

Filtragem Adaptativa IIR Através do Algoritmo de Otimização Swarm de Aprendizado Compreensivo

Leonardo Dagui Oliveira¹, Isaque Suzuki², Taufik Abrão³ & Paul Jean E. Jeszensky⁴

Resumo—Neste trabalho é avaliada a aplicação dos algoritmos de otimização por nuvem de partículas (PSO - *Particle Swarm Optimization*), PSO modificado (MPSO - *Modified PSO*), PSO com aprendizado compreensivo (CLPSO - *Comprehensive Learning PSO*) e genético (GA - *Genetic Algorithm*) para o problema de filtragem adaptativa IIR. A comparação é realizada através do compromisso desempenho *versus* complexidade, mais completo do que o encontrado normalmente na literatura, sendo a complexidade computacional expressa em termos do número de operações necessárias para o algoritmo atingir a convergência. A análise da identificação de sistemas foi realizada considerando filtros IIR com superfícies de erro uni e multimodais, ruído na medição do sinal desejado e avaliação da robustez do algoritmo relativa a variações no tamanho da população.

Palavras-Chave—IIR, Algoritmo Heurístico, Otimização por Nuvem de Partículas, Algoritmo Genético, Complexidade Computacional

Abstract—In this work, it is evaluated the application of the Comprehensive Learning Particle Swarm Optimization (CLPSO) to the adaptive IIR (Infinite Impulsive Response) filtering problem. It is also tested the genetic algorithm (GA), the original and the modified particle swarm optimization (PSO and MPSO). The analysis is carried out considering the performance versus complexity trade-off, where the computational complexity is defined as the number of operations needed for the convergence of the algorithm. The analysis evolves unimodal and multimodal error surfaces, error in the measures of the desired signal and the robustness of the algorithms to variations in the size of population.

Keywords—IIR, Heuristic Algorithm, Particle Swarm Optimization, Genetic algorithm, Computational Complexity.

I. INTRODUÇÃO

Devido à possibilidade de melhor desempenho e menor complexidade em relação à filtragem de resposta impulsiva finita (FIR - *Finite Impulsive Response*), pesquisadores buscam atingir os mesmos resultados para generalização do problema através da utilização de filtragem com resposta impulsiva infinita (IIR - *Infinite Impulsive Response*) [1]. O principal obstáculo a ser superado na filtragem adaptativa IIR é a existência de superfícies de erro multimodais em relação aos coeficientes dos filtros [2]. Assim, algoritmos do tipo gradiente [1], comumente utilizados para o problema FIR, podem não ser eficientes para aplicações de filtragem IIR, pois não garantem o encontro do mínimo global, podendo o algoritmo ficar “preso” em algum mínimo local da superfície de erro, como é o caso do algoritmo LMS (*Least Mean Square*).

^{1,2}Estudante de mestrado do Depto Eng. Elétrica da UEL. E-mail: daguileo@yahoo.com.br

³Professor Adjunto do DEEL - Departamento Eng. Elétrica da UEL - PR, Brazil. E-mail: taufik@uel.br.

⁴Professor Titular do PTC - Depto de Engenharia de Telecomunicações e Controle da EPUSP. E-mail: pj@lcs.poli.usp.br

Recentemente, os algoritmos heurísticos têm sido utilizados e se mostrado muito eficazes para solucionar problemas multimodais. Dentre os métodos aplicados à filtragem IIR, destacam-se o algoritmo *simulated annealing* [3], o algoritmo genético e outras variantes evolucionárias [2], [4], [5], e também algoritmos de otimização baseados em nuvem de partículas [6]–[8].

Neste trabalho, utiliza-se o compromisso desempenho *versus* complexidade na avaliação dos algoritmos heurísticos; A avaliação da complexidade (número de operações necessárias para se atingir a convergência) cria um cenário mais amplo e efetivo da viabilidade das técnicas heurísticas aplicadas ao problema da identificação de sistema.

II. FILTRAGEM ADAPTATIVA IIR

A filtragem IIR caracteriza-se por uma saída dependente dos valores presentes e passados das amostras dos sinais de entrada, bem como das amostras passadas dos sinais de saída:

$$y(n) = \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)y(n-i) \quad (1)$$

onde $a_i(n)$ e $b_i(n)$ são os coeficientes ajustáveis do modelo, $u(n)$ e $y(n)$ são a entrada e a saída do processo para a amostra n , respectivamente. De (1), a função de transferência de um filtro IIR, no domínio z , é simplesmente:

$$H(z) = \frac{\mathbf{A}(z)}{1 - \mathbf{B}(z)} \quad (2)$$

onde $\mathbf{A}(z) = \sum_{i=0}^M a_i z^{-i}$ e $\mathbf{B}(z) = \sum_{i=1}^N b_i z^{-i}$.

