

DESENVOLVIMENTO E APLICAÇÃO DE INTERFACE EM MATLAB PARA IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS COM PSO

Silviane S. Alves – alves.silviane@gmail.com

Universidade Federal do Pará

Rua Augusto Corrêa

66075110 – Belém – Pará

Juan F. Vidal – jfvidal@ufpa.br

Caio Q. Silva – caio.que98@hotmail.com

Rafael M. H. Machado – machadoraf14@gmail.com

Gabriel B. de S. Silva – gabrielbssilva1@gmail.com

Orlando F. Silva – orfosi@ufpa.br

Resumo: Este artigo apresenta a proposta de uma ferramenta educacional criada para aprimorar o processo de ensino aprendizagem de metaheurísticas populacionais aplicada à engenharia de controle. Primeiramente foi escolhido um assunto a ser trabalhado, que foi o de identificação de sistemas, em seguida optou-se por utilizar como metaheurística o PSO (Particle Swarm Optimization), então foi feito um levantamento teórico acerca dos assuntos abordados, dando atenção especialmente ao algoritmo. Posteriormente foram elaboradas interfaces gráficas em ambiente MATLAB/GUI, fazendo uso de parâmetros de interesse que pudessem gerar gráficos e resultados que mostrassem o desempenho do PSO na identificação de sistemas. Por exemplo, os parâmetros do PSO podem ser modificados para que se possa, através da plotagem do sinal de saída do sistema, do sinal identificado e dos resultados apresentados, verificar como as alterações dos parâmetros do PSO influenciam no comportamento do mesmo. Os resultados foram analisados de acordo com o referencial utilizado, constatando-se a representação de seus aspectos qualitativos e quantitativos de forma eficaz e confiável. Como mecanismo de validação da proposta didática, foi criado um roteiro de atividades baseado na interface e este foi repassado aos alunos do 5º e 7º semestres do curso de Engenharia Elétrica da Universidade Federal do Pará. Ao final do roteiro, os estudantes responderam a um breve questionário a respeito da ferramenta proposta a eles e as repostas indicaram boa aceitação e impacto significativo em seus conhecimentos teóricos.

Palavras-chave: Metaheurísticas populacionais. Inteligência de enxame. Sistemas de controle.

1 INTRODUÇÃO

As metaheurísticas se tornaram muito utilizadas nos últimos anos. Em geral, se configuraram como base para muitas aplicações modernas em diversas áreas, principalmente no campo da engenharia, em especial na área de sistemas de controle (MIAZAKI, 2007; LAZINICA, 2009; LATHA *et al*, 2013; COUCEIRO & GHAMISI, 2015; VIDAL *et al*, 2015). Essa disseminação das metaheurísticas é devido à sua simplicidade, pois necessitam de poucas modificações nas estruturas dos algoritmos para adaptá-los ao problema em questão, e a sua exatidão. Além disso, apresentam melhor capacidade de otimização e velocidade de resposta. Tais características fazem delas ferramentas importantes para muitos problemas de engenharia de controle e tornando-as objeto de estudo nas últimas décadas.

Dentro das metaheurísticas, destaca-se a de Inteligência de Enxame (IE), pois seus algoritmos podem se adaptar facilmente a procedimentos e projetos de controladores clássicos, além de serem mais eficientes na solução de problemas de otimização dimensional mais elevada. Os algoritmos inspirados em IE também podem ser modificados para uma classe de modelos de processos instáveis independente da ordem do modelo (LATHA *et al*, 2013).

No entanto, apesar do crescente avanço no conhecimento e aplicação das metaheurísticas, dentro da comunidade acadêmica, a maioria dos discentes da graduação em Engenharia Elétrica não está familiarizada com o assunto e menos ainda com o uso de tais técnicas para solução de problemas de controle, tendo em vista que elas normalmente não são explanadas em sala de aula, em disciplinas como sistemas de controle, quanto às técnicas mais consagradas na literatura pertinente.

Nesse contexto, este trabalho visa implementar, por meio do ambiente GUIDE do *software* MATLAB, um conjunto de interfaces gráficas que utilizam a IE (mais especificamente o *Particle Swarm Optimization* - PSO), para apresentar ao aluno uma visão sistêmica sobre as metaheurísticas, como elas podem auxiliá-lo no ensino aprendido de Engenharia de Controle e instigá-lo a aprofundar seus conhecimentos no assunto.

