

PI controller Optimization for a Heat Exchanger Through Metaheuristic Bat Algorithm, Particle Swarm Optimization, Flower Pollination Algorithm and Cuckoo Search Algorithm

N. C. Damasceno, O. G. Filho

Abstract— This paper presents the modeling and computer simulation of a control system for a shell and tube heat exchanger, using Bat Algorithms, Particle Swarm Optimization, Flower Pollination Algorithm and Cuckoo Search Algorithm. To evaluate the performance of different methods of tuning, we compared the values of the transient of the response to step in eight mesh settings generated. It has also established a comparison between these two types of mesh using the performance indices proposed in the literature, with optimized system by Bat Algorithms got the best values of transient in relation to the Particle Swarm Optimization, Cuckoo Search Algorithm and Flower Pollination Algorithm. Performance indices FPA and PSO obtained better results.

Keywords— PI controller, Bat Algorithm, PSO, Flower Pollination Algorithm, Heat Exchanger, Optimization.

I. INTRODUÇÃO

OS TROCADORES de calor são equipamentos utilizados nas indústrias química e petroquímica nas situações em que é desejável a transferência de calor entre fluidos de temperaturas distintas [1]. Atualmente, existe uma grande variedade daqueles equipamentos dimensionados conforme a necessidade de cada aplicação. Neste estudo, apresenta-se a modelagem de um trocador de calor de casco e tubos do tipo Single-Input, Single Output (SISO) proposto por [2].

Dados publicados por [3] mostram que a determinação dos parâmetros K_p e K_i do controlador PI pelo método de Ziegler-Nichols gera resultados indesejados como sobressinal (M_p) alto e tempo de acomodação (T_s) muito longos, o que na prática pode implicar em desgastes na estrutura do trocador de calor e atrasos desnecessários na estabilização do processo. Uma das alternativas para melhorar estas características é a otimização de parâmetros utilizando-se as metaheurísticas.

Um estudo comparativo publicado por [4] e [5] demonstra que a metaheurística *Bat Algorithm* (BA) tem alcançado os melhores resultados, quando comparada com outros algoritmos difundidos no meio acadêmico. Em outro estudo publicado por [6] os autores demonstram a otimização dos parâmetros de um controlador PID aplicado a um regulador de voltagem automático (AVR) utilizando-se desta metaheurística, em [7] foi proposto um controlador Proporcional-Integral (PI) adaptativo utilizando as técnicas *Extreme Machine Learning* e BA para o controle ótimo de um motor síncrono de ímãs

permanentes. Em ambos os casos houve melhoria de desempenho do elemento controlador e da planta.

Em 2015, o artigo [8] apresentou um estudo dos 5 mais recentes algoritmos bioinspirado para otimizar um SMV (*Support Vector Machine*), propostos para automaticamente selecionar melhor a combinação dos parâmetros livres. Os resultados apresentados neste trabalho mostraram que os modelos propostos possuem taxa de convergência rápida em estágios iniciais das iterações. Eles conseguiram uma melhor precisão comparando preços e previsão de tendências.

O artigo [9] apresenta um projeto de um controlador PID para controle de velocidade de um motor DC usando o algoritmo *Flower Pollination* (FPA). Os resultados mostram ótimo comportamento do método proposto em comparação com o algoritmo *Cuckoo Search* (CS) e o *Artificial Bee Colony*.

Diante do exposto, e tomando como função de custo a resposta da planta do trocador de calor, os métodos heurísticos [8][10][11] fornecem uma alternativa intuitiva e eficiente para estimar os parâmetros de sintonia que venham a melhorar o desempenho da planta. Neste estudo é apresentado uma visão geral do PSO, BA, FPA e CS cujas eficiências são avaliadas em termos das características transientes da resposta e dos índices de desempenho em diferentes estratégias de controle de um controlador do tipo PI.

II. O TROCADOR DE CALOR

O trocador de calor de casco e tubos é uma das variedades mais utilizadas nos processos industriais. Fatores como a ampla faixa operacional de pressão (do vácuo a 600 lb/pol²), temperatura (−230 a 550°C), robustez, confiabilidade, simplicidade de construção e baixo custo de manutenção elegem-no como principal candidato nas situações em que é necessário utilizar dispositivos de transferência térmica [12].

A Fig. 1 mostra um esquema simplificado de um trocador de calor de casco e tubos típico [13]. Neste equipamento, o fluido do processo entra no cabeçote pelo bocal de entrada e é direcionado para o interior dos tubos organizados em um espelho fixo. Quando o fluido do processo se move no interior do trocador de calor este troca energia interna com outro fluido que entra pelo bocal de entrada do casco, resfriando ou aquecendo, conforme a aplicação, até alcançar o bocal de saída dos tubos. Neste processo não há contato direto entre os dois

fluidos; a troca de calor ocorre por efeito de condução térmica.

III. MODELAGEM DO SISTEMA DE CONTROLE

O objetivo principal do sistema de controle proposto é manter a temperatura do fluido do processo que sai do trocador de calor conforme o valor estabelecido no *setpoint* na presença de perturbações em sua vazão de entrada. Neste estudo serão utilizados dados experimentais obtidos em [2].