Para o problema de identificação de sistemas com filtragem adaptativa, tem-se um sistema IIR desconhecido que deve ser reproduzido através de um filtro adaptativo, que deverá resultar em respostas temporal e espectral semelhantes às respectivas respostas do sistema a ser identificado. Dois métodos podem ser adotados para a estimativa dos coeficientes do filtro adaptativo que irá representar o sistema desconhecido: o método do erro de saída (OEM - *Output Error Method*) ou o método da equação de erro (EEM - *Equation Error Method*).

A. Método do Erro de Saída (OEM)

Esse método consiste na aplicação simultânea do sinal de entrada $u(n)$ no sistema desconhecido e no filtro adaptativo, produzindo o sinal desejado $d(n)$ e o sinal de saída do filtro $y(n)$, respectivamente, fig. 1. Para sinais reais, o ajuste dos coeficientes é realizado via minimização da função custo:

$$J(n) = \frac{1}{2}e^2(n), \quad \text{e o sinal de erro: } e(n) = d(n) - y(n) \quad (3)$$

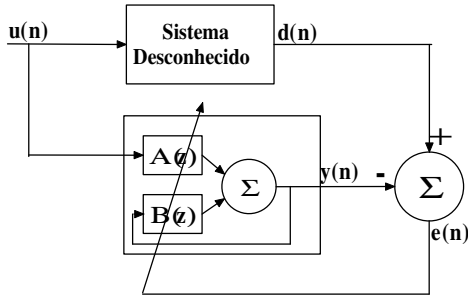


Fig. 1. Diagrama de blocos para identificação de sistema via método OEM.

Dois problemas são inerentes a esse método e causam limitações práticas: a) a superfície de erro se torna multimodal, introduzindo dificuldade adicional na convergência dos algoritmos heurísticos; b) não há garantia de que os pólos se encontrem no interior do círculo unitário, o que gera a possibilidade de filtros instáveis.

B. Método da Equação de Erro (EEM)

As amostras de saída realimentadas do filtro adaptativo são substituídas pelas amostras do sinal de saída, figura 2:

$$y'(n) = \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)d(n-i) \quad (4)$$

De forma semelhante ao método de equação de erro, o erro é:

$$e'(n) = d(n) - y'(n) \quad (5)$$

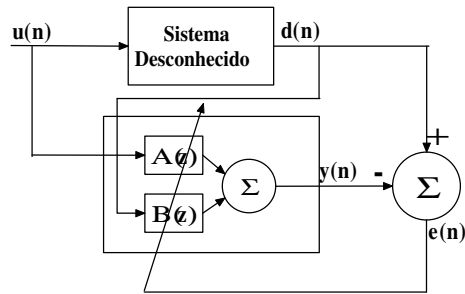


Fig. 2. Diagrama de blocos para identificação de sistema via método EEM.

Devido ao erro nesse método ser linear em relação aos coeficientes, o numerador e o denominador da equação (2) podem ser tratados como dois filtros FIR, eliminando instabilidades e criando superfícies de erro unimodais. Todavia, filtros de ordem reduzida em relação ao do sistema desconhecido e ruídos de medição do sistema podem gerar uma solução que está deslocada em relação à solução ótima.

III. ALGORITMOS HEURÍSTICOS PARA O PROBLEMA DO RECONHECIMENTO DE SISTEMAS

Algoritmos heurísticos são métodos de otimização baseados na aproximação progressiva, que visam a utilização de informação já obtida para direcionar a busca em regiões do universo que sejam mais propensas a conterem a solução para o problema. Os métodos heurísticos se diferenciam basicamente pelo modo como os indivíduos são modificados para as gerações futuras e a utilização da informação já adquirida

durante a busca [9]. Alguns conceitos importantes para a compreensão dos algoritmos heurísticos aplicados ao problema de filtragem adaptativa IIR são definidos a seguir:

- Indivíduo/partícula: vetor candidato contendo os coeficientes, pertencente ao universo de possíveis soluções do problema. O i -ésimo indivíduo/partícula é dado por:

$$\mathbf{x}_i[t] = [a_{i0}[t] \ a_{i1}[t] \ \dots \ a_{iM}[t] \ b_{i1}[t] \ \dots \ b_{iN}[t]]^T \quad (6)$$

contendo os coeficientes estimados do filtro no instante de tempo t ;

- População (p): conjunto de indivíduos/partículas presentes em uma geração/instante de tempo. Para definir o número de indivíduos/partículas da população inicial, são adotados métodos empíricos.
- Geração/Instante de tempo (t): trata-se da iteração em que se encontra o processo de busca.
- Função Objetivo/Custo: utilizada para avaliar o desempenho de cada cromossomo/partícula. Nas simulações, utilizou-se uma janela contendo $L_w = 100$ amostras, e a função custo foi definida como sendo o erro quadrático médio (MSE – Mean Squared Error) sobre essa janela.

$$MSE = \frac{1}{L_w} \sum_{n=1}^{L_w} [d(n) - y(n)]^2 \quad (7)$$

A. Algoritmo Heurístico PSO

Observando o comportamento dos pássaros, alguns pesquisadores notaram o favorecimento de todo o bando a partir do seu comportamento em conjunto. Baseando-se nessa observação, Kennedy e Eberhart desenvolveram em 1995 [10] o algoritmo de otimização por nuvem de partículas (PSO).