Com o intuito de validar o uso da simulação para o objetivo proposto, preparou-se uma aula e um roteiro de atividades para os alunos do curso de Engenharia Elétrica, a fim de direcioná-los e fazê-los acompanhar as simulações para que os mesmos pudessem ter uma experiência agradável com a metaheurística usada, tirassem suas próprias conclusões a respeito da ferramenta, ampliassem seus conhecimentos, bem como introduzir o assunto na rotina acadêmica. O roteiro também incluiu um *feedback* de avaliação.

2 ABORGAGEM TEÓRICA

2.1 *Particle Swarm Optimization* (PSO)

O PSO foi projetado e proposto em 1995 por Eberhart e Kennedy. O PSO é um algoritmo de busca baseado na população de pássaros e que visa simular graficamente o comportamento social dos mesmos. O objetivo da simulação era o de encontrar padrões que permitissem que os indivíduos voassem em uma determinada direção e quando a mudassem que o fizessem de forma sincronizada sem perder a direção (LAZINICA, 2009; COUCEIRO & GHAMISI, 2015).

No algoritmo PSO, cada partícula do enxame é formada por um vetor posição e um vetor velocidade, que representam respectivamente a solução dentro do espaço de busca multidimensional e a direção na qual a partícula está se movimentando em uma determinada iteração do algoritmo. Cada partícula atualiza sua velocidade baseada na velocidade atual e na

melhor posição que explorou até o momento e, conjuntamente, na melhor posição global verificada pelo enxame conforme a “Equação (6)”, em seguida sua posição é atualizada com base neste novo valor de velocidade usando a “Equação (7)”, onde ω é o fator de inércia, c_1 é o fator cognitivo, c_2 é o fator social, r_1 e r_2 são números randômicos entre 0 e 1, $V_i(t)$ e $V_i(t+1)$ são respectivamente a velocidade atual e futura da partícula i , $P_i(t)$ é a melhor posição visitada por esta partícula até o momento, $P_g(t)$ é a melhor posição visitada pelo enxame e θ_i é a posição da partícula i (PARSOPOULOS, 2010).

$$V_i(t+1) = \omega \cdot V_i(t) + c_1 r_1 (P_i(t) - \theta_i(t)) + c_2 r_2 (P_g(t) - \theta_i(t)) \quad (6)$$

$$\theta_i(t+1) = \theta_i(t) + V_i(t+1) \quad (7)$$

Em cada atualização é aplicada uma função *fitness* para validar o sucesso da posição. Então, o processo do PSO é iterado em um número determinado de vezes ou até atingir um valor de resposta satisfatório, baseada em um erro máximo aceito que é estabelecido conforme o critério de parada (LAZINICA, 2009; COUCEIRO & GHAMISI, 2015).

Depois da atualização da velocidade e da posição de todo o enxame é a melhor posição de cada partícula é atualizada juntamente com a melhor posição do enxame.

Graças a sua simplicidade nas equações e nos algoritmos, bem como um número pequeno de parâmetros em relação a outros algoritmos de otimização, o PSO é um método de fácil implementação e bastante utilizado dentre outros algoritmos de busca coletiva.

2.2 Modelagem e identificação de sistemas

Sistemas dinâmicos, ou com memória, são aqueles em que o estado atual depende de estados passados. A dinâmica de muitos sistemas mecânicos, elétricos, térmicos, econômicos, biológicos, dentre outros, pode ser descrita em termos de equações diferenciais e estas, por sua vez, podem ser obtidas a partir das leis físicas que regem o sistema. Por exemplo, as leis de Newton regem sistemas mecânicos e as leis de Kirchhoff regem sistemas elétricos. O conjunto de equações que representa a dinâmica de um sistema é denominado modelo matemático. Normalmente modelos obtidos a partir de leis físicas visam precisão na representação. (OGATA, 2011). Outra maneira de se determinar um modelo matemático é a partir de dados observados do próprio sistema dinâmico, esse processo denomina-se identificação de sistemas. Neste caso, busca-se um modelo que, mesmo não tão preciso, represente suficientemente bem o sistema. Independentemente da técnica utilizada para a obtenção de um modelo, no que diz respeito à sistemas de controle, tanto para a análise quanto para o projeto de um controlador os modelos são fundamentais. Por fim, muitos sistemas reais podem ser representados de modo satisfatório por equações diferenciais de 1ª ou 2ª ordem.

