

# SINTONIA SIMULTÂNEA, PARALELA E COORDENADA DE PSSS UTILIZANDO OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

SERGIO C. MAZUCATO JR.<sup>1</sup>, BRUNO L. G. COSTA<sup>1</sup>, MARCELO F. CASTOLDI<sup>1</sup>, BRUNO A. ANGELICO<sup>1</sup>, DANILO S. SANCHES<sup>1</sup>, RODRIGO A. RAMOS<sup>2</sup>.

1. *Departamento de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná  
Cornélio Procópio, Paraná, Brasil.*

*E-mails: sergiomazucato@gmail.com, brunog\_costa@hotmail.com,  
marcastoldi@utfpr.edu.br, bangelico@utfpr.edu.br,  
danielosanches@utfpr.edu.br*

2. *Departamento de Engenharia Elétrica, Universidade de São Paulo  
São Carlos, São Paulo, Brasil.*

*E-mails: ramos@sc.usp.br*

**Abstract**— Power system controllers are typically designed using trial-and-error techniques, which may require large effort and time from designers to find a satisfactory solution. This work proposes an algorithm to perform a simultaneous and coordinated tuning of the controllers (PSSs) in an automatic form. The algorithm uses an optimization technique based on the ant colony metaheuristic. In addition, a parallel structure for the algorithm is proposed to minimize the computation time. Results show satisfactory performance of the tuned controllers, evidencing the effectiveness of the proposed technique. Furthermore, a significant productivity gain can be achieved if the engineer in charge of this design only supervises the automatic process, instead of performing all calculations by himself/herself.

**Keywords**— Angular Stability, Ant Colony Optimization, Automatic Tuning, Global Search, Power System Stabilizer.

**Resumo**— Controladores de sistemas de potência são tipicamente projetados usando técnicas de tentativa e erro, as quais podem exigir grande esforço e tempo de seus projetistas para encontrar uma solução satisfatória. Este trabalho propõe um algoritmo para realizar uma sintonia simultânea e coordenada de controladoras (PSSs) de forma automática. O algoritmo utiliza uma técnica de otimização baseada na meta-heurística colônia de formigas. Além disso, a estrutura paralela do algoritmo é proposta para minimizar o tempo computacional. Resultados apresentam desempenho satisfatório da sintonia de controladores, evidenciando a eficácia da técnica proposta. Desta maneira, um significativo ganho de produtividade pode ser atingido se o engenheiro responsável por esta tarefa apenas supervisionar o processo de sintonia automática, ao invés de efetuar todos os cálculos.

**Palavras-chave**— Estabilidade Angular, Otimização por Colônia de Formigas, Sintonia Automática, Busca Global, Estabilizadores de Sistemas de Potência.

## 1 Introdução

O sistema elétrico de potência está constantemente sujeitos a perturbações, as quais podem ser causadas por um curto-circuito na linha de transmissão, por um súbito incremento na carga, entre outras. Estas perturbações podem induzir oscilações eletromecânicas no sistema de potência, desde que a velocidade angular dos geradores oscile (Kundur, 1994). Com intuito de amortecer estas oscilações, controladores de sistemas de potência com os devidos parâmetros sintonizados devem ser inseridos em pontos estratégicos do sistema (Kundur, 1994).

Os modos de oscilação previamente mencionados são, tipicamente, local e inter área. Os modos locais representam a interação entre os geradores da mesma área, enquanto os modos inter área representam a interação entre um grupo de geradores de uma área específica, oscilando contra os geradores de outra área (Rogers, 2000). Controladores PSS (do inglês *Power System Stabilizer*) são, geralmente, usados para amortecer oscilações de modos locais. No

entanto, com uma boa sintonia de parâmetros, este controladores podem amortecer também oscilações de modos inter área. Na indústria, a estrutura mais utilizada para este tipo de controlador tem um bloco de ganho, um bloco de *washout* e de um ou dois blocos de compensação de fase.