O algoritmo é baseado no movimento de um grupo de partículas, distribuídas aleatoriamente no espaço (universo de busca), cada uma possuindo uma posição e uma velocidade próprias. Essa velocidade é responsável por inserir um movimento na partícula, alterando a sua posição no espaço em busca de um melhor desempenho. A posição da i -ésima partícula é representada conforme a equação (6), e o movimento é inserido na partícula através da equação:

$$\mathbf{x}_i[t+1] = \mathbf{x}_i[t] + \mathbf{v}_i[t] \quad (8)$$

onde $\mathbf{v}_i[t]$ é a velocidade da i -ésima partícula no instante de tempo t .

A interação entre as partículas é inserida no PSO através do cálculo da velocidade. O conhecimento da melhor posição global do conjunto até o instante t , denominada $\mathbf{x}_g^{best}[t]$, o conhecimento da melhor posição encontrada por cada partícula individualmente $\mathbf{x}_i^{best}[t]$ e a velocidade prévia da própria partícula compõem o cálculo da velocidade:

$$\mathbf{v}_i[t+1] = \omega \cdot \mathbf{v}_i[t] + \phi_1 \cdot \mathbf{U}_{i1}[t](\mathbf{x}_i^{best}[t] - \mathbf{x}_i[t]) + \phi_2 \cdot \mathbf{U}_{i2}[t](\mathbf{x}_g^{best}[t] - \mathbf{x}_i[t]) \quad (9)$$

onde ω é o peso inercial, ϕ_1 e ϕ_2 são os coeficientes de aceleração referentes à melhor posição individual e à melhor posição global, respectivamente, e $\mathbf{U}_{i1}[t]$ e $\mathbf{U}_{i2}[t]$ são matrizes diagonais de dimensão $M + N + 1$ com elementos sendo

números aleatórios com distribuição uniforme $\mathcal{U}(0, 1)$, gerados para a i -ésima partícula.

Os parâmetros de entrada do algoritmo PSO, ω , ϕ_1 e ϕ_2 devem ser adaptados e otimizados para o problema de filtragem IIR, de modo que o algoritmo seja capaz de atingir a convergência, com a menor complexidade computacional (número de gerações) possível. Os fatores ω e ϕ_1 são responsáveis por inserir diversificação à busca, devido ao direcionamento independente de cada partícula, enquanto ϕ_2 é responsável pelo mecanismo de intensificação, através do direcionamento de todas as partículas para uma mesma região. Os passos do PSO estão sintetizados no Algoritmo 1.

Algoritmo 1 PSO - IIR

1. Geração da população inicial com p partículas distribuídas aleatoriamente no espaço dimen. $M + N + 1$.
2. Para $t = 1, 2, \dots$ faça:
 - 2.1. Avalia-se cada partícula $\mathbf{x}_i[t]$ através da função custo, equação (7).
 - 2.2. Caso exista melhoria de desempenho, atualiza-se $\mathbf{x}_g^{best}[t]$ e $\mathbf{x}_i^{best}[t]$. Calcula-se o peso de inércia adaptativo.
 - 2.3. Calcula-se a velocidade $\mathbf{v}_i[t]$ de cada partícula, eq. (9).
 - 2.4. Atualiza-se cada coeficiente de cada partícula, eq. (8).

fim

B. Algoritmo Heurístico PSO Modificado (MPSO)

Recentemente desenvolvido por Krusienski [6]–[8], o algoritmo PSO modificado (MPSO – *Modified Particle Swarm Optimization*) foi apresentado como uma alternativa heurística mais eficiente para a filtragem adaptativa IIR, o que o torna uma importante referência nas análises comparativas.

A sua diferenciação encontra-se na exploração de técnicas de intensificação capazes de superar as limitações do algoritmo PSO original. Em relação ao PSO, o MPSO acrescenta três etapas:

- Mutações: Uma pequena parcela dos parâmetros são afetados pela mutação, que apresenta variância decrescente no decorrer do tempo, visando a intensificação e conseqüente convergência ao final da busca;
- Reorganização ao redor de $\mathbf{x}_g^{best}[t]$: Caso seja encontrado um novo $\mathbf{x}_g^{best}[t]$, uma pequena parte das partículas é deslocada aleatoriamente para a vizinhança dessa nova posição, de modo a intensificar a busca sobre essa região.
- Coeficiente de inércia adaptativo: Cada partícula possui um peso inercial que ajusta sua velocidade na direção em que se obteve melhor desempenho.