a) Representação de Sistemas de 1ª e 2ª Ordem

Sistemas de 1ª ordem podem ser descritos pela função de transferência apresentada na “Equação (1)”, onde K é o ganho do sistema e T é a constante de tempo dada em segundos. Sistemas de 2ª ordem podem ser representados pela função de transferência apresentada na “Equação (2)”, sendo ω_n a frequência natural, ζ o coeficiente de amortecimento e K o ganho do sistema (OGATA, 2011).

$$G(s) = \frac{K}{Ts+1} \quad (1)$$

$$G(s) = \frac{K\omega_n^2}{s^2+2\zeta\omega_n s + \omega_n^2} \quad (2)$$

b) Identificação de Sistemas baseado na Função do Erro Médio Quadrático

O erro médio quadrático (EMQ) é uma métrica estatística para avaliar a diferença entre uma estimativa e o verdadeiro valor da quantidade estimada. É utilizado como um critério para selecionar o melhor estimador entre muitos em potencial.

Sendo assim, para realizar a identificação de sistemas, utilizou-se uma metodologia baseada na minimização do EMQ entre a resposta do sistema que será identificado e as respostas criadas a partir das partículas do enxame do PSO com estimação de parâmetros (VIDAL *et al.*, 2015). A “Figura 1” apresenta a estrutura utilizada para realizar a identificação de sistemas de 1ª e 2ª ordem com a aplicação do PSO. Primeiramente especifica-se a ordem da função de transferência do modelo que será identificado. Depois, aplica-se um sinal de entrada degrau unitário $u(t)$ ao sistema e a função de transferência $G_n(s)$ representada pela partícula n da população. Em seguida, coletam-se os sinais de saída do sistema real $y(t)$ e $y_n(t)$. Calcula-se o erro entre os sinais de saída conforme a Equação (3).

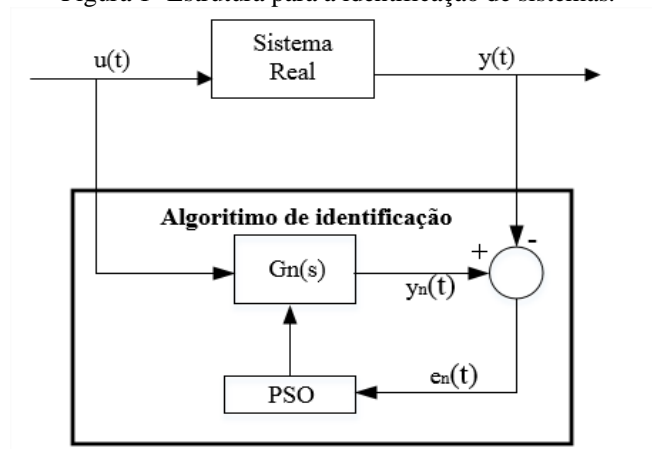
$$e_n(t) = y_n(t) - y(t) \quad (3)$$

Então, obtido e_n calcula-se o EMQ de acordo com a “Equação (4)”, sendo m o tamanho do vetor $e_n(t)$, e seguidamente calcula-se a função de avaliação conforme a “Equação (5)” para cada partícula do enxame.

$$\bar{e}_n(t) = \frac{\sum_{t=1}^m [e_n(t)]^2}{m} \quad (4)$$

$$F_n = \frac{1}{0.01 + \bar{e}_n} \quad (5)$$

Figura 1- Estrutura para a identificação de sistemas.



Fonte: Os autores.

A finalidade é de obter as partículas com os maiores valores de avaliação, em razão destas possuírem o menor EMQ entre a sua resposta específica e a resposta do sistema real.