O elemento sensor da malha de controle consiste de um termopar acoplado no laço de realimentação do sistema que será responsável pela transdução e transmissão do valor da temperatura do trocador de calor em um sinal padrão de 4-20 mA, cujo controlador recebe o sinal e o compara com o *setpoint* da variável de processo, computa o valor de ajuste e modula a ação de controle de acordo com a configuração de seu algoritmo interno.

O sinal de controle é enviado para o conversor I/P da válvula controladora que efetua a conversão do sinal padrão 4-20 mA para o padrão 3-15 psi, manipulando o seu atuador interno até que a variável controlada convirja ao parâmetro estabelecido pelo *setpoint*. A Fig. 2 mostra um arranjo básico do sistema de controle proposto [2].

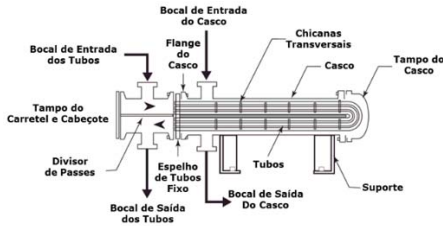


Figura 1. Esquema simplificado de um Trocador de calor de casco e tubos.

O trocador de calor tem como resposta a vazão de vapor $qm(s)$ injetada no bocal de entrada do casco um ganho de $50^\circ C/(kg/s)$ e uma constante de tempo de 30 segundos. O processo $G_o(s)$ apresenta a seguinte função de transferência:

$$G_o(s) = \frac{50}{30s+1}.$$

A resposta do trocador de calor à temperatura do fluido injetado no bocal de entrada dos tubos possui módulo igual a $1^\circ C/^\circ C$ e constante de tempo de 30s. Esta perturbação pode ser sintetizada pela seguinte função de transferência:

$$G_d(s) = \frac{1}{30s+1}.$$

A válvula de controle possui uma capacidade máxima de vazão da ordem de 1.6 kg/s, sendo intrinsecamente linear, e sua constante de tempo é da ordem de 3s. A faixa de trabalho da válvula é de 3 a 15 psi, o ganho do conversor I/P da válvula K_v e sua respectiva função de transferência G_v podem ser calculados conforme demonstrado a seguir: o conversor I/P possui ganho constante $\frac{\Delta P}{\Delta I} = \frac{(15-3)psi}{(20-4)mA} = \frac{12}{16} = 0.75 psi/mA$.

Calcula-se o ganho da válvula $K_v = \frac{1.6(kg/s)}{(15-6)psi} = \frac{1.6}{12} = 0.13 kg/s \cdot psi^{-1}$. Assim, a função de

transferência da válvula é representada por:

$$G_v(s) = 0.75 \cdot \frac{0.13}{3s+1}.$$

O sensor de temperatura escolhido opera na faixa compreendida entre $50^\circ C$ e $150^\circ C$, com constante de tempo de 10s. O cálculos de ganho K_s e sua função de transferência

$$G_s(s) \text{ são demonstrados por } K_s = \frac{(20-4)mA}{(150-50)^\circ C} \text{ e}$$

$$G_s(s) = \frac{0.16}{(10s+1)^\circ C}, \text{ respectivamente.}$$

Um fator de escala $K_r = 0.16 mA/^\circ C$ é multiplicado ao *setpoint* para que este amplifique o sinal de entrada do sistema e a resposta do sistema convirja adequadamente ao valor do *setpoint*.

Para realizar o controle efetivo da temperatura do fluido de saída é adicionado à malha do sistema um controlador do tipo PI. Este controlador foi proposto por apresentar vantagens como a simplicidade de utilização, facilidade de implementação, boa eficiência e extensa aplicabilidade [14], sendo um dos tipos de controlador mais utilizados no mundo.

A equação a seguir descreve a função de transferência do controlador PI na forma iterativa, a qual é a representação comumente adotada nos controladores comerciais:

$$C_P I(s) = K_p + K_i \cdot \frac{1}{s}, \text{ onde } K_i = \frac{K_p}{T_i}.$$

A Fig. 3 mostra o diagrama de blocos do sistema de controle projetado no ambiente de simulação Matlab, que engloba todas as funções de transferência relacionadas ao processo e ao controlador, interligadas de acordo com a lógica de controle por retroalimentação ou *feedback control*.

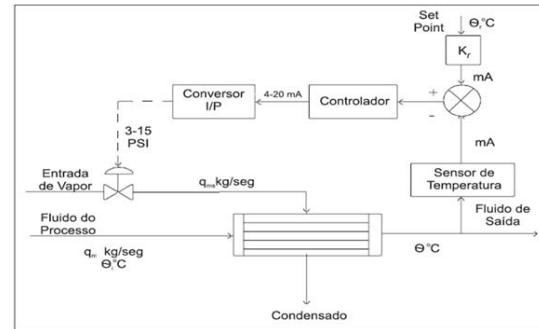


Figura 2. Arranjo básico do sistema de controle.

Para estimar os valores de ganho K_u e período P_u de amortecimento críticos é necessário encontrar primeiramente a função característica do sistema, que é feito substituindo-se a função de transferência do controlador por um elemento de ganho estático na malha de controle. Em seguida, obtêm-se a função de transferência de malha fechada resultando em

$$G_2(s) = \frac{4.875 K_c r}{900s^3 + 400s^2 + 43s + 0.798 K_c r + 1}.$$

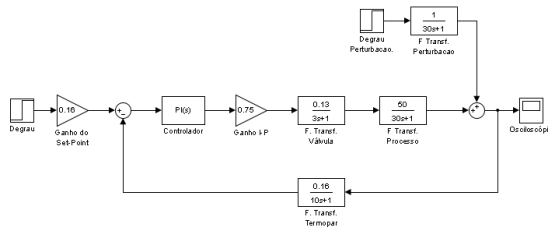


Figura 3. Diagrama de blocos do sistema de controle em modo de retroalimentação.