Os passos para o MPSO são mostrados no Algoritmo 2.

C. Algoritmo Heurístico CLPSO

Publicado por Liang *et al* [12], em 2006, o algoritmo PSO de aprendizado compreensivo (CLPSO) foi proposto como uma alternativa eficiente de solução para problemas multimodais. Diferentemente do MPSO, o CLPSO não acrescenta novos operadores, mas busca aumentar a diversificação do

Algoritmo 2 MPSO - IIR

1. Geração da população inicial com p partículas distribuídas aleatoriamente no espaço dimen. $M + N + 1$.
2. Para $t = 1, 2, \dots$ faça:
 - 2.1. Avalia-se cada partícula $\mathbf{x}_i[t]$ através da função custo, eq. (7).
 - 2.2. Caso exista melhoria de desempenho, atualiza-se $\mathbf{x}_g^{best}[t]$ e $\mathbf{x}_i^{best}[t]$. Calcula-se o peso de inércia adaptativo.
 - 2.3. Caso tenha sido encontrado um novo $\mathbf{x}_g^{best}[t]$, desloca-se algumas partículas para a nova vizinhança.
 - 2.4. Calcula-se a velocidade $\mathbf{v}_i[t]$ de cada partícula através da eq. (9) e atualiza-se a sua posição através da eq. (8).
 - 2.5. Insere-se a mutação nas partículas.

fim

PSO através de alterações no seu próprio princípio básico de funcionamento, o cálculo da velocidade:

$$\mathbf{v}_i[t + 1] = \omega \cdot \mathbf{v}_i[t] + \phi_c \cdot \mathbf{U}_{i_1}[t](\mathbf{x}_j^{best}[t] - \mathbf{x}_i[t]) \quad (10)$$

onde ϕ_c é o coeficiente de aceleração único e $\mathbf{x}_j^{best}[t]$ é a melhor posição da j -ésima partícula, que será adotada pela i -ésima partícula. A troca da melhor posição de uma partícula no cálculo da velocidade é realizada através de uma probabilidade de troca; a seleção da posição $\mathbf{x}_j^{best}[t]$, a ser utilizada na atualização da velocidade $\mathbf{v}_i[t + 1]$, é definida através de uma escolha aleatória na população. Os passos para implementação do algoritmo CLPSO são indicados no Algoritmo 3.

Algoritmo 3 CLPSO - IIR

1. Geração da população inicial com p partículas distribuídas aleatoriamente no espaço dimen. $M + N + 1$.
2. Para $t = 1, 2, \dots$ faça:
 - 2.1. Avalia-se cada partícula $\mathbf{x}_i[t]$ através da função custo, eq. (7).
 - 2.2. Caso exista melhoria de desempenho para a i -ésima partícula, armazena-se a posição atual em $\mathbf{x}_i^{best}[t]$.
 - 2.3. Calcula-se a velocidade $\mathbf{v}_i[t]$ de cada partícula via eq. (10).
 - 2.4. Atualiza-se cada coeficiente de cada partícula, eq. (8).

fim

D. Algoritmo Heurístico GA

Para o problema de filtragem, adotou-se um modelo de GA real (contínuo), possuindo os operadores mutação, seleção e *crossover*. É adotada a estratégia de elitismo, através da seleção apenas dos p_s melhores indivíduos para reprodução. Assim como na genética, o operador *crossover* é responsável por criar indivíduos com características mescladas entre os seus predecessores. O *crossover* é realizado através de uma

ponderação entre os coeficientes de dois indivíduos selecionados para gerar o descendente:

$$\mathbf{x}_i[t+1] = \mathbf{R}\mathbf{x}_{k_1}[t] + (\mathbf{I} - \mathbf{R})\mathbf{x}_{k_2}[t], \quad i = 1 \rightarrow p \quad (11)$$

onde \mathbf{R} é uma matriz diagonal, sendo seus elementos números aleatórios com distribuição uniforme $\mathcal{U}(0,1)$ gerados para cada indivíduo; $\mathbf{x}_{k_1}[t]$ e $\mathbf{x}_{k_2}[t]$ são selecionados entre os p_s indivíduos de forma aleatória.

A mutação dos novos indivíduos é inserida através de uma distribuição $\mathcal{N}(0, \sigma^2)$, cuja variância σ^2 decresce a cada nova geração, tendo em vista a convergência ao final da busca. Os passos para o GA estão indicados no Algoritmo 4.

