# OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS APLICADA AO PROJETO DE CONTROLE MULTIVARIÁVEL

PHELIPE S. OLIVEIRA[1], MISSILENE DA S. FARIAS[2], LUCIANO S. BARROS[1], LUIZ G. DE Q. SILVEIRA JUNIOR[1].

1. *Universidade Federal do Rio Grande do Norte*
*Centro de Tecnologia*
*Campus Universitário Lagoa Nova*
*Natal, Rio Grande do Norte, Brasil. CEP 59078-970*
2. *Universidade Federal do Semiárido*
*Programa de Pós-graduação em Sistemas de Comunicação e Automação*
*Mossoró, Rio Grande do Norte, Brasil. CEP 59.625-900*
*E-mails:* phsena2000@gmail.com, missilene@ufersa.edu.br,lsalesbarros@dee.ufrn.br,junior@ct.ufrn.br

**Abstract—** Control design to linear MIMO systems requires the state space representation, where the most used techniques are the LQR and the LQG, both based on states feedback. In despite of the good results obtained from these methods, the control design is not a straightforward task due to the trial and error method involved in the definition of weight matrices. In such cases, may be hard tuning the controller parameters in order to obtain the optimal behavior of the system. In this work, it proposes a technique in which there are no trial and error processes involved and the control design is carried out to fulfill specifications, for maximum overshoot and settling time, previously specified. The proposed technique is based on the modification of the states transition matrix and uses an algorithm based on particle swarm optimization, once the goal in control system design methods is the same employed by minimizing a cost function which models the system performance on combinatorial optimization problem. Performance evaluation and comparison with the usual technique based on genetic algorithms are performed via computer simulation. The obtained results show that is possible to design controllers which fulfill design specifications and this new approach may lead to new alternatives for exploiting the potential of particle swarm optimization on states feedback control design.

**Keywords—** particle swarm optimization, multivariable control, state feedback, evolutionary computation, design of controllers.

**Resumo—** Neste artigo é proposto um algoritmo de otimização por enxame de partículas que aplicado ao projeto de controladores de sistemas multivariáveis busca parâmetros do controlador que garantam uma resposta dinâmica respeitando condições de sobressinal e tempo de acomodação, especificadas pelo projetista do controle. O objetivo é verificar se esta técnica de computação evolutiva apresenta bons resultados com uma baixa complexidade temporal para que posteriormente seja aplicada ao controle adaptativo, mais especificamente, estimar parâmetros online no escalonamento de ganhos. O desempenho do algoritmo de otimização por enxame de partícula é comparado com um Algoritmo Genético que foi ajustado previamente para o projeto de controle de sistemas multivariáveis.

**Palavras-chave—** otimização por enxame de partículas, controle multivariável, realimentação de estados, computação evolutiva, projeto de controladores.

## 1 Introdução

Enquanto a teoria de controle clássico propicia condições para estabilidade, a teoria de controle moderno além de garantir a estabilidade proporciona técnicas que otimizam os critérios de desempenho (Ogata, 2010). Além disso, o controle moderno trata de sistemas multivariáveis e trás uma simplificação ao projeto de sistemas de controle em razão de se basear no modelo de um sistema de controle real (Dorf and Bishop, 2009). Entretanto, o modelo da planta é uma aproximação do sistema real e provavelmente ignora alguns detalhes de operação, que quando somados a possíveis perturbações com a inexatidão em sensores e atuadores pode levar a um sistema instável.

Um controle robusto poderia ser proposto para tratar com tais sistemas. Apesar de o controle robusto garantir estabilidade e manter um nível esperado nos critérios de desempenho diante de variações na dinâmica da planta e incertezas no sistema de controle, a amplitude das variações pode ser tão grande que as especificações de desempenho dificilmente serão atendidas com um ganho de realimentação linear e com coeficientes constantes.

Para manter um nível esperado no desempenho em se tratando de sistemas multivariáveis, é necessária a aplicação de um controle adaptativo. Portanto, busca-se uma ferramenta que possibilite uma adaptação online nos parâmetros do controlador. A princípio, o intuito é utilizar o escalonamento de ganhos e introduzir uma ferramenta para possibilitar o ajuste online dos parâmetros do controlador.