A partir da função $(900s^3 + 420s^2 + 43s + 0.798K_c r + 1 = 0)$ e aplicando o critério de estabilidade de Routh encontra-se $K_u = 23.8$. Na função característica, substituindo a variável s por $j\omega$ são determinados os valores de frequência de amortecimento e período críticos $\omega_u = 0.218$ e $P_u = 28.79$. Estes valores serão utilizados na estimativa dos parâmetros K_p e K_i pelo método de Z-N na seção VI deste artigo.

IV. CONTROLADOR DE ALIMENTAÇÃO AVANTE

Para atenuar os efeitos da perturbação na vazão de entrada do trocador de calor $qm(s)$ será modelado um controlador de alimentação avante ou *feedforward controller*. A vantagem de se utilizar este controlador é que ele não acrescenta malha de realimentação secundária no sistema, comportando-se como um elemento de malha aberta [15]. Os passos a seguir mostram como é feito o cálculo da função de transferência.

Temos que: $G_p(s) = \frac{4.875}{90s^2 + 33s + 1}$, $G_d(s) = \frac{1}{30s + 1}$. A função de transferência do controlador de alimentação avante $F(s)$ pode ser calculada por $F(s) = -\frac{G_d}{(G_p(s))^{-1}(\lambda s + 1)}$, em que $0 < \lambda \leq 1$ é necessário para tornar $F(s)$ causal, escolhendo $\lambda = 1$ teremos $F(s) = \frac{-18.461s^2 - 6.769s - 0.205}{30s^2 + 31s + 1}$.

V. ALGORITMOS DE COMPUTAÇÃO BIOINSPIRADA

Nesta seção são apresentados quatro algoritmos de computação bioinspirada utilizados para sintonia do PI.

A. PARTICLE SWARM OPTIMIZATION

O *Particle Swarm Optimization* (PSO), também chamado de enxame de partículas, é uma inteligência inspirada no comportamento dos bandos de pássaros [16]. Os pássaros buscando alimentos e a interações entre as aves ao longo dos voos são mecanismos de otimizações.

O PSO possui uma população de partículas em que cada uma representa uma solução para o problema. No nosso caso, um K_p , K_i . Cada partícula pode ser representada por um objeto que possui associado a ele um vetor posição, e um vetor velocidade, aonde a posição em relação ao espaço de busca e a velocidade guia a posição das partículas durante a execução do processo.

O algoritmo básico do PSO consiste na equação de velocidade e posição, respectivamente:

$$v_i(k+1) = w \cdot v_i(k) + c_1 r_1 \cdot (pbest_i - x_i(k)) + c_2 r_2 \cdot (gbest - x_i(k))$$

$$x_i(k+1) = x_i(k) + \Delta t \cdot v_i(k+1)$$

onde $i = 1, \dots, npop$, $npop$ é o tamanho da população. $pbest$ (*personal best*) e $gbest$ (*global best*) são, respectivamente a melhor posição obtida por uma partícula em uma determinada posição e de toda população em uma determinada vizinhança; w é a constante de inércia; c_1 é um fator social; c_2 é o fator cognitivo; r_1 e r_2 são números aleatórios gerados usando uma distribuição uniforme no intervalo $[0,1]$; e $\Delta t = 1$. Nas simulações computacionais apresentadas na seção VIII utilizou-se um fator social de 1.2 e fator cognitivo de 0.12. O valor de w foi 0.9.

B. BAT ALGORITHM

Proposto por [10], o Algoritmo Morcego ou *Bat Algorithm* é uma metaheurística baseada na técnica de eco localização de algumas espécies de morcego quando em voos noturnos. Neste modelo computacional é gerado um conjunto de morcegos codificado na forma de vetor, representando cada um uma solução candidata. O objetivo é alcançar a presa, ou seja, a solução ótima que minimiza a função de custo.

Inicialmente, todos os n morcegos x_i ($i = 1, \dots, n$) são inicializados com os seguintes parâmetros: Taxa de pulso r_i , velocidade $\vec{v}_i = 0$, amplitude A_i , frequência f_i e posição \vec{x}_i aleatória. Para cada instante t a velocidade e a posição são atualizadas por, respectivamente:

$$v_i^j(t) = v_i^j(t-1) + [x_{cgbest}^j - x_i^j(t-1)]f_i$$

$$x_i^j(t) = x_i^j(t-1) + v_i^j(t)$$

$f_i \in [f_{min}, f_{max}]$ é atualizada e ponderada pela variável $\beta \in [0,1]$, que representa um número randômico gerado a partir de uma distribuição uniforme. A variável x_{cgbest}^j representa a atual melhor solução global para uma variável de decisão j , o que é conseguido comparando todas as soluções fornecidas por n morcegos.

O algoritmo realiza a busca local implementada na forma de passeio aleatório (*random walk*) com o fim de explorar o domínio de soluções candidatas do problema: $x_i^{novo} = x_i^{velho} + \varepsilon A_m$. Onde ε é um número aleatório extraído de uma distribuição uniforme e A_m representa a média da amplitude de todos os morcegos no instante t .