Algoritmo 4 GA - IIR

1. Geração da população inicial com p indivíduos distribuídos aleatoriamente no espaço dimen. $M + N + 1$.
2. Para $t = 1, 2, \dots$ faça
 - 2.1. Avalia-se cada indivíduo através da função custo, eq. (7).
 - 2.2. Seleção dos melhores indivíduos para produção da próxima geração.
 - 2.3. Aplicação do operador *crossover* para obtenção de novos indivíduos e em seguida a mutação.
 - 2.4. Seleção dos p melhores indivíduos (entre genitores e descendentes) para avançar à próxima geração.

fim

IV. RESULTADOS

Foram identificados vários sistemas IIR com diferentes características em relação à curva de erro, ao número de coeficientes a serem estimados e à presença de ruído no sinal desejado. Os quatro algoritmos se mostraram muito sensíveis aos seus parâmetros de entrada e ao sistema IIR avaliado. Foram adotados parâmetros fixos, obtidos através do desempenho médio entre os diferentes sistemas.

PSO: $\omega = 0,5$ e decrescente, $\phi_1 = 1,2$ e $\phi_2 = 1,2$;

MPSO: $\omega = 0,5$ e decrescente, $\phi_1 = 1,2$, $\phi_2 = 1,2$, probabilidade de um coeficiente sofrer mutação igual a 0,125, com variância decrescente; probabilidade das partículas serem rearranjadas ao redor de $\mathbf{x}_g^{best}[t]$ igual a 0,05;

CLPSO: $\omega = 0,5$ e decrescente, $\phi_c = 1,8$ e probabilidade de troca de $\mathbf{x}_i^{best}[t]$ igual a 0,8.

GA: 6 indivíduos selecionados para geração dos descendentes, sendo cada novo indivíduo gerado a partir do *crossover* entre dois indivíduos escolhidos aleatoriamente no grupo selecionado; mutação decrescente no decorrer da busca.

Os parâmetros de entrada acima foram otimizados por tentativa e erro, de forma não exaustiva, porém considerando um conjunto de valores relativamente amplo.

Os resultados apresentados a seguir são relativos a um número mínimo de 100 realizações. Os gráficos de MSE são representativos para as realizações em que o algoritmo convergiu para o mínimo global; esta seção também analisa a percentagem das realizações em que os algoritmos não convergiram. Para todos os algoritmos, adotou-se inicialização aleatória para as partículas, com distribuição $\mathcal{U}(-2, 2)_{M+N+1}$.

A. Sistema I: Filtro Unimodal, $SNR = \infty$

O primeiro sistema avaliado é um filtro unimodal e sem ruído [4]. Trata-se do caso mais simples, sendo a entrada um ruído Gaussiano branco (WGN – *White Gaussian Noise*) e o filtro adaptativo de baixa ordem. A função de transferência para o sistema desconhecido, $H_p(z)$, e para o filtro adaptativo, $H_a(z)$ são dadas por:

$$H_p(z) = \frac{1}{1 - 1,2z^{-1} + 0,6z^{-2}}; \quad H_a(z) = \frac{b_0}{1 + a_1z^{-1} + a_2z^{-2}}$$

A superfície de erro não possui mínimos locais. Utilizando entrada $\mathcal{N}(0,1)$, o mínimo global possui erro quadrático médio igual a zero. Por se tratar de um filtro unimodal, o algoritmo LMS é suficiente para solucionar o problema. No caso dos algoritmos heurísticos, a sua avaliação mostra a capacidade de intensificação de busca do algoritmo e o ajuste fino dos coeficientes do filtro. Verifique-se que mesmo para uma população $p = 20$, os algoritmos ainda convergem para o provável fundo de escala, com exceção do PSO que converge previamente para um patamar de ≈ -30 dB.

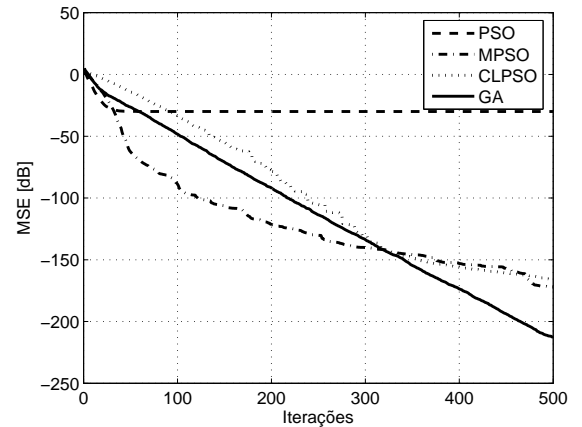


Fig. 3. Curva de convergência dos algoritmos heurísticos para o sistema com superfície de erro unimodal, Sistema I, considerando $p = 20$.