Atualmente, a indústria usualmente utiliza o método de tentativa e erro para sintonizar PSSs. Esse método tenta encontrar uma solução viável apenas depois de um grande tempo, uma vez que o projetista responsável por esta tarefa tem de lidar com vários parâmetros ao mesmo tempo. Além disso, é comum em projetos que envolvam sintonia de PSSs utilizar um grande número de pontos de operação do sistema de potência. Este procedimento é adotado, a fim de verificar se os controladores sintonizados são eficazes para amortecer os modos de oscilação de todas as condições operacionais consideradas no projeto. Neste caso, o projetista tem que verificar se o ajuste oferece um amortecimento satisfatório para cada ponto de operação considerado. Se, pelo menos em uma condição de operação considerada o ajuste não fornecer um amortecimento satisfatório, o projeto deve ser feito novamente.

De modo a lidar com estas dificuldades, métodos de sintonia automática têm sido utilizados como alternativa aos métodos de tentativa e erro. Estes métodos proporcionam uma sintonia simultânea e coordenada de controladores, evitando interações adversas entre eles. Além disso, estes métodos consideram muitas condições de operação simultaneamente, assegurando a estabilidade e um amortecimento satisfatório. Inicialmente, métodos de otimização local foram empregados neste processo. No entanto, a necessidade de boas condições iniciais representa uma grande desvantagem (Werner, 2003; De Oliveira, 2009). Com o advento de técnicas de computação evolutiva, o processo de sintonia automática tem sido testado como métodos de otimização global, como, por exemplo, algoritmos genéticos (Do Bonfim, 2000; Chirra, 2009), otimização por enxame de partículas (Chatterjee, 2009), evolução diferencial (Castoldi, 2012), entre outros. No entanto, o processamento destes métodos é relativamente complexo e lento quando comparado com métodos de otimização local.

Este trabalho propõe o uso de um método relativamente novo de otimização global, baseado na teoria de colônia de formigas conhecida por ACO (do inglês, *Ant Colony Optimization*) para realizar o projeto de sintonia de controladores de sistemas de potência. Esta técnica tem sido testada, com resultados satisfatórios, em problemas envolvendo fluxo de carga (Vlachogiannis, 2005), despacho econômico de energia (Ciornei, 2012), problemas de distribuição de potência (Zhu, 2010), entre outros. Além disto, para reduzir o tempo computacional, o algoritmo foi implementado em uma estrutura paralela, uma vez que a maioria dos computadores atuais possui mais de um núcleo de processamento. Desta maneira, a intervenção humana do projetista é muito reduzida, no trabalho proposto, assim este encarregado pode dedicar seu tempo a outras tarefas e apenas verificar a viabilidade da solução após o processo de sintonia automática. Neste sentido, o ganho de produtividade para a indústria é evidente.

O presente trabalho está organizado da seguinte maneira: a Seção 2 descreve a formulação do problema de sintonia, a Seção 3 apresenta a teoria ACO. A Seção 4 mostra os resultados obtidos e a Seção 5 envolve as principais conclusões.

## 2 Formulação do Problema de Sintonia

O modelo do sistema de potência é usualmente baseado em um conjunto de equações diferenciais não lineares da forma:

$$\begin{aligned}\dot{\bar{x}}(t) &= f(\bar{x}(t), \bar{u}(t)) \\ \bar{y}(t) &= g(\bar{x}(t), \bar{u}(t))\end{aligned}\quad (1)$$

onde  $\bar{x}(t) = [\bar{x}_1(t), \dots, \bar{x}_n(t)]^T \in \mathbb{R}^{n \times 1}$  é o estado do sistema,  $u(t) = [u_1(t), \dots, u_p(t)]^T \in \mathbb{R}^{p \times 1}$  é o vetor de

entrada de controle e  $y(t) = [y_1(t), \dots, y_q(t)]^T \in \mathbb{R}^{q \times 1}$  é o vetor de medidas de saída.