O algoritmo para quando r_i alcança um valor mínimo preestabelecido ou quando um limite máximo de iterações é alcançado, chamadas condições de parada. Neste trabalho, utilizou-se uma amplitude e taxa de pulso inicial de 0.5. As frequências máxima e mínima formam, respetivamente, 2 e 0. A população do algoritmo é semelhante ao PSO, ou seja, cada morcego representa um K_p e K_i .

C. FLOWER POLLINATION ALGORITHM

Flower Pollination Algorithm (FPA) é um algoritmo inspirado no método de polinização das flores, proposto por [17]. Os polinizadores podem voar longas distâncias, logo são

considerados polinizadores globais, e seus comportamentos podem ser descritos pela distribuição de probabilidade de Lévy.

A implementação do algoritmo utilizando a distribuição de Lévy possui duas regras básicas: 1 – A escolha da direção deve ser aleatória. A geração de uma direção pode ser feita utilizando uma distribuição uniforme; e 2 – A geração dos passos deve obedecer a distribuição de Lévy.

Yang idealizou as seguintes regras [17]: 1 - Polinização biótica e cruzada são consideradas como um processo de polinização global com os transportadores de pólen executando voos de Lévy; 2 - Polinização abiótica e autopolinização são consideradas como polinização local; 3 - Lealdade a uma flor pode ser considerada como tendo uma probabilidade de reprodução proporcional à similaridade das duas plantas envolvidas e; 4 - Polinização local e global são controladas por uma probabilidade $p \in [0,1]$. Devido à proximidade física e outros fatores como o vento, a polinização local pode ter uma significativa parte nas atividades gerais de polinização.

O melhor indivíduo representado por g_* . A primeira regra, juntamente com a lealdade de uma flor, pode ser descrita matematicamente como: $x_i^{t+1} = x_i^t + L(x_i^t - g_*)$; em que x_i^t é o pólen i no vetor de soluções x_i na iteração t , e L a força da polinização, cujo valor é gerado pela distribuição de Lévy. Os insetos podem se mover a uma grande distância com vários passos de distância, podendo usar um voo de Lévy para imitar esta característica. Ou seja, $L > 0$ a partir de uma distribuição de Lévy

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0),$$

onde $\Gamma(\lambda)$ é a função *Gamma* e s_0 um passo mínimo.

A polinização local (regra 2) e a lealdade a uma flor é representado como $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$, onde x_j^t e x_k^t são pólenes de diferentes plantas da mesma espécie na mesma iteração. Isto basicamente imita a lealdade de uma flor em uma vizinhança limitada. Caso x_j^t e x_k^t fossem da mesma espécie ou selecionadas da mesma população, isto tornaria um passeio aleatório local (*local random walk*) tomando-se ϵ a partir de uma distribuição uniforme.

Utiliza-se uma probabilidade de troca de acordo com a regra 4, ou probabilidade de proximidade p para chavear entre fazer polinização global ou intensificar a polinização local. O motivo deste parâmetro é que a maioria das atividades de polinização pode ocorrer tanto em escala local ou global. Em termos práticos, flores adjacentes ou não tão distantes da vizinhança são mais suscetíveis a polinização em comparação com aquelas mais distantes. Neste artigo utilizou-se p igual a 8 e cada flor representa um K_p e K_i .

D. CUCKOO SEARCH ALGORITHM

O algoritmo *Cuckoo Search* (CS) foi proposto por [18]. Ele é baseado no comportamento reprodutivo de certas espécies de cucos. Ao invés dos cucos criarem seus filhotes as mães colocam seus ovos no ninho das fêmeas de outras espécies. Porém, se a mãe anfitriã descobre que no meio de seus ovos há ovos intrusos ela simplesmente descarta estes ou abandona o ninho.

No contexto do algoritmo, cada ninho contém apenas um ovo, ou seja, cada ninho é um conjunto de soluções candidata contendo uma solução na busca, o qual no nosso caso é representado por K_p e K_i . E ele apresenta três regras: 1 – Cada pássaro bota um ovo de cada vez em um ninho escolhido aleatoriamente; 2 – Os melhores ninhos são transferidos para as próximas gerações e; 3 – O número de ninhos anfitriões é fixo, e o ovo botado pelo cuco é descoberto pela anfitriã com uma probabilidade p_a .

Na regra 1 o ovo antigo é substituído por outro gerado através do voo de Lévy aplicado sobre o ovo anterior. A regra 2 é responsável por garantir a convergência da solução através do elitismo.

Da mesma forma que no FPA, o grande contribuidor de desempenho é o voo de Lévy, que é um tipo de caminhada aleatória responsável por explorar o espaço de busca. Durante a nova geração de soluções $x(t+1)$ o voo de Lévy é calculado pela seguinte expressão [19]: $x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\gamma)$, em que α é o tamanho do passo relacionado com a escala do problema em questão. Na maioria dos casos, usa-se $\alpha = 1$ (que é o nosso caso) e $\text{Lévy} \sim u = 1^{-\gamma}$, ($1 < \gamma \leq 3$). O operador \oplus é um multiplicador matricial. A taxa de descoberta p_a utilizado é 0.25.