3 METODOLOGIA

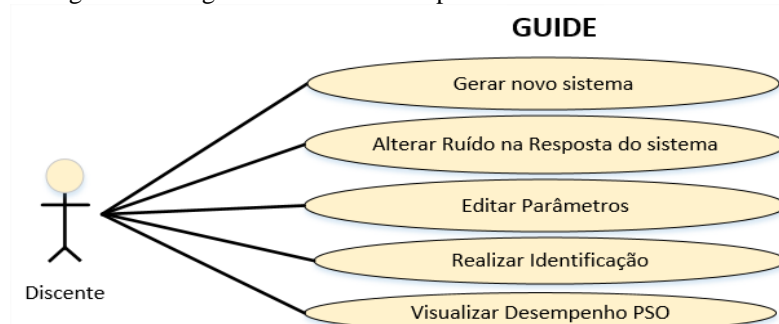
O Matlab é um *software* composto por um conjunto de diferentes ferramentas que o torna uma excelente solução para aplicações em diferentes campos de estudos tais como a área de sistemas de controle, inteligência computacional, processamento de sinais entre outras. É usado por usuários de diferentes níveis técnicos, desde discentes de graduação até pesquisadores experientes, isso graças a sua gama de recursos e versatilidade. Outro recurso interessante oferecido pelo Matlab é a *Graphic User Interface* (GUI), uma ferramenta para criação de janelas gráficas interativas com os usuários que proporciona a construção de modelos através dos componentes da GUI, como botões, caixa de texto e *popup menu*. Os usuários podem alterar seus parâmetros de maneira simples e observar o resultado da computação resultante destas alterações.

Os subtópicos a seguir, descrevem a metodologia utilizada neste trabalho para a criação da interface proposta para o ensino de conceitos de sistemas de controle e a aplicação de técnicas metaheurísticas na identificação de sistemas dinâmicos de 1ª e 2ª ordem utilizando a ferramenta GUI do Matlab.

3.1 Interface Proposta

De acordo com os conceitos apresentados no tópico 2 podemos notar que a metodologia para realizar a identificação de sistemas com metaheurísticas usando a função *fitness* baseada no EMQ é eficiente, porém envolve a parametrização de um número alto de parâmetros do algoritmo usado. Diferentes conjuntos de valores de parâmetros podem levar a desempenhos distintos, sendo assim um estudo exaustivo para o entendimento e efeito de cada parâmetro se faz necessário. A interface foi desenvolvida com o objetivo de fornecer um ambiente simples para que os usuários possam realizar de maneira fácil e rápida mudanças nas configurações do PSO e observar o efeito de tais mudanças no desempenho do algoritmo. Para a definição de quais funcionalidades seriam implementadas na interface foi utilizado o diagrama de caso de uso da UML (do Inglês, *Unified Modelling Language*) (MELO, 2002), a “Figura 2” mostra o diagrama de caso de uso para o sistema desenvolvido.

Figura 2 – Diagrama de caso de uso para a interface desenvolvida.