Usualmente, a análise de estabilidade e as leis de controle podem ser representadas por modelos lineares. Este processo geralmente envolve a linearização do sistema de equações (1) em torno de um equilíbrio específico, que pode ser descrito no espaço de estados por

$$\begin{aligned}\dot{x}_z(t) &= A_z x_z(t) + B_z u_z(t) \\ y_z(t) &= C_z x_z(t)\end{aligned}\quad (2)$$

Em (2) as matrizes  $A_z \in \mathbb{R}^{n \times n}$ ,  $B_z \in \mathbb{R}^{n \times p}$  e  $C_z \in \mathbb{R}^{q \times n}$  são matrizes constantes. O vetor  $x_z(t) \in \mathbb{R}^{n \times 1}$  representa um desvio de um certo ponto de equilíbrio  $\bar{x}_{ez} \in \mathbb{R}^{n \times 1}$  e  $y_z(t) \in \mathbb{R}^{q \times 1}$  representa desvios de  $\bar{u}_{ez}$  e  $\bar{y}_{ez}$ , respectivamente.

Na indústria, há uma prática típica de selecionar um número de condições de operações, definindo diferentes pontos de equilíbrio, de maneira que estas condições representem todo o conjunto de operações que o sistema pode atingir. Desta maneira, o modelo (2) descreve toda a linearização do sistema, onde  $z = 1, \dots, L$  são os  $L$  pontos de operação considerados.

Considerando um controlador com realimentação dinâmica, o mesmo pode ser representado pelas seguintes equações:

$$\begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c \dot{y}_z(t) \\ u_z(t) = C_c x_c(t) \end{cases}\quad (3)$$

onde  $A_c \in \mathbb{R}^{n_c \times n_c}$ ,  $B_c \in \mathbb{R}^{n_c \times p}$  e  $C_c \in \mathbb{R}^{p \times n_c}$ . Posteriormente, rearranjando (2) e (3), o sistema em malha fechada pode ser escrito como

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} A_z & B_z C_c \\ B_c C_z A_z & A_c + B_c C_z B_z C_c \end{bmatrix} \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}\quad (4)$$

A equação (4) pode ser escrita de uma forma mais compacta

$$\dot{\tilde{x}}(t) = \tilde{A}_z \tilde{x}(t)\quad (5)$$

onde  $\tilde{x}(t) = \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}$  e  $\tilde{A}_z = \begin{bmatrix} A_z & B_z C_c \\ B_c C_z A_z & A_c + B_c C_z B_z C_c \end{bmatrix}$ .

Assim, de acordo com (4), alterando os parâmetros das variáveis de controle  $A_c$ ,  $B_c$  e  $C_c$ , que são matrizes de controladores (PSSs), torna-se possível controlar os autovalores do sistema em malha fechada e, além disso, o amortecimento do sistema. A Figura 1 mostra a estrutura de PSS mais comum.

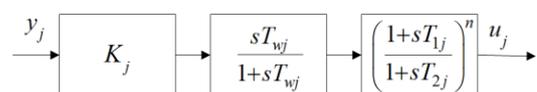


Figura 1. Diagrama de blocos PSS.

O subscrito  $j$  em todos os parâmetros dos controladores da Figura 1 denota que este controlador é adicionado ao  $j$ -th AVR. O Bloco intermediário na Figura 1 é conhecido como filtro de *washout* e garante que o ganho do controlador é zero sobre condições de regime permanente. Este bloco tem uma ação derivativa representada pelo termo  $sT_{wj}$ , a qual pode ser incorporada na planta para simplificar a formulação do projeto. Percebe-se que isto não implica que um sinal derivativo ideal necessite ser medido nesta planta, uma vez que este termo pode ser combinado com um dos polos do controlador depois de finalizado o projeto, e o resultado pode ser implementado como um bloco de *washout* típico (De Oliveira, 2009).

Considerando um controlador com dois blocos de avanço-atraso, as matrizes no espaço de estados assumem o seguinte formato:

$$A_{Cj} = \begin{bmatrix} -\alpha_j & 0 & 0 \\ \gamma_j - \alpha_j \beta_j & -\gamma_j & 0 \\ \beta_j \gamma_j - \alpha_j \beta_j^2 & \gamma_j - \beta_j \gamma_j & -\gamma_j \end{bmatrix}; \quad B_{Cj} = \begin{bmatrix} 1 \\ \beta_j \\ \beta_j^2 \end{bmatrix};$$

$$C_{Cj} = \begin{bmatrix} 0 & 0 & K_j \end{bmatrix} \quad (6)$$

onde  $\alpha_j = 1/T_{wj}$ ,  $\beta_j = T_{1j}/T_{2j}$ ,  $\gamma_j = 1/T_{2j}$ .