De modo geral, o algoritmo inicia com uma população de n ninhos. Enquanto o número máximo de geração ou critério de parada não for atingido faz-se os seguintes procedimentos: toma-se um cuco aleatoriamente através do voo de Lévy; em seguida, avalia-se a qualidade P_i (ISTSE); escolhe-se um ninho j entre os n aleatoriamente. Se P_i for menor que P_j substitui j pela nova solução. Uma fração p_a dos piores ninhos são abandonados e novos construídos; mantém-se melhores soluções ou ninhos com soluções de qualidade e; ordena-se as soluções e acha-se a melhor atual. No caso deste trabalho, o critério de parada é o número máximo de geração.

VI. MÉTODO DE SINTONIA

O método *Ziegler-Nichols* (Z-N) de sintonia é um dos mais conhecidos e aplicados nos controladores atuais [20]. Nesta regra, os parâmetros K_p , K_i e K_d do controlador são computados a partir dos valores de ganho K_u e período P_u críticos utilizando-se de fórmulas previamente estabelecidas [21].

O segundo método abordado neste trabalho é o *Skogestad Internal Model Control* (SIMC) [22]. Inicialmente sintetiza-se a resposta do sistema à excitação em um modelo de primeira ordem com tempo morto: $G(s) = \frac{ke^{-\theta s}}{\tau s + 1}$. O valor de k representa o ganho do processo, τ a constante de tempo dominante e θ o atraso de tempo efetivo. Estes valores são estimados a partir da análise gráfica. A partir daí calcula-se a constante de tempo de malha fechada τ_c e obtêm-se os parâmetros do PID. A vantagem deste método é a sua versatilidade, cujos parâmetros podem ser ajustados para se obter um controle firme (mais rápido) ou suave.

VII. ÍNDICE DE DESEMPENHO

A análise da resposta do sistema baseada nos índices de desempenho fornece aos profissionais da área de controle evidências quantizadas sobre o grau de otimização do elemento controlador [23]. Os principais índices de desempenho são: $IAE = \int_0^T |e(t)|dt$, $ISE = \int_0^T e^2(t)dt$, $ITAE = \int_0^T t|e(t)|dt$, $ITSE = \int_0^T te^2(t)dt$ e $ISTSE = \int_0^T t^2e^2(t)dt$. Cada um é calculado no intervalo de tempo $0 \leq t \leq T$, em que T é o tempo necessário para que a resposta da planta estabilize um valor final. Em sistemas de segunda ordem é uma prática comum escolher $T = T_s$.

Controladores sintonizados pelos índices IAE e ISE ponderam o erro uniformemente, tendendo a reduzi-lo nos estágios iniciais da resposta. Seu uso é interessante apenas em situações que exigem baixo custo computacional.

Já os controladores sintonizados pelos índices ponderados pelo tempo (ITAE, ITSE, ISTSE) tendem a melhorar a *performance* do erro nos estágios intermediário e final da resposta da planta, quando uma parte significativa do tempo de simulação já é decorrida. Apesar de possuírem custo computacional maior, estes índices propõem significativa redução no tempo de acomodação. Isto é particularmente interessante em sistemas lentos, como é o caso do trocador de calor abordado neste artigo, cuja implementação dos algoritmos bioinspirado utiliza como função de custo o índice ISTSE.

VIII. SIMULAÇÃO COMPUTACIONAL

Nesta seção é apresentada uma série de experimentos que utilizam os diversos métodos descritos nas seções anteriores. Nos experimentos com os algoritmos bioinspirado considera-se 20 execuções das metaheurísticas no qual cada resultado é calculado sua média. A população em cada algoritmo foi 10 e o número máximo de iteração (ou geração) 60. Os códigos foram implementados em ambiente Matlab. Os testes foram realizados em um computador com processador Intel Core I7 e 6 GB de memória RAM.

Os controladores CPI(s) e F(s) especificados nas seções anteriores foram implementados no ambiente de simulação do SIMULINK®. Inicialmente foram realizados testes utilizando apenas a malha de controle retroalimentado (Fig. 7).

A Fig. 8 mostra as respostas da planta a uma excitação do tipo degrau unitário, em que ZN e SIMC representam a reação da planta quando seu controlador principal CPI(s) é sintonizado pelos métodos tradicionais. As curvas ZN-BA e SIMC-BA representam a resposta quando CPI(s) é inicialmente sintonizado pelos métodos tradicionais e em seguida otimizado pela metaheurística BA.

Neste experimento, incluiu-se na simulação o PSO, CS e FPA e realizou-se um comparativo. A Fig. 9 apresenta a simulação com as quatro metaheurísticas, o ZN e o SIMC. A curva do BA, PSO, CS e FPA representam a resposta quando

CPI(s) é inicialmente sintonizado pelo método SIMC e em seguida otimizado.

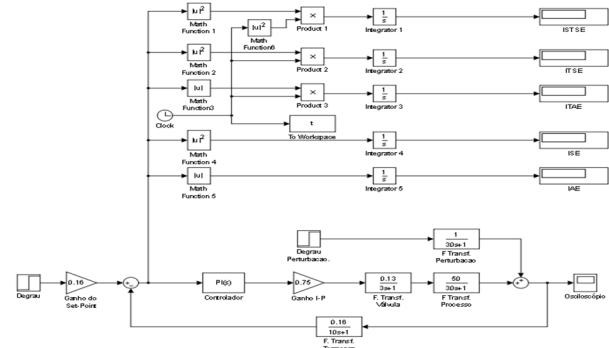


Figura 7. Sistema de Controle em Malha Retroalimentada.