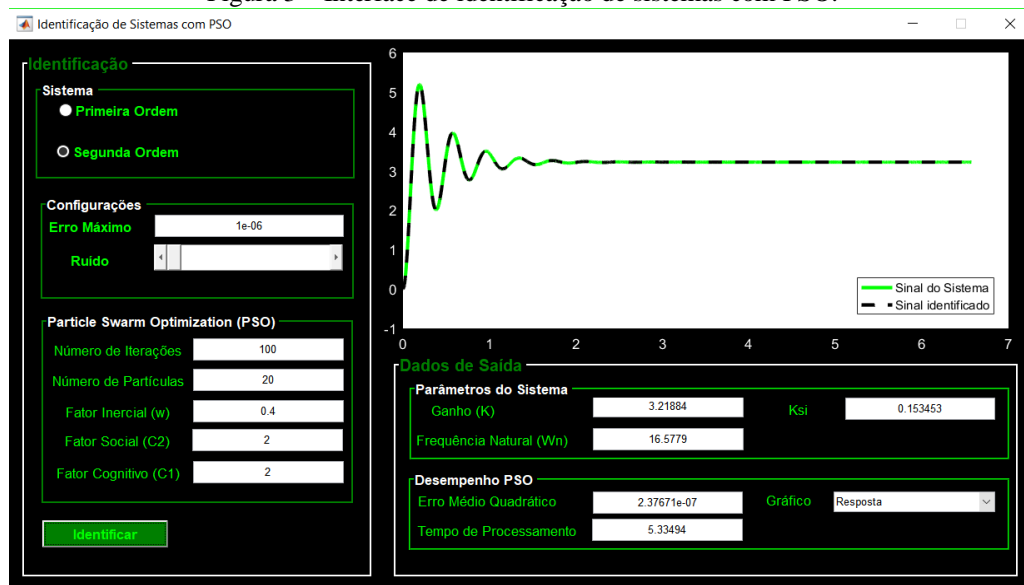


Fonte: Os autores.

Como se observa na “Figura 2”, o sistema tem cinco funcionalidades principais: Gerar um novo sistema; Alterar o ruído na resposta do sistema; Editar os parâmetros; Realizar a identificação e Visualizar o desempenho do PSO, que serão melhor descritas nos próximos tópicos. A “Figura 3” mostra a interface completa desenvolvida que é composta por três seções: identificação, dados de saída e o campo para plotagem dos gráficos. Na seção de identificação estão contidos os parâmetros do algoritmo PSO, o tipo de sistema a ser identificado, valor de ruído da resposta e o erro máximo que se deseja alcançar entre a

resposta identificada e a real. Na seção de dados de saída são disponibilizados os dados referentes ao desempenho do PSO na identificação, ou seja, os parâmetros do sistema encontrado, erro médio obtido na identificação e o tempo de processamento do PSO.

Figura 3 – Interface de identificação de sistemas com PSO.



Fonte: Os autores.

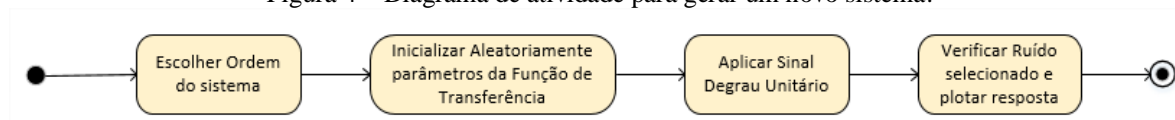
3.1 Funcionalidades da Interface Desenvolvida

Nesta seção são descritos com mais detalhes as funcionalidades disponibilizadas na interface assim como o diagrama de atividades de cada estudo de caso para facilitar o entendimento da lógica implementada.

a) Funcionalidade: Gerar Novo Sistema

Esta funcionalidade permite que o usuário escolha a característica do sistema que será identificado usando o PSO - as opções são um sistema de 1ª ou de 2ª ordem. A Figura 4 representa o diagrama de atividade para execução desta função. Inicialmente após a seleção no campo “Sistema” a GUIDE irá selecionar o modelo correspondente à opção do usuário e de forma aleatória inicializará os seus parâmetros da função de transferência. Uma vez que a função de transferência foi criada, um sinal de entrada do tipo degrau é aplicado e gera a resposta que é então plotada no gráfico da interface.

Figura 4 – Diagrama de atividade para gerar um novo sistema.

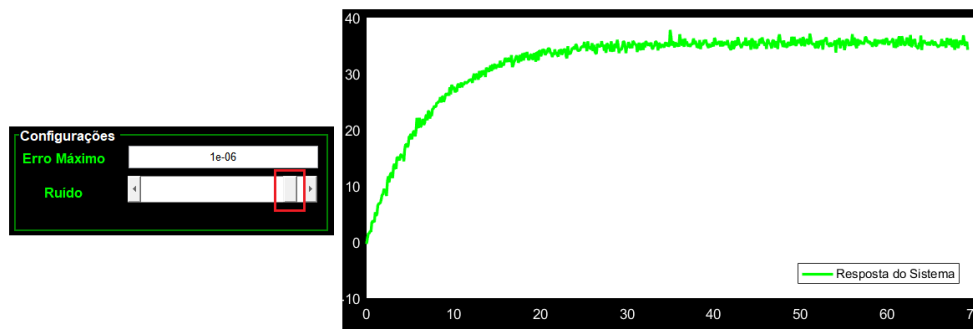


Fonte: Os autores.

b) Funcionalidade: Alterar Ruído na Resposta do Sistema

Permite que o usuário adicione um ruído na resposta a ser identificada para que possam ser aproximadas das condições da maioria das aplicações reais de identificação. Para fazer isso o usuário precisa só alterar a entrada do componente *slider* “Ruído” da interface. A “Figura 5” mostra uma resposta com o ruído adicionado.

