

Abordagens bio-inspiradas de otimização estocástica aplicadas em projeto de filtro para um processo de dinâmica inversa

Marco Antonio Reichert Boaretto
Engenharia de Controle e
Automação, Pontifícia Universidade
Católica do Paraná (PUCPR),
Curitiba, PR, Brasil
marco.boaretto@hotmail.com

Marcos Cesar Gritti
Engenharia de Controle e
Automação, Pontifícia Universidade
Católica do Paraná (PUCPR),
Curitiba, PR, Brasil
cesargritti@gmail.com

Leandro dos Santos Coelho
Programa de Pós-Graduação em
Engenharia de Produção e Sistemas
(PPGEPS), Escola Politécnica
Pontifícia Universidade Católica do
Paraná (PUCPR) e Programa de Pós-
Graduação em Engenharia Elétrica
(PPGEE), Universidade Federal do
Paraná (UFPR),
Curitiba, PR, Brasil
lscelho2009@gmail.com

Resumo—Neste trabalho, propomos a solução de um problema de design de filtro aplicado à um processo de dinâmica inversa, análise de caminhada, usando três técnicas de otimização conhecidas como evolução diferencial, otimizador dos lobos cinzentos e algoritmo dos morcegos. O problema consiste em encontrar as melhores frequências de corte para um filtro Butterworth passa-baixa que resultará em um menor *fitness*, neste caso será a combinação dos momentos intersegmentais das juntas de um modelo musculoesquelético em 2-D. Todas as três técnicas alcançaram uma performance eficaz comparadas com os valores de referência, provando que todas as três funcionam muito bem na otimização de filtros com características de amplitude não convencionais.

Palavras chave—Evolução diferencial; Otimizador dos lobos cinzentos; Algoritmo dos morcegos; Otimização de filtros; Filtro IIR; Dinâmica Inversa; Análise de caminhada;

I. INTRODUÇÃO

Filtros digitais vêm sendo aplicados em uma grande variedade de aplicações como em comunicações, indústria química, reatores nucleares e sistemas de controle. Porém a performance do filtro depende muito de como ele foi projetado, cada filtro é projetado de acordo com sua aplicação para atender determinados requisitos e funcionar com a robustez desejada, entretanto nem sempre é um trabalho trivial ao utilizar técnicas comuns de aproximação de filtros. Neste tipo de situação a solução recomendada é fazer o uso de técnicas mais avançadas que conseguem lidar com as dificuldades em projetar um filtro para características de amplitude não convencionais [1].

O projeto de um filtro Butterworth passa-baixa para um processo de dinâmica inversa, pode apresentar certas dificuldades como por exemplo, certas frequências de corte determinadas para o filtro passa-baixa podem levar a perda de componentes de alta frequência causados por impactos de movimento [2].

Neste trabalho com a finalidade de obter um filtro passa-baixa com um desempenho ótimo para o processo de dinâmica

inversa, serão utilizadas as técnicas de otimização, evolução diferencial (ED)[3][4] que provou ser uma técnica muito bem-sucedida na aplicação de projeto de filtros digitais [1][5-8]. E duas técnicas recentes o *Grey Wolf Optimizer* (GWO) [9-12] e o *Bat Algorithm* (BA) [13-16].

O restante do documento está dividido na seguinte forma, na seção II é apresentado o algoritmo da ED suas características e etapas. Na seção III o algoritmo BA é descrito junto com suas principais características e regras que definem o método de otimização inspirado nos morcegos. Na seção IV é mostrado o algoritmo GWO. Na seção V, é detalhado o processo na qual o filtro será aplicado. Na seção VI são mostrados os resultados obtidos com os testes, seguidos pela conclusão do trabalho na seção VII.

II. O ALGORITMO DA ED

O algoritmo de ED proposto por Price e Storn em 1995 [3], originalmente para resolver o problema de Chebychev, mostrou ser uma simples, porém eficaz abordagem de otimização para resolver uma variedade de problemas de referência como também muitas aplicações do mundo real [17]. Por ser um algoritmo baseado em busca de população consegue resolver problemas de mínimo local e convergência prematura de uma maneira fácil e rápida em contraste com algoritmos baseados em solução simples.

O algoritmo de ED é derivado dos Algoritmos Genéticos (AG) [18], portanto certa semelhança é observada ao comparar a estrutura de ambos algoritmos, entretanto a ED possui algumas peculiaridades que a tornam uma técnica de otimização mais poderosa que um simples AG no problema de design de filtros [7].