A terceira etapa da simulação consiste em introduzir o controlador de alimentação avante F(s) na planta, cuja função é atenuar o distúrbio provocado pela vazão do fluido de entrada (Fig. 10).

No gráfico da Fig. 11 estão plotadas todas as oito curvas de resposta ao degrau, em que ZN-FF e SIMC-FF representam a resposta da planta na presença de F(s), quando seu controlador CPI(s) é sintonizado pelos métodos tradicionais. As curvas ZN-BA-FF e SIMC-BA-FF representam a resposta da planta quando CPI(s) é inicialmente sintonizado pelos métodos tradicionais e otimizado pela metaheurística BA, na presença de F(s).

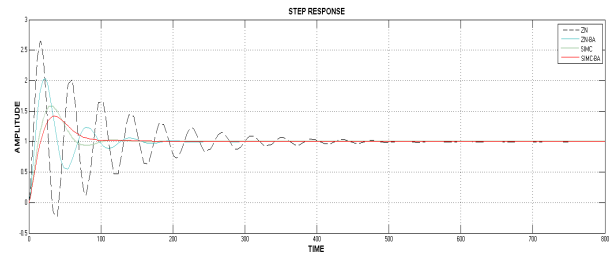


Figura 8. Respostas das Plantas ao Degrau com o BA.

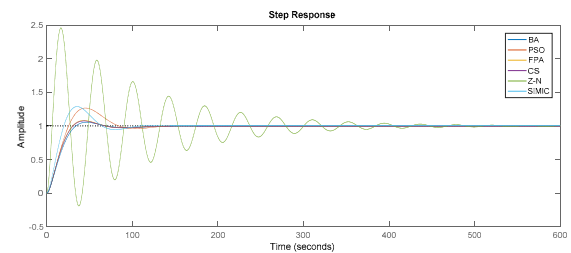


Figura 9. Respostas das Plantas ao Degrau com as quatro metaheurísticas.

O máximo sobressinal M_p representa o valor máximo do pico da curva de resposta. Seu valor é dado em porcentagem relativa ao valor especificado no *setpoint*. O tempo de acomodação T_s é representado para que a curva de resposta convirja a um valor próximo ao *setpoint*.

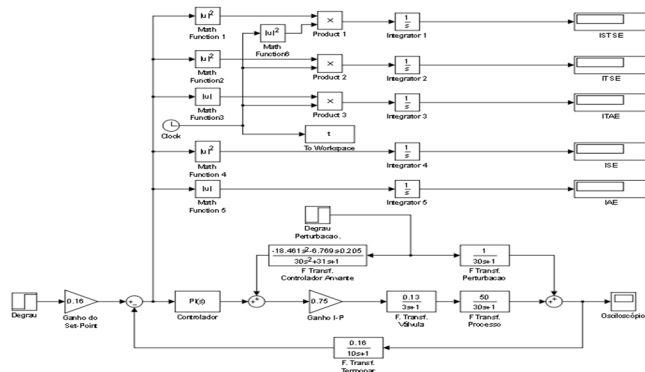


Figura 10. Sistema de Controle em Malha Retroalimentada com controlador de alimentação avante.

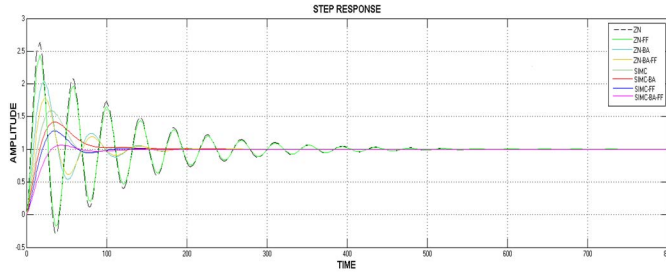


Figura 11. Respostas das Oito Plantas ao Degrau.

Neste trabalho utilizou-se a faixa de 2%. As características das respostas transitórias dos sistemas simulados foram organizadas conforme exposto na Tabela I.

TABELA I
CARACTERÍSTICAS DAS RESPOSTAS TRANSITÓRIAS

SINTONIA	Mp (%)	Ts (s)
ZN	163.0	419.0
ZN-FF	144.0	374.2
ZN-BA	103.0	145.2
ZN-BA-FF	78.0	121.5
SIMC	28.9	98.9
SIMC-BA	41.6	81.6
SIMC-FF	28.8	84.6
SIMC-BA-FF	5.4	101.0
SIMC-PSO-FF	26.6	124.0
SIMC-FPA-FF	8.0	105.0
SIMC-CS-FF	7.4	106.0
ZN-PSO-FF	42.2	177.0
ZN-FPA-FF	82.4	170.0
ZN-CS-FF	82.4	170.0

Observa-se que os sistemas sintonizados pelo método de Ziegler-Nichols (ZN, ZN-FF, ZN-BA, ZN-BA-FF, ZN-PSO-FF, ZN-FPA-FF e ZN-CS-FF) apresentaram os maiores valores de M_p e T_s . Em termos práticos estas curvas representam plantas de controle com estabilidade relativa menor e longos atrasos na acomodação. Dentre as 14 configurações estudadas o sistema SIMC-BA apresentou o melhor tempo de acomodação (81.6 segundos). Já o SIMC-BA-FF o melhor desempenho gerando um sobressinal de apenas 5.4%. A Tabela II propõe um comparativo entre os índices de desempenho IAE, ISE, ITAE, ITSE e ISTSE dos 17 controladores simulados.