A partir de (4), fica evidente que o problema de sintonia de parâmetros de controladores consiste em encontrar uma matriz tripla ( $A_C, B_C, C_C$ ) que garanta que as matrizes  $z=1, \dots, L$ , cumpram o critério de desempenho desejado. O critério mais aceito pela indústria, e utilizado neste trabalho, afirma que o desempenho do sistema não linear de malha fechada pode ser considerado satisfatório se todos os autovalores de todas as matrizes  $\tilde{A}_z$ ,  $z=1, \dots, L$ , apresentarem uma razão de amortecimento maior que um dado valor mínimo predeterminado  $\xi_{\min}$ . Neste trabalho foi realizada uma sintonia sequencial e uma paralela baseada no algoritmo ACO para a sintonia dos parâmetros dos controladores, com objetivo de fornecer um amortecimento satisfatório para todos os autovalores das matrizes  $\tilde{A}_z$ ,  $z=1, \dots, L$ .

### 3 Otimização por Colônia de Formigas

A meta-heurística ACO foi desenvolvida para solucionar problemas de otimização baseando-se no comportamento de formigas (Dorigo, 2004). Os indivíduos são introduzidos em uma sociedade altamente estruturada, a qual desempenha tarefas muitas vezes bastante complexas, como por exemplo a busca por alimento.

Durante a busca por alimento, as formigas liberam uma substância chamada feromônio, que ajuda a encontrar o menor trajeto entre seu ninho e a fonte de alimentação (Dorigo, 2004). O ACO partiu desta

ideia e foi originalmente utilizado para solucionar problemas de otimização combinatória.

O algoritmo implementado neste trabalho é uma versão contínua do ACO, baseado em Socha (2008), onde a memória representada pelo depósito de feromônio é substituída por um arquivo de soluções.

Inicialmente, um arquivo de  $T$  soluções é criado, desta maneira a matriz apresenta  $k$  soluções candidatas, geradas aleatoriamente para cada uma das  $n$  variáveis do problema, com um intervalo predeterminado (restrições para cada variável). Neste trabalho, um arquivo padrão com 25 soluções ( $k=25$ ) foi utilizado, sendo 30 o número de variáveis para cada solução candidata ( $n=30$ ), devido ao número de controladores, ( $nc=10$ ), com três parâmetros cada. Estes parâmetros são  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $K$ . No entanto, o parâmetro  $\alpha$  não foi considerado no arquivo de soluções por ser relacionado a constante de *washout*, a qual é definida inicialmente pelo projetista (com intervalo entre 3s e 20s). A Figura 2 ilustra a configuração do arquivo de soluções.

$T$	1	2	3	4	5	6	...	$n-2$	$n-1$	$n$
$S_1$	$K_1^1$	$\beta_1^1$	$\gamma_1^1$	$K_2^1$	$\beta_2^1$	$\gamma_2^1$	...	$K_{nc}^1$	$\beta_{nc}^1$	$\gamma_{nc}^1$
$S_l$	$K_1^l$	$\beta_1^l$	$\gamma_1^l$	$K_2^l$	$\beta_2^l$	$\gamma_2^l$	...	$K_{nc}^l$	$\beta_{nc}^l$	$\gamma_{nc}^l$
$S_k$	$K_1^k$	$\beta_1^k$	$\gamma_1^k$	$K_2^k$	$\beta_2^k$	$\gamma_2^k$	...	$K_{nc}^k$	$\beta_{nc}^k$	$\gamma_{nc}^k$

Figura 2. Arquivo de soluções.

O arquivo de soluções é ordenado de acordo com a qualidade de cada solução e avaliado através da função custo (*fitness*), relacionada com o amortecimento mínimo entre todos os pontos de operação, dado por

$$f(K, \beta, \gamma) = -\min(\min(\text{damp}(\tilde{A}_z))) \quad (7)$$

para  $z=1, \dots, L$ . É importante reparar que a minimização em (7) corresponde a maximizar o amortecimento do modo menos amortecido (essa é a razão do sinal negativo na função objetivo). O  $\min$  interior encontra o menor amortecimento de cada autovalor da matriz em malha fechada  $\tilde{A}_z$  e o  $\min$  exterior determina o menor amortecimento entre todos os mínimos encontrados pelo primeiro  $\min$ .