Em busca da ferramenta, foi desenvolvido em trabalhos anteriores um algoritmo genético (AG) aplicado ao projeto de controle multivariável e testado em alguns sistemas (Oliveira et al. 2010), (Barros et al. 2012). Inclusive, em Silveira Júnior et al. (2012) o algoritmo teve alguns de seus parâmetros, tais como, tamanho de população, tipos de seleção de indivíduos para reprodução e faixas de possíveis valores para os indivíduos da população ajustados por uma campanha exaustiva de testes e simulações com controles multivariáveis.

Mesmo com o ajuste dos parâmetros do algoritmo genético que propiciaram boas soluções do ponto

de vista do projeto, a complexidade temporal não possibilitou a utilização da técnica para aplicação na estimação de parâmetros de forma online.

Neste artigo, o intuito é encontrar uma nova alternativa que forneça boas soluções com menor complexidade temporal em relação ao AG. Como a computação evolutiva cresce demasiadamente e apresenta métodos capazes de encontrar soluções próximas do ótimo global, propomos a utilização da Otimização por Enxame de Partículas (PSO, para Particle Swarm Optimization) que foi introduzida por *Kennedy* e *Eberhart* (1995). A escolha é devido a característica de cooperação existente entre as partículas do enxame, o que diferencia da luta pela sobrevivência adotada no AG. Isto pode trazer benefícios se esta característica resultar em menor complexidade temporal para encontrar boas soluções.

O algoritmo PSO é uma técnica de otimização baseada no comportamento de um enxame inteligente, que terá o objetivo de buscar parâmetros do controlador para obter um comportamento dinâmico desejado em função dos requisitos de máximo sobressinal e tempo de acomodação, duas medidas previamente definida pelo projetista do sistema de controle. A utilização desta técnica vem crescendo na resolução de problemas de otimização (Parsopoulos and Vrahatis, 2004), (Janson and Middendorf, 2005), (Min et al., 2011).

O presente artigo está organizado em 5 seções. A Seção 2 descreve conceitos relacionados ao projeto de controladores multivariáveis baseados na definição das saídas do sistema. A Seção 3 aborda o problema da escolha de parâmetros do controlador por meio da Otimização por Enxame de Partículas e detalha o funcionamento do algoritmo PSO. A Seção 4 explana os resultados obtidos a partir de uma comparação do desempenho do PSO com o Algoritmo Genético. A última seção do artigo, a Seção 5, menciona as conclusões obtidas perante os resultados.

## 2 Projeto de Controle via Realimentação de Estados

A saída do sistema representado no espaço de estados é obtida a partir da Equação abaixo:

$$y(t) = (C - DK)e^{(A-BK)t}\big(BR(t) + x(0)\big) + DR(t) \quad (1)$$

Onde $A, B, C$ e $D$ são as matrizes de estado, de entrada, de saída e de transmissão direta, respectivamente. A matriz $R$ simboliza a entrada de referência, $x(0)$ representa as condições iniciais e $K$ é a matriz de realimentação de estados. $x(t)$ é o estado do sistema dinâmico e $y(t)$ é a equação de saída que pode relacionar a saída com as variáveis de estado e os sinais de entrada.

Pode-se observar que a resposta temporal depende dos autovalores da matriz $A - BK$. O objetivo do algoritmo PSO é encontrar uma matriz $K$ que garanta uma saída do sistema de acordo com os índices de desempenho, máximo sobressinal e tempo de acomodação, estabelecidos pelo projetista.

## 3 Projeto de Controlador Multivariável por Otimização de Enxame de Partículas

A Otimização por Enxame de Partículas é uma técnica evolucionária baseada na simulação de um comportamento social. Foi introduzida por *Kennedy* e *Eberhart* (1995) com base em análises de como bandos de aves ou cardumes de peixes conseguiam interagir para solucionar um problema de forma distribuída. Após as observações da natureza, criou-se um algoritmo que consiste em um enxame de partículas espalhadas em um espaço multidimensional movimentando-se com base em suas próprias informações, como também a partir de informações trocadas com outras partículas. Ao invés da competição pela sobrevivência, existe cooperação entre as partículas.