Baseado em [4] o algoritmo de ED pode ser definido pelas seguintes etapas.

A. Geração da população inicial

Nesta etapa são determinados os limites máximo $b_{j,U}$ e mínimo $b_{j,L}$ para gerar o domínio do espaço de busca para o cálculo do vetor inicial $x_{j,i,0}$ (1). Em contraste com outros algoritmos evolutivos a ED não usa funções de distribuição de probabilidade com intuito de adicionar variações na população, ao invés utiliza a diferença de vetores aleatoriamente selecionados como fonte de variações aleatórias. Os parâmetros de controle como fator de perturbação e probabilidade de cruzamento são definidos na inicialização.

$$x_{j,i,0} = rand_j(0,1)*(b_{j,U} - b_{j,L}) + b_{j,L} \quad (1)$$

onde, $rand_j$ é a função geradora de números aleatórios uniformemente distribuídos entre 0 e 1. O subscrito j indica que a cada parâmetro é gerado um valor aleatório novo.

B. Mutação

O processo de mutação (2) é a diferença entre dois vetores escolhidos aleatoriamente, que passam a ser multiplicados à um fator de perturbação, que controla a taxa de evolução populacional. Um terceiro vetor, também selecionado aleatoriamente, é adicionado a equação originando o vetor mutante $v_{i,g}$. Uma possível razão para ED funcionar tão bem é que a mutação é acionada por diferenças entre o valor dos parâmetros de membros da população contemporânea [19]. Price e Storn também propuseram diferentes técnicas de mutação chamadas de estratégias de mutação, as mais utilizadas pela literatura são [17], “DE/rand/1”, “DE/current-to-best/1” e “DE/best/1”.

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2)$$

onde, F é o fator de perturbação, enquanto $x_{r0,g}$, $x_{r1,g}$ e $x_{r2,g}$ são os vetores selecionados aleatoriamente.

C. Cruzamento

Conhecido também por reprodução, é onde ocorre a reprodução, entre os vetores mutante e alvo para gerar um novo vetor teste formado pela combinação de informação dos dois vetores.

D. Seleção

É na seleção que o vetor alvo e o vetor teste competem de acordo com seu *fitness*, o vetor que tiver o melhor fitness passa para a próxima geração.

Os passos se repetem até um critério de parada ser encontrado. Na Fig. 1, o pseudocódigo da ED.

Pseudocódigo ED
 Definição de $b_{j,U}$ e $b_{j,L}$ e parâmetros de inicialização.
 Geração da população inicial (1).
enquanto critério de parada não atingido.
 para 1 até população máxima (Np).
 Selecionar três indivíduos diferentes r_1 , r_2 e r_3 .
 Mutação (2)
 Cruzamento.
 Seleção dos melhores indivíduos resultantes dos processos anteriores.
 fim para
 fim enquanto

Fig. 1. Pseudocódigo ED.

III. BA

O algoritmo BA proposto por Yang em 2010 [16], se baseia no comportamento de eco localização, ou seja, da percepção de posição através do eco utilizado pelos morcegos para buscar o ótimo [13]. Os morcegos utilizam como método de navegação a emissão de pulsos que se encontram em uma região entre 25-150Khz [14], com esse método os morcegos conseguem identificar obstáculos tão pequenos como um fio de cabelo humano [15]. Certos morcegos insetívoros, Fig. 2, conseguem caçar suas presas e discernir entre os diferentes tipos de insetos sem necessidade haver luz no local.

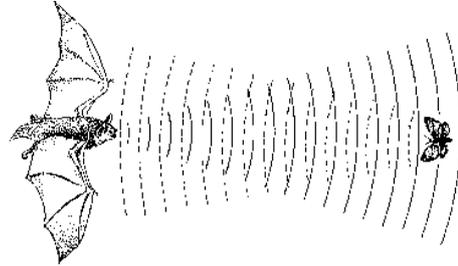


Fig. 2. Sonar.