Em seguida, para cada vetor do arquivo de soluções ( $l=1, \dots, k$ ) um peso  $\omega_l$  é calculado de acordo com a função Gaussiana com média  $u=1$  e desvio padrão  $\sigma=qk$ , como:

$$\omega_l = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(l-u)^2}{2\sigma^2}} = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \quad (8)$$

onde  $q$  é um parâmetro do algoritmo. Se  $q$  é pequeno, as melhores soluções tem maior probabilidade de serem escolhidas. Por outro lado, se  $q$  é elevado, a

escolha é mais uniforme [13]. Uma colônia de 20 formigas ( $M = 20$ ) foi adotada, o que significa que 20 dos 25 elementos são escolhidos para o arquivo de soluções para atualizar a posição. O método da roleta foi adotado para escolher as 20 formigas, baseado no vetor de probabilidade cumulativa dado por

$$p_l = \frac{\omega_l}{\sum_{r=1}^k \omega_r} \quad (9)$$

O próximo passo consiste em encontrar o desvio padrão  $\sigma^i$  correspondente para cada elemento do vetor de soluções  $n$ -dimensional. Assumindo que a posição  $S_l^i$  (com  $l=1, \dots, k$ ,  $i=1, \dots, n$ ) do vetor de soluções  $S_l$ , o desvio padrão deste elemento pode ser calculado como:

$$\sigma_l^i = \frac{\zeta}{k-1} \sum_{j=1, j \neq l}^k |S_j^i - S_l^i| \quad (10)$$

onde  $\zeta$  é a taxa de evaporação do feromônio (parâmetro de entrada). Em consequência, uma variável aleatória com média zero e desvio padrão  $\sigma_l^i$  é adicionada a solução  $S_l^i$ , e então novas direções ( $S_{\text{mod}}$ ) são geradas para cada uma das formigas.

Com objetivo de evitar que  $S_{\text{mod}}$  ultrapasse as fronteiras predefinidas para o problema, as seguintes restrições são impostas:

$$\begin{aligned} S_{\text{mod}} \geq S_{\text{max}} &\Rightarrow S_{\text{mod}} = S_{\text{max}} \\ S_{\text{mod}} \leq S_{\text{min}} &\Rightarrow S_{\text{mod}} = S_{\text{min}} \end{aligned} \quad (11)$$

O algoritmo realiza a busca até que o modo menos amortecido alcance um valor superior ou igual a 5%.

Como mencionado anteriormente, algumas restrições são aplicadas a cada parâmetro do algoritmo, visando sua implementação prática. A variação de 3–20 s para a constante de tempo de *washout* é dado por (12).

$$\alpha_j = \alpha_{j\_const}, \quad 0,05 \leq \alpha_{j\_const} \leq 0,33 \quad (12)$$

Os parâmetros  $\beta_j$ ,  $j=1, 2, \dots, nc$ , são relacionados a compensação de fase  $\phi_j$  que cada controlador fornece ao respectivo sinal de entrada, dado por:

$$\beta_j = \frac{T_{1j}}{T_{2j}} = \frac{1 + \sin(\phi_j)}{1 - \sin(\phi_j)} \quad (13)$$

A máxima compensação de fase definida pela indústria é de  $70^\circ$ . Assim, o intervalo de variação de  $\beta_j$  é

$$0 \leq \beta_j \leq 32 \quad (14)$$

O parâmetro  $\gamma_j$  é calculado de acordo com (15):

$$\gamma_j = \omega \sqrt{\beta_j} \quad (15)$$

onde  $\omega$  é a frequência de oscilação do modo de interesse  $\lambda$ . O objetivo principal é amortecer ambos os modos locais e inter-área. Os modos locais possuem,

tipicamente, uma frequência de oscilação  $\omega \leq 9$ . Desta maneira o intervalo de  $\gamma_j$  pode ser definido como:

$$0 \leq \gamma_j \leq 50 \quad (16)$$

Finalmente, o ganho estático  $K_j$  do controlador é baseado em valores práticos adotado pela indústria, definido como:

$$0 \leq K_j \leq 20 \quad (17)$$

enquanto que para maiores ganhos, saturações na saída do controlador podem ser induzidas mesmo com pequenas perturbações.