Assim como em outros algoritmos evolutivos, o algoritmo PSO possui uma população de indivíduos, e cada indivíduo, no caso do PSO chamado de partícula, armazena sua posição, que é uma solução em potencial inicializada de forma aleatória. A diferença é que o PSO não utiliza métodos para extrair uma nova população de outra população existente. Mas, de geração em geração suas partículas evoluem a partir de três parâmetros: fator de sociabilidade, que direciona a partícula para a melhor posição descoberta pelo enxame; fator de individualidade, uma atração da partícula pela melhor posição que já visitou e por fim, a velocidade, que indica a variação da posição de cada partícula.

A Figura 1 apresenta o fluxograma do algoritmo PSO proposto neste trabalho. A seguir, cada bloco da figura será apresentado com um nível maior de detalhes.

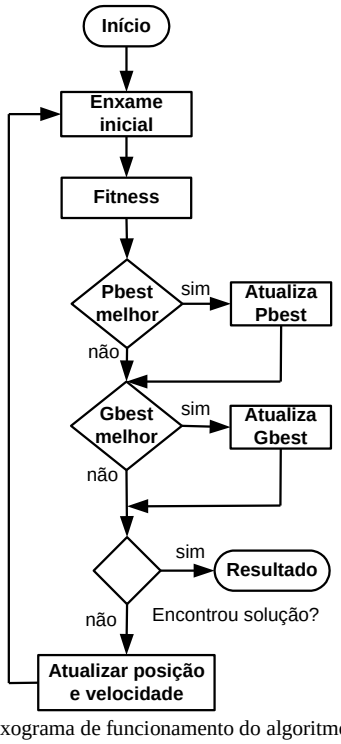



Figura 1. Fluxograma de funcionamento do algoritmo PSO.

Cada partícula no PSO é associada a dois vetores: posição e velocidade. Contando ainda com acesso aos vetores que constituem a melhor posição já visitada pelo enxame e a melhor posição já visitada pela própria partícula. A primeira etapa, enxame inicial, do algoritmo PSO, consiste em inicializar os vetores posição e velocidade de forma aleatória em todas as partículas do enxame. Visto que o vetor posição, $x_i(\mathrm{t}) = \{x_{\mathrm{i1}}(\mathrm{t}), \dots, x_{\mathrm{in}}(\mathrm{t})\}$, representa uma matriz de ganho, K, a inicialização dos seus elementos apesar de ser aleatória respeita uma faixa de valores que os elementos da matriz K podem assumir. Isto porque em se tratando de projeto de controle, fica impraticável implementar um controlador que demande de muito gasto energético devido a valores elevados na matriz de ganho. Já os elementos do vetor velocidade, $v_i(\mathrm{t}) = \{v_{\mathrm{i1}}(\mathrm{t}), \dots, v_{\mathrm{in}}(\mathrm{t})\}$, podem ser limitados para assegurar que as partículas permaneçam dentro do espaço de busca.

A próxima etapa consiste em utilizar uma função de *fitness*, também chamada de função custo para avaliar cada partícula do enxame. A Equação de saída do sistema mostrada em (1), na Seção anterior, é considerada a função de *fitness* do algoritmo PSO. Com o conhecimento das matrizes A, B, C e D que representam o sistema no espaço de estados e com a substituição da posição de uma partícula (matriz K) na Equação (1), é possível obter o comportamento da saída do sistema. O *score* ou valor de aptidão de uma partícula é o valor de máximo sobressinal e tempo de acomodação calculados a partir da(s) saída(s). Uma partícula será considerada solução quando os valores de máximo sobressinal e tempo de acomodação calculados apresentarem valores menores ou iguais aos valores especificados pelo projetista do controlador.