B. Sistema II: Filtro Multimodal, $SNR = \infty$

Este sistema é um filtro multimodal e sem ruído [4]. Trata-se de um caso mais complexo que o anterior, sendo o filtro adaptativo de ordem reduzida em relação ao sistema desconhecido. A função de transferência do sistema a ser identificado e do filtro adaptativo, respectivamente, são:

$$H_p(z) = \frac{1}{(1 - 0,6z^{-1})^3}, \quad H_a(z) = \frac{b_0}{1 + a_1z^{-1} + a_2z^{-2}} \quad (12)$$

sendo a entrada um ruído colorido obtido através da filtragem do WGN pelo filtro FIR com função de transferência $H_c(z) = (1 - 0,6z^{-1})^2 (1 + 0,6z^{-1})^2$

Uma vez que a superfície de erro para este caso apresenta um mínimo local, a utilização desse filtro visa a avaliação do desempenho dos algoritmos perante um sistema multimodal, considerando o desempenho obtido e ainda a capacidade de se atingir o mínimo global. A figura 4 mostra as curvas de convergência dos algoritmos para as realizações em que os algoritmos convergiram para o mínimo global, considerando $p = 50$. Observe que os algoritmos convergem para o mesmo ponto, com o PSO sendo mais rápido.

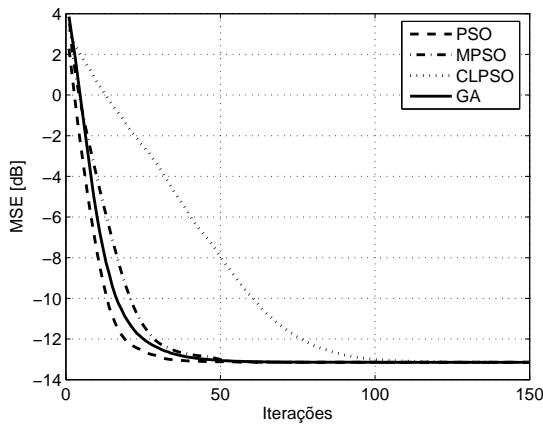


Fig. 4. Curva de convergência, Sistema II; considerando $p = 50$ e apenas as realizações em que os algoritmos convergiram para o mínimo global.

No entanto, a capacidade de encontrar o mínimo global da curva de erro não é idêntica para todos os algoritmos. A figura 5 mostra a curva de convergência considerando todas as realizações e mesma população. Observe que o algoritmo CLPSO é o único a convergir em todas as realizações para o mínimo MSE. A figura 6 explica o melhor desempenho global do algoritmo CLPSO, através da percentagem de realizações em que os algoritmos não convergiram, em função da população utilizada na busca. O CLPSO se mostrou mais eficiente e robusto quanto à variação desse parâmetro.

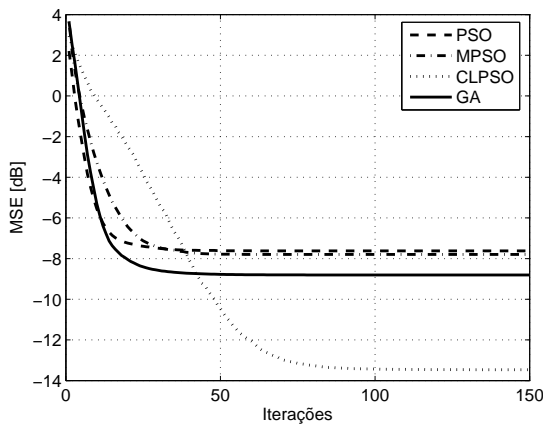


Fig. 5. Curva de convergência dos algoritmos do Sistema II, considerando $p = 50$ e todas as realizações.

C. Sistema III: Filtro Multimodal, SNR = 20dB

Este sistema é um filtro de baixa ordem, multimodal e SNR=20dB [4]. Trata-se de um caso mais simples que o anterior, sendo o filtro adaptativo de mesma ordem em relação ao sistema desconhecido. A função de transferência do sistema desconhecido e a do filtro adaptativo são:

$$H_p(z) = \frac{1}{(1 - 0,7z^{-1})^2}, \quad H_a(z) = \frac{b_0}{1 + a_1z^{-1} + a_2z^{-2}} \quad (13)$$

sendo a entrada um ruído colorido obtido através da filtragem do WGN por um filtro FIR com função transferência $H_c(z) = (1 - 0,7z^{-1})^2 (1 + 0,7z^{-1})^2$.

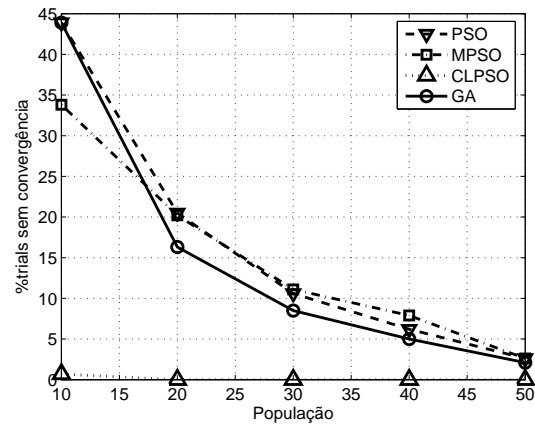


Fig. 6. Percentagem de realizações em que os algoritmos não atingiram o mínimo global para o sistema da eq. (12).