#### 4 Aplicação do Algoritmo Proposto

A fim de demonstrar a eficiência do algoritmo proposto, algumas simulações foram feitas sobre o sistema interligado *benchmark New England/New York* mostrado na Figura 3. Todos os geradores foram descritos por um modelo de sexta ordem (Pal, 2005), e equipados com um AVR estático de modelo de primeira ordem, com ganhos de 50 p.u./p.u. e constante de tempo de 0,01s. O sistema de transmissão foi projetado como um circuito passivo e as cargas do sistema como impedâncias constantes.

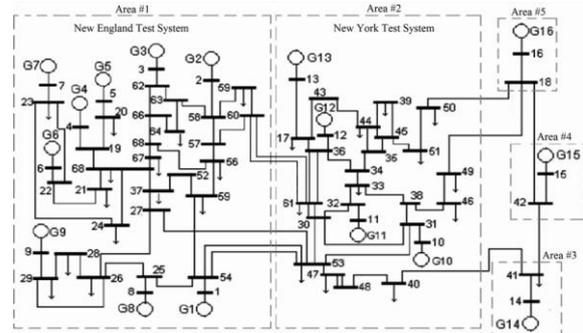


Figura 3. Sistema teste *New England/New York*.

O sistema sem nenhum controlador é altamente instável e, então, é necessário inserir PSSs para controlar seus modos de oscilação. Um PSS foi inserido em cada gerador visando fornecer um amortecimento satisfatório para os modos locais e inter-área presentes no sistema.

Os testes neste trabalho consideraram variações de aumento de 30% na carga total do sistema e redução de 25% da carga total do sistema em comparação com o caso base (Pal, 2005), que também foi considerado. Outras 21 condições de operação foram distribuídas no intervalo de carga do sistema, criando um conjunto de 24 condições de operação.

Após uma cautelosa análise de controlabilidade/observabilidade (via resíduos e fatores de participação), 10 PSSs foram inseridos no sistema de potência. Os PSSs foram inseridos nos geradores G2, G3, G6, G7, G8, G9, G10, G11, G12 e G16 visando fornecer um amortecimento satisfatório para os modos de oscilação presentes no sistema. Este amortecimento foi possível realizando uma sintonia simultânea

nea e coordenada dos PSSs, utilizando o algoritmo proposto.

Ao analisar o algoritmo sequencial aplicado ao problema foi possível perceber que há apenas uma operação (verificação do amortecimento) com alto tempo de execução, quando comparada com outras operações (utilizando um microcomputador com processador dual core 2,0 GHz e 3 GB DDR RAM).

O custo computacional foi significativamente alto, e reduzido a praticamente metade quando paralelizado. Dessa maneira, a única operação paralelizada no algoritmo proposto foi o cálculo do amortecimento. A Tabela 1 apresenta o custo de tempo de cada operação do programa.

Tabela 1. Custo de tempo para cada operação.

Operação	Tempo	
	Não-Paralelo	Paralelo
Ponderação da Solução	150 $\mu$ s	170 $\mu$ s
Seleção da Formiga	16ms	16ms
Atualização da Posição	5ms	5ms
Cálculo do Custo	700ms	400ms

A principal diferença entre os algoritmos é a função *fitness* (cálculo do amortecimento), que foi realizada de forma paralela. Em outras palavras, cada núcleo do processador realiza a tarefa para uma solução diferente, ao mesmo tempo, enquanto no algoritmo sequencial todos os núcleos realizam a tarefa para a mesma solução.

Para paralelizar o código, a ferramenta de paralelização *parfor toolbox* do software Matlab<sup>®</sup> foi utilizada. Esta ferramenta permite que o programa execute algumas partes do código de forma independente (unidades de trabalho), isto é, múltiplas tarefas concomitantemente. A ferramenta constrói um laço que distribui automaticamente as tarefas aos múltiplos espaços de trabalho do Matlab<sup>®</sup>.