Após calcular o *score* de todas as partículas o próximo passo é atualizar os vetores $Pbest_i\,(t) = \{Pbest_{\mathrm{i1}}(\mathrm{t}), \dots, Pbest_{\mathrm{in}}(\mathrm{t})\}$, a melhor posição encontrada pela partícula $i$ até a geração $t$ e $Gbest(t) = \{Gbest_1(\mathrm{t}), \dots, Gbest_{\mathrm{n}}(\mathrm{t})\}$, a melhor posição encontrada pelo enxame até a geração $t$. Na sequência, verifica-se se o vetor $Gbest$ atende aos critérios de parada. Existem algumas maneiras de definir o critério de parada, seja por número de gerações ou por estagnação das partículas. O critério de parada utilizado neste trabalho foi a descoberta de uma solução. Entretanto, se o algoritmo atingir um total de 300 gerações os valores do vetor $Gbest$ atual serão utilizados como solução.

Caso o critério de parada seja satisfeito, o algoritmo armazena a solução e encerra a sua execução. Caso contrário, utilizam-se as funções de evolução em busca de melhorias nas posições das partículas para que haja uma convergência direcionada para uma solução.

A atualização do vetor velocidade que tem o papel de auxiliar a convergência da partícula é mostrada na Equação (2), a seguir:

$$v_{in}(\mathrm{t}+1) = v_{in}(\mathrm{t}) + c_1 r_1 (Pbest_{in} - x_{in}) + c_2 r_2 (Gbest_n - x_{in}) \quad (2)$$

Onde $c_1$ e $c_2$ são coeficientes de aceleração que fazem com que a partícula contribua de forma cognitiva e social, são referentes à $Pbest$ e $Gbest$, respectivamente. $r_1$ e $r_2$ são dois números randômicos gerados entre 0 e 1. A atualização da posição de cada partícula é feita com o incremento da velocidade, conforme a Equação (3), abaixo:

$$x_i(\mathrm{t}+1) = x_i(\mathrm{t}) + v_i(\mathrm{t}+1) \quad (3)$$

Pode-se notar que as equações (2) e (3) não apresentam meios para limitar os valores do vetor velocidade, contudo, o algoritmo PSO é capaz de encontrar a região do ótimo global de maneira rápida, mas uma vez dentro desta região, possivelmente enfrentará dificuldade em atualizar o vetor velocidade para prosseguir numa busca mais refinada. Para tratar com este problema um peso denominado de peso inercial ($w$) é introduzido para ponderar a velocidade da geração anterior. A Equação canônica de atualização da velocidade apresentada em (2) passa a ser:

$$v_{in}(\mathrm{t}+1) = w v_{in}(\mathrm{t}) + c_1 r_1 (Pbest_{in} - x_{in}) + c_2 r_2 (Gbest_n - x_{in}) \quad (4)$$

Agora, com a manipulação do peso inercial é possível manipular o algoritmo para que faça buscas mais abrangentes ou com maior nível de refinamentos. Valores maiores para o peso inercial correspondem com buscas globais, enquanto que valores menores possibilitam buscas locais. Para aprimorar a convergência do algoritmo PSO para uma solução é aconselhável utilizar valores maiores do peso inercial no início da execução do algoritmo e diminuir no decorrer da execução. Isto é possível com o cálculo a adaptação de $w$ de acordo com a equação a seguir (Shi and Eberhart, 1998).

$$w = \begin{cases} w_{max} - \dfrac{w_{max} - w_{min}}{ger_{max}}\, ger, & ger \le ger_{max} \\ ger_{min}, & ger \ge ger_{max} \end{cases}$$

Onde $w_{max}$ e $w_{min}$ são os valores de máximo e mínimo que o peso inercial pode assumir e $ger$ representa o número de gerações. O valor de $w$ inicia em um patamar mais elevado e decresce com a evolução das gerações, chegando a um valor mínimo quando a geração corrente ultrapassa a geração escolhida como máxima.

Outros parâmetros, além de $w$ que podem auxiliar o algoritmo em buscas globais ou locais são os coeficientes de aceleração $c_1$ e $c_2$ que controlam diretamente o impacto individual ($Pbest$) e coletivo ($Gbest$) de cada partícula em relação à evolução do enxame.