Figura 5 – Gráfico de uma resposta sem ruído.

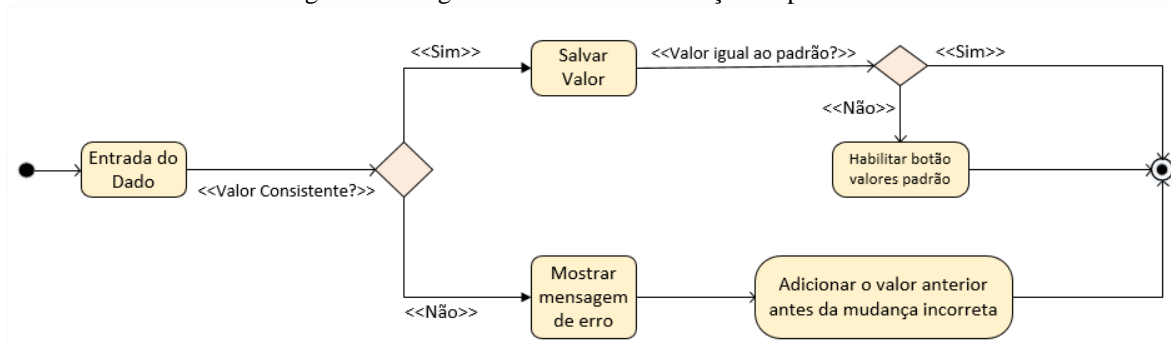


Fonte: os autores.

c) Funcionalidade: Editar Parâmetros

Permite que os usuários alterem os parâmetros do algoritmo PSO e o valor de erro desejado na identificação. O objetivo é que os discentes possam testar e comprovar os conceitos explorados em sala de aula de forma a absorvê-los de maneira mais fácil. A “Figura 6” mostra o diagrama de atividade desta funcionalidade. Após a inserção do novo valor desejado o sistema avalia a consistência do dado. Quando uma informação inconsistente é inserida (valores negativos ou não numéricos), o sistema envia uma mensagem informando o erro e indicando os valores aceitáveis para aquele parâmetro e então altera o dado errado colocando o valor que estava antes da mudança incorreta. No caso de um valor consistente ser inserido, o sistema salva o novo valor e verifica se o mesmo é igual ao valor padrão para aquele parâmetro, no caso de ser diferente o sistema habilita um botão que permite o *reset* de todos os parâmetros para os valores padrões.

Figura 6 – Diagrama de atividade de edição de parâmetros.



Fonte: Os autores.

d) Funcionalidade: Realizar Identificação

Possibilita que os usuários realizem a identificação dos parâmetros do sistema que gerou a resposta mostrada no gráfico. Após pressionar o botão “Identificar” a interface passa para o PSO os valores dos parâmetros setados e os sinais de saída e de entrada aplicados no sistema. Por fim, depois de realizar o processamento, o algoritmo retorna os valores encontrados e os exibe no campo “dados de saída”.

e) Funcionalidade: Visualizar Desempenho PSO

Possibilita aos usuários visualizarem o valor do EMQ entre a resposta gerada e a estimada, o tempo de simulação e escolher o tipo de gráfico (resposta, *fitness* ou erro) que querem visualizar.

3.2 Avaliação da Interface

Para avaliação da interface desenvolvida utilizou-se um roteiro de simulação a fim de orientar os alunos, do 5º e 7º semestres de Engenharia Elétrica da Universidade Federal do Pará, a fazerem uso da interface de identificação de sistemas e assim tirar suas conclusões e compará-las com seus conhecimentos teóricos. Antes de iniciarem as atividades foi mostrado como fazer uso da interface e configuração dos parâmetros do PSO através de uma aula em laboratório. Ao final do roteiro, solicitou-se aos alunos que respondessem perguntas objetivas (para fins de levantamento estatístico de avaliação da interface), destacando as mostradas na "Tabela 1".