O conceito básico de um laço paralelo (*parfor-loop*) no Matlab<sup>®</sup> é o mesmo que o de um laço *for* padrão: o Matlab<sup>®</sup> executa uma série de sentenças (corpo do laço) em um intervalo de valores. Parte do corpo do laço paralelo é executado no cliente Matlab<sup>®</sup> (onde o laço paralelo é executado) e parte é executado em outros espaços de trabalho do Matlab<sup>®</sup>. Os dados necessários sobre o qual opera o laço paralelo são enviados do cliente para outros espaços de trabalho, onde a maior parte do cálculo computacional ocorre, e os resultados são enviados de volta ao cliente, onde são agrupados.

Para comparar o tempo de execução dos algoritmos, foram realizadas execuções sequenciais e paralelas. Foi considerado uma população inicial com 20 formigas e parâmetros de 0,01 para  $q$  e 0,025 para  $\zeta$ . Estes parâmetros foram selecionados após um grande número de testes com o algoritmo sequencial.

O melhor membro da população inicial (o mesmo para ambos os algoritmos) forneceu um amorte-

cimento de -13.07% para o ponto menos amortecido. Pode-se observar que o melhor membro da população inicial é uma solução absolutamente não factível, e muito distante do objetivo (5%), o que valida ainda mais o método. Os resultados da simulação podem ser vistos na Tabela 2, a qual também apresenta o ganho de tempo com o algoritmo paralelo, conforme o esperado. É importante ressaltar que a única melhoria no algoritmo paralelo é o tempo de execução, ou seja, ambas as soluções apresentaram a mesma qualidade. Como o ACO é um método meta-heurístico as soluções não são iguais porém apresentam o mesmo grau de factibilidade.

Com objetivo de verificar a validade das soluções encontradas, uma simulação não linear foi feita no sistema. Um ponto não considerado na etapa do projeto foi usado para demonstrar a robustez da sintonia de controladores. Neste caso, foram removidas as linhas #57 e #60 do caso base. Para gerar uma resposta não linear foi aplicado um curto-circuito no instante de tempo 2 s, durante 16 ms, na linha entre as barras #47 e #48. A frequência do gerador 1 é mostrada na Figura 4. O resultado da simulação não linear mostra que os controladores sintonizados pelo algoritmo proposto são eficientes para amortecer oscilações eletromecânicas causadas por pequenas perturbações. Além disso, o algoritmo proposto fornece resultados rapidamente, o que torna possível sua aplicação em sistemas maiores.

Tabela 2. Parâmetros dos controladores sintonizados.

Controlador	Não-paralelo – 21 it – 377,66 seg			Paralelo – 21 it – 184,87seg		
	Ganho	T1	T2	Ganho	T1	T2
PSS 1 (2)	20,00	0,47	0,07	14,20	0,14	0,06
PSS 2 (3)	12,91	0,19	0,09	20,00	0,45	0,12
PSS 3 (6)	8,67	0,28	0,11	4,74	0,89	0,16
PSS 4 (7)	13,20	0,62	0,11	12,04	0,18	0,06
PSS 5 (8)	20,00	0,47	0,07	5,41	0,46	0,07
PSS 6 (9)	13,67	0,75	0,11	10,27	0,31	0,06
PSS 7 (10)	19,47	0,22	0,14	19,92	0,33	0,11
PSS 8 (11)	9,52	0,14	0,05	11,09	0,29	0,06
PSS 9 (12)	8,91	0,54	0,07	13,93	0,45	0,06
PSS 10 (16)	4,91	0,22	0,05	3,74	0,31	0,06
Amort. (%)	5,02			5,06		

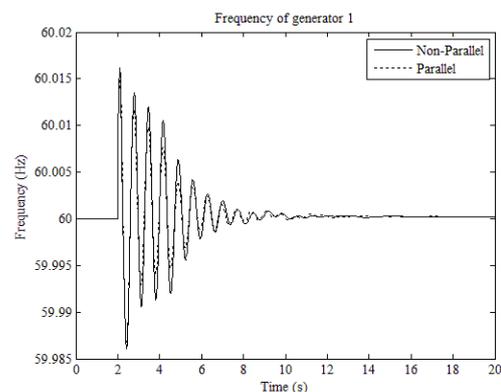


Figura 4. Simulação não linear do sistema de potência.