De acordo com a Figura 1 é possível descrever o algoritmo PSO em 6 etapas:

*Etapa 1: Atribuir valores aleatórios aos vetores posição e velocidade de todas as partículas do enxame.*

*Etapa 2: Avaliar todas as partículas com a função de custo (Fitness).*

*Etapa 3: Atualizar $Pbest$ e $Gbest$ caso possuam melhor score que os respectivos valores da geração anterior.*

*Etapa 4: Caso uma solução seja encontrada ou geração atual igual a 300 armazenar o valor de $Gbest$ e encerrar a execução do algoritmo. Caso contrário, prosseguir para a próxima etapa.*

*Etapa 5: Atualizar os vetores posição e velocidade de acordo com as Equações (3) e (4) e voltar para Etapa 2.*

Os resultados que utilizaram este algoritmo são apresentados na Seção 4, logo em seguida.

## 4 Resultados

Para averiguar o desempenho do algoritmo PSO alguns testes foram realizados e comparações foram feitas com o Algoritmo Genético obtido de Silveira Júnior et al. (2012). Vale ressaltar, que o algoritmo genético utilizado para comparação teve alguns de seus parâmetros ajustado por meios de testes exaustivos. Foram testados 8 tamanhos de população (20, 40, 60, 80, 100, 120 e 160), 5 faixas de inicialização para os possíveis valores da matriz de ganho de realimentação ([-1, 1], [-3, 3], [-4, 4], [-5, 5] e [-20, 20]) e dois métodos de seleção de indivíduos para reprodução (torneio e roleta com *ranking*). Estes parâmetros quando combinados formaram 80 configurações diferentes para o AG. Em cada configuração foram feitas 40 execuções totalizando 3200 testes.

Portanto, o AG utilizado neste artigo para comparações com o algoritmo PSO já foi bastante testado e apresenta uma configuração bem sintonizada o que o torna apto a encontrar boas soluções para o projeto de controlador respeitando as condições de desempenho, sobressinal e tempo de acomodação.

A seguir estão as matrizes do sistema utilizado para realizar as simulações.

$$A = \begin{bmatrix} -0.9221 & -0.2315 & 0.0002 & 0.0095 \\ 0.2315 & -0.9221 & -0.0003 & -0.0007 \\ 3.6129 & 45.8243 & 0 & 0.3652 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.9767 & 0 \\ 0 & 0.9767 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} e$$

$$C = \begin{bmatrix} 0.5908 & -0.0747 & 0 & -0.0025 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Para este sistema foi estabelecido um máximo sobressinal de 10% e um tempo de acomodação de 6 segundos. Na busca pela matriz de ganho feita pelo algoritmo PSO e pelo AG é estipulada uma faixa de valores possíveis para os elementos da matriz. Isto porque os algoritmos de busca para este propósito podem facilmente encontrar uma matriz de ganho com valores elevados, que aplicado ao sistema de controle produz uma resposta dentro das condições estabelecidas, mas, que na prática não é implementável, pois demanda de um gasto de energia maior do que o disponível.

Os valores dos principais parâmetros ajustáveis utilizados pelo algoritmo PSO e pelo AG estão apresentados na Tabela 1 e na Tabela 2, respectivamente.

Tabela 1. Parâmetros utilizados pelo algoritmo PSO.

| **Parâmetro PSO** | **Valor** |
|---|---|
| Faixa de valores possíveis | [-8, 8] |
| Peso inercial $w$ | 0.6 |
| Coeficiente aceleração $c_1$ | 0.8 |
| Coeficiente aceleração $c_2$ | 0.8 |
| Tempo de acomodação | 6 seg. |
| Máximo sobressinal | 10% |
| Quantidade de partículas | 80 |

Tabela 2. Parâmetros utilizados pelo AG.

| **Parâmetro AGD** | **Valor** |
|---|---|
| Faixa valores | [-8, 8] |
| Tipo de seleção | Torneio |
| Taxa de cruzamento | 60% |
| Taxa de mutação | 2% |
| Tempo de acomodação | 6 seg. |
| Máximo sobressinal | 10% |
| Quantidade de indivíduos | 80 |

O algoritmo PSO encontrou uma solução na 6ª geração, os valores da matriz de ganho são mostrados a seguir:

$$K = \begin{bmatrix} 5.953501 & 1.044878 & 0.099201 & 5.398498 \\ 1.850765 & 7.834564 & 6.327643 & 5.509696 \end{bmatrix}$$