A superfície de erro novamente apresenta mínimo local. Há porém a presença de ruído no sinal desejado, o que irá limitar o MSE em -20 dB, avaliando a robustez dos algoritmos face aos ruídos de medição. Levando-se em conta todas as realizações, a tabela I mostra a percentagem de realizações em que os algoritmos não convergiram para o mínimo global. Novamente, ressalte-se a robustez do algoritmo CLPSO que convergiu para o mínimo global em 100% das realizações, para ambas as populações avaliadas.

TABELA I

PERCENTUAL DE REALIZAÇÕES EM QUE OS ALGORITMOS NÃO CONVERGIRAM PARA O MÍNIMO GLOBAL, SISTEMA III.

Algoritmo	$p = 20$	$p = 50$
PSO	11,0%	1,2%
MPSO	9,4%	2,0%
CLPSO	0,0%	0,0%
GA	6,4%	0,8%

D. Sistema IV: Filtro Unimodal, SNR = 80dB

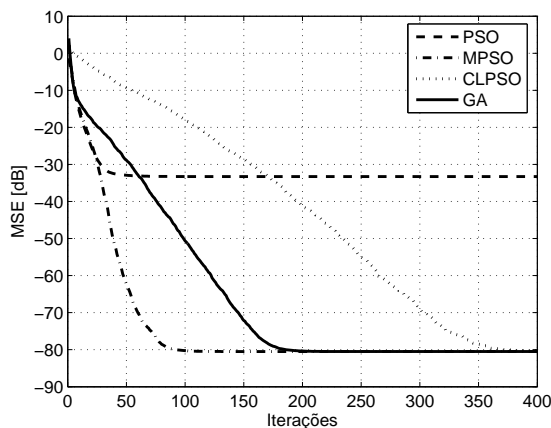
O próximo sistema a ser avaliado é um filtro unimodal e com ruído [6]. A entrada é um WGN e o filtro adaptativo escolhido para identificar o sistema possui ordem suficiente. A função de transferência para o sistema desconhecido, $H_p(z)$, e para o filtro adaptativo, $H_a(z)$, são, respectivamente:

$$\frac{1,25z^{-1} - 0,25z^{-2}}{1 - 0,3z^{-1} + 0,4z^{-2}}; \quad e \quad \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

Trata-se de um problema onde o número de coeficientes a ser estimado é cinco (a_1, a_2, b_0, b_1, b_2). A superfície de erro não possui mínimos locais. Utilizando entrada $\mathcal{N}(0, 1)$, o mínimo global é limitado em -80 dB devido ao ruído. A utilização desse filtro tem o objetivo de verificar o efeito do aumento do número de coeficientes sobre a capacidade de busca fina dos algoritmos. A figura 7 mostra que apenas o PSO não foi capaz de atingir a convergência em -80 dB, demonstrando novamente problemas de intensificação e estagnação.

V. COMPLEXIDADE COMPUTACIONAL

A complexidade computacional necessária para a convergência do algoritmo heurístico é representada aqui através do número de operações (No), dentre elas, comparações,

Fig. 7. Curva de convergência dos algoritmos para o Sistema IV, com $p = 40$;

multiplicações e gerações de número aleatórios. Neste trabalho adota-se a simplificação de custo computacional idêntico para todas as operações consideradas.

O número de operações dos algoritmos heurísticos é função das seguintes variáveis: número de amostras (L_w), número de coeficientes a se estimar (c), população do algoritmo heurístico (p), número de iterações (g) necessárias à convergência do algoritmo, percentual de partículas que são deslocadas sobre a melhor posição global (p_{rnd}) no algoritmo MPSO e o número de indivíduos selecionados (p_s) no GA.

Para a avaliação de um único vetor candidato, o cálculo da função custo envolve o cômputo de $L_w \cdot (c + 1) + 1$ operações, sendo semelhante a todos os algoritmos. A tabela II expressa o número de operações computacionais que cada algoritmo deve realizar para atingir a convergência. Note-se que a primeira parcela das respectivas expressões para a complexidade corresponde ao maior número de operações do problema, referente principalmente ao cálculo da função custo.

TABELA II
COMPLEXIDADE DOS ALGORITMOS HEURÍSTICOS.