TABELA II

ÍNDICES DE DESEMPENHO DOS SISTEMAS SIMULADOS

SINTONIA	IAE	ISE	ITAE	ITSE	ISTSE
ZN	11.68	0.80	1232	41.36	4380
ZN-FF	10.60	0.67	1105	33.38	3550
ZN-BA	5.54	0.43	232.3	9.57	410.7
ZN-CS	5.37	0.42	220.6	8.92	374.6
ZN-PSO	4.05	0.33	125.9	4.85	154.7
ZN-FPA	5.37	0.42	220.6	8.91	374.5
ZN-BA-FF	4.80	0.36	189.5	6.29	263.7
SIMC	4.73	0.40	147.3	7.40	281.0
SIMC-BA	5.06	0.39	198.4	6.92	298.7
SIMC-CS	5.05	0.39	197.4	6.88	295.9
SIMC-FPA	5.03	0.39	194.3	6.91	295.3
SIMC-PSO	5.60	0.46	214.6	10.36	475.4
SIMC-FF	3.88	0.37	91.51	3.92	89.78
SIMC-BA-FF	3.93	0.42	82.3	4.45	77.81
SIMC-CS-FF	3.91	0.42	82.2	4.41	76.94
SIMC-FPA-FF	3.89	0.41	80.2	4.36	75.23
SIMC-PSO-FF	4.68	0.43	136.5	5.69	168.7

Os sistemas SIMC-FF e ZN-PSO apresentaram os melhores valores de erro absoluto e erro quadrático (IAE, ISE), respectivamente. O sistema SIMC-FPA-FF apresentou o melhor desempenho no índice ITAE, o SIMC-FF o melhor no índice ITSE e o SIMC-FPA-FF no ISTSE. Um fator que contribuiu para este fenômeno foi a relativa rapidez com a qual a sua resposta convergiu ao degrau do *setpoint*. E finalmente, com relação ao tempo entre as metaheurísticas. O BA obteve o melhor tempo com 21.51 segundos usando o SMIC.

IX. CONCLUSÃO

Neste artigo foram estudados 17 tipos de configuração diferentes para o controlador PI do trocador de calor, sendo 4 deles sintonizados da forma tradicional e 13 otimizados pelo BA, PSO, CS e FPA. Os resultados das simulações demonstraram que no geral os controladores otimizados pelo BA apresentaram melhorias em sua eficiência, sendo o sistema otimizado com controladores SIMC-BA e de alimentação avante (SIMC-BA-FF) os que obtiveram os melhores resultados na análise de transientes da resposta M_p e T_s . Na comparação utilizando o índice de desempenhos IAE, o SIMC-FF, SIMC-BA-FF, SIMC-CS-FF e SIMC-FPA-FF obteve resultados próximos uns dos outros. No entanto, o SIMC-FF obteve melhor resultado. O ZN-PSO obteve melhor resultado no índice ISE. Já o FPA o melhor desempenho no índice ITA e ISTSE.

O trocador de calor apresentado neste trabalho foi modelado e simulado por *software* em condições ideais. Sabe-se que na prática existem fatores que interferem na qualidade da ação de controle, como ruídos de medição, perdas térmicas e agarramentos no atuador da válvula controladora, que pioram os transientes da resposta da planta. Os autores deste trabalho propõem como trabalho futuro a avaliação da *performance* das metaheurísticas na otimização de um trocador de calor em condições reais.

REFERÊNCIAS

- [1] SHAH, R. K., SEKULI, D. P. Classification of heat exchangers: Introduction, 1(1-2), 2007.