Já o AG encontrou uma solução na 4ª geração, a solução é apresentada abaixo:

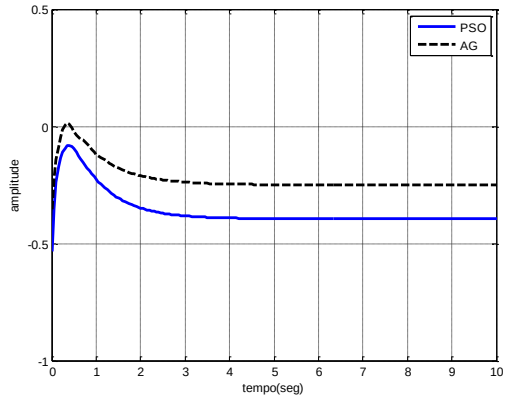
$$K = \begin{bmatrix} 6.438442 & 1.510518 & 1.198511 & 4.862449 \\ -1.963439 & 3.970014 & 5.065436 & 4.719975 \end{bmatrix}$$

Os autovalores obtidos pelos algoritmos PSO e pelo AG juntamente com os autovalores do sistema sem controle são apresentados na Tabela 3, a seguir:

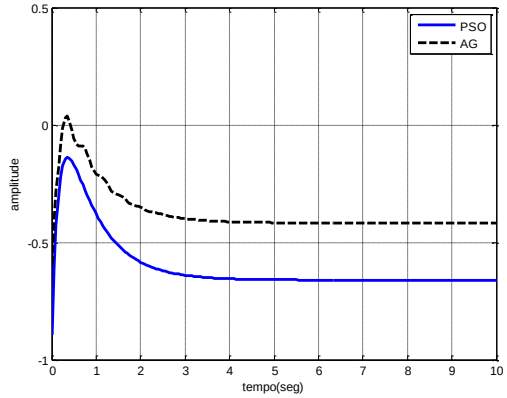
Tabela 3. Autovalores dos Algoritmos PSO, AG e sistema sem controle.

| **Autovalores** | | |
|---|---|---|
| Sem Controle | PSO | AG |
| 0.6348<br>-1.1407<br>-0.6691+0.3886i<br>-0.6691- 0.3886i | -3.8819+16.1756i<br>-3.8819-16.1756i<br>-6.2305<br>-1.3166 | -1.8003+15.0502<br>-1.8003-15.0502<br>-7.1786<br>-1.2309 |

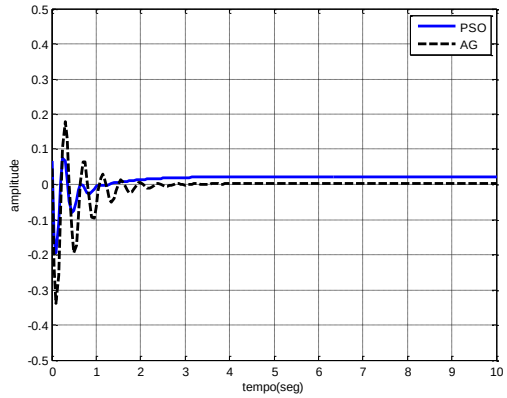
As saídas para o sistema controlado pelo algoritmo PSO são apresentadas juntamente com as saídas utilizando o AG como técnica de busca da matriz de realimentação de estados. De acordo com as dimensões das matrizes A, B, C e D do sistema, este possui 4 saídas. As Figuras 2(a), 2(b), 2(c) e 2(d) ilustram as 4 saídas obtidas.



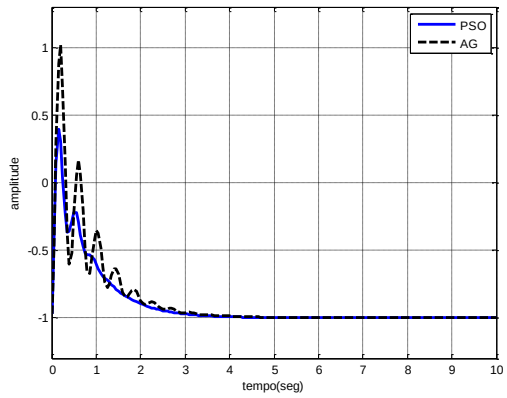

(a) Saída 1.