Algoritmo	Número de Operações, N_o
PSO	$g [pc (L_w + 5) + p (L_w + 3) + 1] + 1$
MPSO	$g [pc (L_w + 8 + p_{rnd}) + p (L_w + 6 + p_{rnd}) + 6]$
CLPSO	$g [pc (L_w + 3) + p (L_w + 7) + 1] + 1$
GA	$g [pc (L_w + 3) + (3c + 6) (p - p_s) + p (L_w + 2) + 6]$

Realizou-se uma rápida análise para a verificação do efeito da janela de amostragem utilizada no cálculo da função custo, eq. (7), sobre o desempenho dos algoritmos, buscando uma minimização de L_w , mantendo-se as características necessárias à identificação do sistema. Constatou-se que para valores de $L_w > 50$, a entrada do sistema é suficiente para a obtenção da resposta do sistema desconhecido. Valores inferiores não garantem a convergência dos algoritmos, pois para os casos analisados não representam as características espectrais necessárias à representação do sistema.

Assim, a partir da tabela II avaliou-se o desempenho dos algoritmos em função do número de operações para três dos sistemas discutidos anteriormente. A tabela III indica

o número de operações necessárias para que os algoritmos atinjam o MSE mínimo. Adotou-se uma população específica para cada algoritmo tal que fosse garantido uma probabilidade de sucesso na convergência superior a 95% das buscas.

TABELA III
NÚMERO DE OPERAÇÕES N_o : CONVERGÊNCIA EM 95% DAS BUSCAS

Algoritmo	Sistema II	Sistema. III	Sistema IV
PSO	$4, 4.10^5$	$1, 6.10^5$	$9, 8.10^5$
MPSO	$5, 7.10^5$	$2, 0.10^5$	$8, 8.10^5$
CLPSO	$3, 0.10^5$	$1, 5.10^5$	$1, 7.10^6$
GA	$5, 6.10^5$	$3, 8.10^5$	$1, 3.10^6$

VI. CONCLUSÕES

Para os sistemas avaliados, o algoritmo CLPSO foi o mais eficiente em sistemas multimodais, necessitando de uma menor população e apresentando ainda uma maior probabilidade de convergência para o mínimo global em relação às outras técnicas. O algoritmo MPSO foi superior para buscas em superfícies unimodais. O GA apresentou bom desempenho em ambos os sistemas, porém com uma maior complexidade. O algoritmo PSO também apresentou bons resultados, porém sendo muito sensível ao tamanho da população.

A análise da complexidade mostrou que os algoritmos possuem número de operações por iteração muito próximos, com $N_o^{CLPSO} < N_o^{PSO} < N_o^{GA} < N_o^{MPSO}$, uma vez que o cálculo da função custo é a parcela mais onerosa. Um número de amostras para a janela de cálculo da função custo igual a 50 mostrou-se suficiente para garantir uma representação fiel dos quatro sistemas analisados.

O algoritmo CLPSO mostrou-se mais robusto em relação ao tamanho da população, enquanto o PSO é o mais sensível.

REFERÊNCIAS

- [1] S. Haykin, "Adaptive Filter Theory", 4ª edição, Prentice Hall, 2002.
- [2] S. G. Ng, S. H. Leung, C. Y. Chung, A. Luk e W. H. Lau, "The Genetic Search Approach - A New Learning Algorithm for Adaptive IIR Filter", *IEEE Signal Processing Magazine*, november, 1996, p. 38-46.
- [3] N. Benvenuto and M. Marchesi (1989). "Digital filters design by simulated annealing", *IEEE Transactions on Circuits and Systems*, 36(3):459-460.
- [4] H. Fan e W. K. Jenkins, "A New Adaptive IIR Filter", *IEEE Transactions on Circuits and Systems*, CAS-33(10):939-947, outubro de 1986.
- [5] M. S. White e S. J. Flockton, "Chapter in Evolutionary Algorithms in Engineering Applications", Editors: D. Dasgupta and Z. Michalewicz, Springer Verlag, 1997.
- [6] D. J. Krusienski, "Enhanced Structured Stochastic Global Optimization Algorithms for IIR and Nonlinear Adaptive Filtering", *Doctor Thesis*, Pennsylvania State University, 2004;
- [7] D. J. Krusienski, W. K. Jenkins, "Design and performance of adaptive systems based on structured stochastic optimization strategies", *IEEE Circuits and Systems Magazine*, Volume 5, Issue 1, 2005, p. 8 - 20
- [8] D. J. Krusienski, W. K. Jenkins, "A modified particle swarm optimization algorithm for adaptive filtering", *ISCAS 2006 - IEEE International Symposium on Circuits and Systems*, 21-24 May, 2006, p. 4 pp.
- [9] M. C. Goldberg; H. P. L. Luna *Otimização Combinatória e Programação Linear*. Rio de Janeiro, Campus, 2000.
- [10] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", *IEEE International Conference on Neural Networks*, p. 1942-1948, 1995.
- [11] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.
- [12] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions", *IEEE Transactions on Evolutionary Computation*, vol. 10, n. 3, p. 281-295, June 2006.