- [2] GOPAL, M. Basic Principles Of Feedback And Characteristics of Proportional-Integral-Derivative Modes of Control, 2 edn, 2002.
- [3] NITHYA, S. et al. Model Based Controller Design for Shell and Tube Heat Exchange-r. Sensors & Transducers Magazine (e-digest). Grimsby, ON, Canada, pág. 1677-1686. out. 2007.
- [4] YANG, Shin-she; GANDOMI, Amir Hossein. Bat algorithm: a novel approach for global engineering optimization. Engineering Computations: International Journal for Computer-Aided Engineering and Software. [s. L.], pág. 464-483. jul. 2012.
- [5] WANG, Gaige; GUO, Lihong. A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization. Journal Of Applied Mathematics. [s. L.], pág. 1-21. jan. 2012.
- [6] OMAR, Bendjehaba; SAIDA, Ishak Boushaki. Bat Algorithm for Optimal Tuning of PID Controller in an AVR System. International Conference on Control, Engineering & Information Technology (CEIT'14) Proceedings, pág. 158-170, 2014.
- [7] PUJANTARA, Margo; ABDILLAH, Muhammad. Adaptive PI Controller Berbasis Bat Algorithm (BA) dan Extreme Learning Machine (ELM) Untuk Kontrol Kecepatan Permanent Magnet Synchronous Motor (PMSM). JAVA Journal of Electrical and Electronics Engineering, v. 9, n. 2, 2011.
- [8] HEGAZY, Osman; SOLIMAN, Omar S.; SALAM, Mustafa Abdul. Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction, International Journal of Advanced Computer Research; Mar2015, Vol. 5 Issue 18, pág. 35, 2015.
- [9] DWI, Lastomo; MUHAMMAD, R. Djalal; WIDODO; IMAM, Robandi. Optimization of PID Controller Design for DC Motor Based on Flower Pollination Algorithm; USU Conference Systems, The 2015 International Conference on Electrical, Telecommunication and Computer Engineering; 2015.
- [10] YANG, X.-s. A New Metaheuristic Bat-Inspired Algorithm. In: YANG, X.-s. Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gon-zalez et al.): Studies in Computational Intelligence. Berlin: Springer, pág. 65-74, 2010.
- [11] ZHANGJUN, ZHANG, Kanyu. A Particle Swarm Optimization Approach for Optimal Design of PID Controller for Temperature Control in HVAC, 2011 Third International Conference on Measuring Technology and Mechatronics Automation (Volume:1), DOI: 10.1109/ICMTMA.2011.63, pág. 230-233, Publisher: IEEE, 2011.
- [12] MCKETTA JR, John J. (Ed.). Heat transfer Design Methods. CRC Press, 1991.
- [13] THOMAS, Charles. Process technology equipment and systems. Cengage Learning, 2014.
- [14] ZHANG, Jianming; WANG, Ning; WANG, Shuqing. A developed method of tuning PID controllers with fuzzy rules for integrating processes. In: American Control Conference, 2004. Proceedings of the 2004. IEEE, pág. 1109-1114, 2004.
- [15] CASTRUCI, P. B. L.; BITTAR, A.; SALES, R. M. Controle Automático. Rio de Janeiro: LTC, 2011.
- [16] KENNEDY, J.; EBERHART, R. Particle Swarm Optimization, In Proceedings of IEEE International Conference on Neural Networks, DOI:10.1109/ICNN.1995.488968, pág.1942-1948, 1995.
- [17] YANG, Xin-She, Flower pollination algorithm for global optimization: Unconventional Computation and Natural Computation 2012, Lecture Notes in Computer Science, Vol. 7445, pág. 240-249, 2012.
- [18] YANG, Shin-she; DEP, Suash. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Engineering with Computers, Volume 29, Issue 1, pág. 17-35, January, 2013.
- [19] KISHNAN, M.; PAREEK, S.; GUPTA, R. Optimal Tuning of PID controller by Cuckoo Search via Lévy flights, Advances in Engineering and Technology Research (ICAETR), 2014 International Conference on, DOI: 10.1109/ICAETR.2014.7012927, IEEE, pág. 1-5, 1-4 de agosto, 2014.
- [20] ÅSTRÖM, K. J.; HÄGGLUND, T. Revisiting the Ziegler-Nichols step response method for PID control. Journal Of Process Control, v. 14, n. 6, pág. 635-650, 2004.
- [21] ZIEGLER, J. G.; NICHOLS, N. B. Optimum Settings for Automatic Controllers. trans. ASME, v. 64, n. 11, 1942.
- [20] CHRISS, G.; SIGURD, S. Optimal PI-control and verification of the simc tuning rule, Proceedings IFAC Conference on Advances in PID Control, Italy, 2012.
- [23] SHINNERS, Stanley M. Modern control system theory and design. John Wiley & Sons, Cap. 5. pág. 290-292, 1998.



Nielsen Castelo Damasceno, é graduado em Engenharia da Computação pela Universidade Potiguar (2007). Especialista em Engenharia de Automação de processos à gás Natural pela Universidade Potiguar e CTGás (2009). Mestrado em Engenharia Elétrica na Universidade Federal do Rio Grande do Norte - UFRN (2010). Doutorado em Engenharia Elétrica na UFRN (2016). Atualmente é professor da Universidade Potiguar (UnP). Tem experiência na área de Engenharia Elétrica e Computação. Atuando principalmente nos seguintes temas: Algoritmos de otimização, metaheurísticas, microcontroladores e microprocessadores, Inteligência Computacional, Visão Computacional, Análise de Componentes Independentes, Clustering, Computação de alto desempenho e BigData.



Oscar Gabriel Filho é graduado em Engenharia Elétrica (1972) pela Universidade Federal de Juiz de Fora - UFJF, mestre (1997) e doutor (2004) em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte - UFRN. Foi chefe de obras da AEG-Telefunken do Brasil S.A. (1972-1977), chefe de manutenção, operações e gerente de dutos e terminais da Petróleo Brasileiro S.A.- PETROBRAS (1977-1994) e assistente técnico da Cia. Potiguar de Gás-POTIGÁS (1996-1998). Foi professor, pesquisador, criador e diretor de cursos de graduação e de pós-graduação na área tecnológica e das engenharias (1997-agosto de 2011), da Universidade Potiguar (UnP). Atualmente é pesquisador visitante (desde 01/12/2011) do Programa Petrobras de Formação de Recursos Humanos (PFRH), atuando no Projeto UFRN/Petrobras PRH-PB20, de Automação Industrial na Indústria de Petróleo e Gás. Possui facilidade de comunicação e relacionamento, capacidade de liderança, organização e estratégia corporativa voltadas para resultados. Possui interesse nas áreas de Controle de Processos, Automação Industrial, Inteligência Artificial e Petróleo e Gás Natural.