USA.

Chen, S.-F. (2007). Particle swarm optimization for pid controllers with robust testing, *International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 956–961.

Coelho, A. A. R. and Coelho, L. d. S. (2004). *Identificação de Sistemas Dinâmicos Lineares*, UFSC, Florianópolis.

Costa, B. L. G. (2013). *Sintonia de Controladores PID via Metaheurísticas de Otimização*, 86f. Trabalho de Conclusão de Curso (Graduação) — Curso de Engenharia Industrial Elétrica: Ênfase em Eletrotécnica. Universidade Tecnológica Federal do Paraná: Cornélio Procópio.

Dorf, R. C. and Bishop, R. H. (2001). *Sistemas de Controle Modernos*, 8 edn, LTC, Rio de Janeiro.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Vol. 4, pp. 1942–1948.

Knospe, C. (2006). Pid control, *IEEE Control Systems Magazine*, Vol. 26, pp. 30–31.

Larsen, R. W. (2011). *LabVIEW for Engineers*, Prentice Hall, USA.

Oi, A., Nakazawa, C., Matsui, T., Fujiwara, H., Matsumoto, K. and Nishida, H. (2008). Pid optimal tuning method by particle swarm optimization, *SICE Annual Conference*, pp. 3470–3473.

Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer, *The 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, Anchorage, pp. 69–73.

Valerdi, R. (2011). Heuristics for systems engineering cost estimation, *Systems Journal, IEEE* 5(1): 91–98.

Visioli, A. (2006). *Practical PID Control*, Springer-Verlag, London.