Tabela 1 – Perguntas de avaliação da interface.

Nº	Perguntas
01	A interação com o sistema é agradável e incita a curiosidade do discente em relação aos assuntos abordados em aula?
02	O uso do <i>software</i> permitiu que você aprofundasse os conhecimentos sobre o assunto abordado?
03	Você acredita que o <i>software</i> pode ser um bom instrumento para o ensino das metaheurísticas populacionais aplicada à Engenharia de Controle?
04	Qual o seu nível de conhecimento sobre os assuntos abordados antes das explicações e uso do <i>software</i> ?

Fonte: Os autores.

4 RESULTADOS

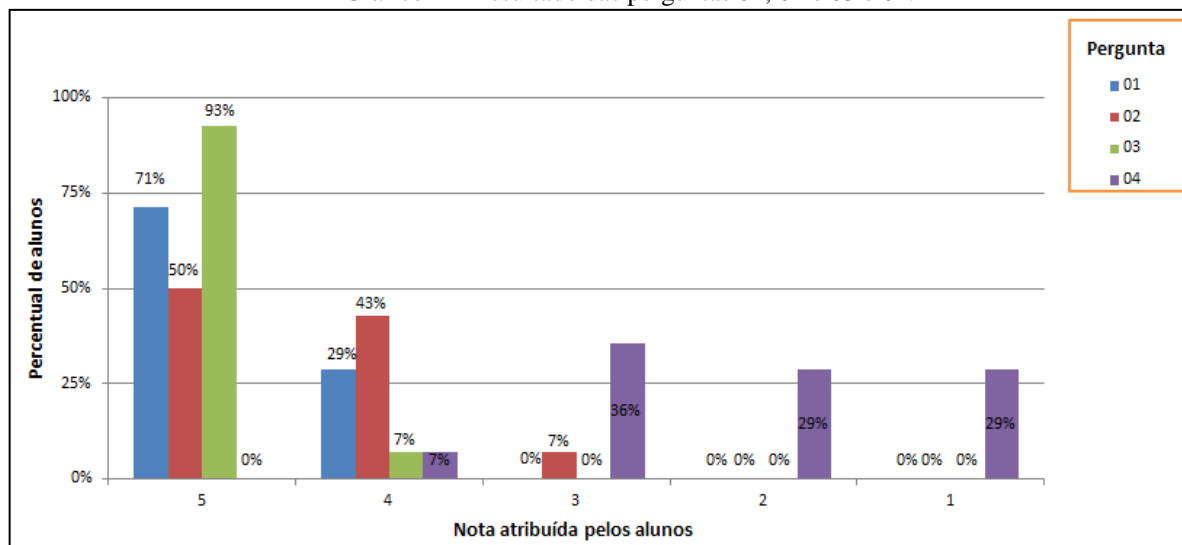
4.1 Análise dos resultados

Os resultados obtidos nas simulações de cada interface foram comparados à teoria referente aos tópicos trabalhados, dando-se atenção para a influência de cada parâmetro fornecido pelo usuário. Os alunos conseguiram perceber que parâmetros como fator cognitivo (c_1) e fator social (c_2), além do fator inercial (ω), influenciam significativamente o desempenho do PSO.

4.2 Avaliação das atividades práticas

Foram levantadas estatísticas a partir das perguntas da "Tabela 1" que foram respondidas ao final das atividades. Para a avaliação pediu-se que fosse conferida uma nota de 1 a 5 a cada pergunta, onde 5 representa a nota máxima. Do "Gráfico 1" observa-se que para a pergunta 1, 71% dos alunos consideram que a interface instiga a curiosidade do discente acerca do assunto abordado. Para a pergunta 2, 50% dos alunos atribuíram nota 5 e 43% nota 4, demonstrando que os mesmos puderam aprofundar os conhecimentos. Na pergunta 3, 93% dos alunos atribuíram nota 5, indicando que o uso da interface como instrumento de ensino de matheurísticas populacionais é viável. Na pergunta 4, 36% dos alunos atribuíram nota 3 e 29% atribuíram nota 2 e 1 indicando que seus conhecimentos eram razoáveis, relativamente ou baixos acerca do tema antes da apresentação da ferramenta o que comprova a falta de contato dos discentes da graduação com o tema explorado.