(b) Saída 2.




(c) Saída 3.




(d) Saída 4.

Figura 2. Comparação entre o algoritmo PSO e o AG.

O algoritmo PSO obteve saídas ligeiramente melhores em comparação com as saídas obtidas pelo AG. Isto não significa que em todas as execuções o algoritmo PSO irá obter vantagem, pois, o que está sendo levado em consideração é que os dois conseguiram encontrar valores para a matriz de realimentação de estados e quanto aplicado ao sistema obteve respostas obedecendo aos critérios de tempo de acomodação e sobressinal.

## 5 Conclusão

Neste artigo foi desenvolvido um algoritmo PSO com a tarefa de buscar parâmetros para um controlador multivariável. O desempenho do PSO foi comparado com um dos algoritmos evolutivos mais difundidos, o algoritmo genético. O AG utilizado para comparação foi bastante testado no ajuste de seus parâmetros de tamanho de população, tipo de seleção e inicialização.

Apesar dos resultados serem bem parecidos com os obtidos por meio do AG, mostra que são satisfatórios no caso do algoritmo PSO, devido ao fato deste último não ter passado por nenhuma campanha de testes para ajustar seus parâmetros, a exemplo de peso inercial e coeficientes de aceleração que podem influenciar diretamente na convergência do algoritmo para uma solução.

Os resultados mostram que o algoritmo PSO conseguiu solucionar o problema de forma satisfatória. E que existem alternativas para que a complexidade temporal das soluções seja ainda menor. Um melhor ajuste nos parâmetros do algoritmo PSO ou a adaptação dos parâmetros durante sua execução ou ainda o emprego de técnicas de divisão do enxame em bairros inserindo certo nível de paralelismo pode levar este algoritmo para uma aplicação que utilize estimação online.

## Agradecimentos

Os autores agradecem a CAPES pelo suporte financeiro fornecido através do Projeto PE-087/2008 do Programa Pró-Engenharias e ao CNPq também por auxílio financeiro.

## Referências

Ogata, K. (2010). Engenharia de Controle Moderno, 5ª ed. Prentice-Hall.

Dorf, R. C. and Bishop, R. H. (2009). Sistemas de Controle Modernos, 11ª ed. LTC Editora.

Kennedy, J. and Eberhart, R. Particle swarm optimization. Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service, Piscataway, NJ, 1995.

Oliveira, P. S.; Barros, L. S. and Silveira Júnior, L. G. Q. (2010). Genetic Algorithm Applied to State Feedback Control Design. In: IEEE Latin

America Transmission and Distribution Conference and Exposition – T&D 2010, São Paulo.

Parsopoulos, K. E. and Vrahatis, M. N. (2004). On the computation of all global minimizes through particle swarm optimization. IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 221-224, Jun. 2004.

Janson, S. and Middendorf, M. (2005). A hierarchical particle swarm optimizer and its adaptive variant. IEEE Trans. Syst., Man, Cyber. B, Cybern., vol. 35, no. 6, pp. 1272-1282, Dec. 2005.

Min, H.; Jianchao, F. and Wang, J. (2011). A Dynamic Feedforward Neural Network Based on Gaussian Particle Swarm Optimization and its Application for Predictive Control. IEEE Trans. On Neural Networks, vol. 22, no. 9, Sep 2011.

Shi, Y. and Eberhart, R. C. (1998). A Modified Particle Swarm Optimizer, In: Proocedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp 69-73.

Silveira Júnior, L. G. Q., Barros, L. S., Oliveira, P. S., Projeto de Sistema de Controle Multivariável Baseado em Algoritmo Genético Otimizado. In: Congresso Brasileiro de Automática – CBA Campina Grande, 2012.

Barros, L. S., Mota, W. S., Silveira Júnior, L. G. and Oliveira, P. S., DFIG Rotor Side Control through Gain-Scheduling Designed by Genetic Algorithm. In: IEEE/PES Transmission and Distribuition: Latin America Conference and Exposition, Montedideo 2012.