## 5 Conclusões

Foi apresentado um procedimento automático para auxiliar projetistas a realizar a sintonia de PSSs de maneira eficiente. O método utiliza uma técnica relativamente nova conhecida por Otimização por Colônia de Formigas com uma estrutura paralelizada e sintoniza de forma coordenada e simultânea os controladores do sistema. Os resultados mostram grande robustez do algoritmo proposto, alcançando uma sintonia satisfatória de PSSs considerando muitas condições de operações ao mesmo tempo. Outro aspecto importante foi o uso do algoritmo paralelizado, o qual garantiu um tempo inferior de execução quando comparado com a implementação convencional, sem o paralelismo. Além disso, o engenheiro encarregado da tarefa de sintonia dos PSSs pode focar seu trabalho em outras tarefas importantes e apenas supervisionar o projeto, enquanto o computador executa os cálculos envolvidos no processo de sintonia.

## Agradecimentos

Os autores agradecem o suporte da CAPES, CNPq, Fundação Araucária e a FAPESP para esta pesquisa, sob o número de processo 2011/01035-7.

## Referências Bibliográficas

- Castoldi, M. F.; Mazucato Jr, S. C.; Rodrigues, C. R. and Ramos, R. A (2012). Simultaneous and coordinated tuning of PSSs and PODs using differential evolution. Proceedings of IEEE Power and Energy Society General Meeting, pp. 1-8.
- Chatterjee, A.; Ghoshal, S. P. and Mukherjee, V (2011). Chaotic ant swarm optimization for fuzzy-based tuning of power system stabilizer. International Journal of Electrical Power & Energy Systems, vol. 33, pp. 657-672.
- Chirra, A. and Patil, D. R (2009). Automatic Tuning of PSS Through FACTS by Using Genetic Algorithm. in proceedings of International Conference on Control, Automation, Communication and Energy Conservation , pp. 1-6.
- Ciornei, I. and Kyriakides, E (2012). A GA-API Solution for the Economic Dispatch of Generation in Power System Operation. IEEE Transactions on Power Systems, vol. 27, pp. 233-242.
- De Oliveira, R. V.; Kuiava R.; Ramos, R. A. and Bretas, N. G (2009). Automatic Tuning Method for the Design of Supplementary Damping Controllers for Flexible Alternating Current Transmission System Devices. IET – Generation, Transmission and Distribution, vol. 3, pp. 919-929.
- Do Bomfim, A. L. B.; Taranto, G. N. and Falcao, D. M (2000). Simultaneous Tuning of Power System Damping Controllers Using Genetic Algorithms. IEEE Transactions on Power Systems, vol. 15, pp. 163-169.
- Dorigo, M. and Stützle, T (2004). Ant Colony Optimization. Scituate, USA: Bradford Company.
- Kundur, P. Power System Stability and Control. New York: McGraw-Hill.
- Pal, B. and Chaudhuri, B (2005). Robust Control in Power Systems. Springer Science+Business Media.
- Rogers, G. Power System Oscillations. Kluwer Academic Publishers, 2000.
- Socha, K. and Dorigo, M (2008). Ant Colony Optimization for Continuous Domains. European Journal of Operational Research, vol. 185, pp. 1155 - 1173.
- Vlachogiannis, J. G.; Hatziargyriou, N. D. and Lee, K. Y. Ant Colony System-Based Algorithm for Constrained Load Flow Problem. IEEE Transactions on Power Systems, vol. 20, pp. 1241-1249, 2005.
- Werner, H.; Korba, P. and Yang, T. C (2003). Robust Tuning of Power System Stabilizers Using LMI Techniques. IEEE Transactions on Control Systems Technology, vol. 11, pp. 147-152.
- Zhu, J.-m. and Ren, H.-e (2010). Improved ant colony algorithm for Power Distribution Network Planning. Proceedings of 2nd IEEE International Conference on Advanced Computer Control, pp. 357-360.