Gráfico 1 – Resultado das perguntas 01, 02 e 03 e 04.



Fonte: Os autores.

5 COSIDERAÇÕES FINAIS

De modo geral, por meio do questionário de avaliação do *software*, pôde-se comprovar o êxito dessa metodologia. A utilização da interface desenvolvida foi capaz de estimular o interesse dos discentes pelo assunto abordado, de contribuir para a melhor compreensão do tema, demonstra grande potencial como instrumento para o ensino de Engenharia de Controle e mostrou-se como meio de estímulo ao aprendizado da disciplina.

Contudo, pelas sugestões apresentadas também no questionário, percebeu-se que ainda há muitas melhorias a fazer. Uma delas é poder fazer a identificação de sistemas reais, além de poder salvar os dados em outros formatos como txt ou xls (Excel). Para trabalhos futuros espera-se poder incluir o uso de outras metaheurísticas como ACO (*Ant Colony Optimization*) ou BFO (*Bacterial Foraging Optimization*) e juntá-las numa interface mais geral.

REFERÊNCIAS

- LATHA, K.; RAJINIKANTH, V.; SUREKHA, P. M. PSO-based PID controller design for a class of stable and unstable systems. **Journal Hindawi**, Chennai, v. 2013, p.2, 2013.
- LAZINICA, Aleksandar (org.). **Particle swarm optimization**. Kirchengasse: InTech. 2009.
- MELO, Ana Cristina. **Desenvolvendo Aplicações com UML do Conceitual à Implementação**. 1º ed. Rio de Janeiro: Brasport. 2002.
- MIAKAZI, Mauro. **Sistema de controle multi-robô baseado em colônia de formigas artificiais**, 2007. Tese (Mestrado) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2007.
- OGATA, Katsujko. **Engenharia de controle moderno**. 5ª ed. São Paulo: Pearson. 2011.
- PARSOPOULOS, Konstantinos E. **Particle swarm optimization and intelligence: advances and applications: advances and applications**. New York: IGI Global. 2010.

COUCEIRO, Micael; GHAMISI, Pedram. **Fractional order darwinian particle swarm optimization: applications and evaluation of an evolutionary algorithm**. New York: Springer. 2015.

VIDAL, Juan F.; FREITAS, Victor S.; SILVA, Orlando. F.; MEDEIROS, Renan L. P.; CASTRO, Adriana R. G.; Estratégia de Identificação de sistemas e Controle PID via Algoritmo Genético em Modelo de Ilhas. In: XII-SABI-Simpósio Brasileiro de Autonomia Inteligente, 2015, Rio Grande do Norte. **Anais**. Natal, 2015.

DEVELOPMENT AND APPLICATION OF INTERFACES IN MATLAB TO IDENTIFICATION OF SYSTEMS BASED ON PSO

Abstract: *This article presents the proposal created to make better the learning process of population metaheuristics applied to control engineering. First chosen the subject to be work, which was the identification of systems, next chosen use how metaheuristics the PSO (Particle Swarm Optimization), then was done a theoretical survey was made on the subjects discussed, giving special attention to the algorithm. After graphical interfaces was developed on the environment MATLAB/GUI, making use parameters of interest that could generate graphs and results that show the performance of the PSO in the identification of systems. For example, the parameters of PSO can be modified so that, by plotting exit signal of the system, of the identified signal and the presented results, to verify how to alter the parameters of the PSO influences the behavior of the same. The results was analyze according to the reference used, showing the representation qualitative and quantitative aspects in an effective and reliable. The mechanism of validation the didactic proposal, an interface script created and this passed on to the students of the 5th and 7th semesters of the Electrical Engineering course of the Universidade Federal do Pará. The final script, the students answered a brief quiz about the implement proposed to them and the answers indicated good acceptance and significant impact on their theoretical knowledge.*

Key-words: *Population metaheuristic. Swarm Intelligence. Control Systems.*