O algoritmo BA pode ser idealizado pelo seguinte conjunto de regras:

- Todos os morcegos fazem uso da eco localização para percepção de distância, e eles também reconhecem a diferença entre comida/presa e barreiras e obstáculos de alguma maneira mágica.
- Morcegos voam aleatoriamente com uma velocidade v_i em uma posição x_i com uma frequência fixa f_{min} variando o comprimento de onda λ e sonoridade A^0 para encontrar sua presa. Eles automaticamente ajustam o comprimento de onda (ou frequência) de seus pulsos emitidos e ajustam também a taxa de emissão $r \in [0,1]$, dependendo da proximidade do seu alvo.
- Embora a sonoridade possa variar em várias formas, assume-se que a sonoridade varia de um grande (positivo) A^0 para um mínimo valor constante A_{min} .

Para cada morcego i existe uma posição x_i e uma velocidade v_i que são atualizadas a cada iteração gerando os respectivos vetores x_i^t (3) e v_i^t (4).

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

onde, x_i^{t-1} é a posição da iteração anterior.

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \quad (4)$$

onde, v_i^{t-1} é a velocidade da iteração anterior, x^* corresponde a melhor solução global atual, f_i corresponde a frequência que é atualizada a cada iteração pela equação (5).

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (5)$$

onde, f_{max} e f_{min} correspondem respectivamente as frequências máxima e mínima definidas para o problema em interesse, e

β corresponde a um vetor aleatório que segue uma distribuição uniforme de $[0,1]$.

O seguinte pseudocódigo, Fig. 3, mostra um resumo dos passos do BA.

```

Algoritmo dos Morcegos.
Função objetivo  $f(x)$ ,  $x = (x_1, \dots, x_d)T$ .
Inicializar a população de morcegos  $x_i$  ( $i = 1, 2, \dots, n$ ) e  $v_i$ .
Definir frequência de pulsos  $f_i$  em  $x_i$ .
Inicializar taxa de pulso  $r_i$  e sonoridade  $A_i$ .
enquanto ( $t < \text{Max número de iterações}$ )
  Gerar novas soluções por ajuste de frequência, e atualizar
  velocidades e posições/soluções [equações (2) até (4)].
  se ( $\text{rand} < r_i$ )
    Selecionar uma solução dentre as melhores soluções.
    Gerar uma solução local de acordo com a melhor
    solução selecionada.
  fim se
  Gerar uma nova solução por voo aleatório.
  se ( $\text{rand} < A_i$  &  $f(x_i) < f(x^*)$ )
    Aceitar as novas soluções.
    Aumentar  $r_i$  e reduzir  $A_i$ .
  fim se
  Classificar os morcegos e encontrar o melhor  $x^*$  atual
fim enquanto
pós processo resultados e visualizações
  
```

Fig. 3. Pseudocódigo BA.

Como mostra (6), a cada nova iteração t é gerada uma nova solução local x_{novo} , em base da melhor solução já selecionada no processo de seleção.

$$x_{\text{novo}} = x_{\text{velho}} + \varepsilon A^t \quad (6)$$

onde, $\varepsilon \in [-1,1]$ é um número aleatório.

IV. GWO

O algoritmo GWO proposto por Mirjalili et al. em 2014 [9], inspirado no comportamento social de caça dos lobos cinzentos (*Canis Lupus*), é uma nova abordagem no âmbito da inteligência de enxames. Apesar de novo e inexplorado, o algoritmo vem se mostrando promissor em diferentes áreas do conhecimento. O próprio autor o põe em prova em [10] com o objetivo de aperfeiçoar o resultado de treinamento de redes neurais do tipo perceptron multicamadas, do inglês *Multi-Layer Perceptron* (MLP), para problemas de classificação.

Por natureza lobos cinzentos preferem viver em grupos, normalmente contendo, em média, entre 5 a 12 indivíduos (lobos). Cada grupo constitui-se de uma composição hierárquica, onde pode-se classificar quatro subgrupos, nomeados: alfas (α), betas (β), deltas (δ) e ômegas (ω). No GWO, os subgrupos α , β e δ referem-se aos três melhores lobos do grupo, onde o critério de seleção é desempenhado pela aptidão de cada membro, ou seja, pela função objetivo do problema.

Estes têm por função guiar o subgrupo ω (os que não foram selecionados) à provável localização da presa (solução ótima). Não existem estudos publicados, até o momento, sobre o efeito da movimentação de todos os indivíduos do grupo ou somente um subgrupo seletivo. O algoritmo utilizado neste presente trabalho movimentava apenas o subgrupo ω .

O mecanismo de busca do GWO, diferente dos algoritmos clássicos da inteligência de enxames, cria uma área de busca radial consequente da utilização de três referências ao invés de uma. O raio de busca varia de acordo com o parâmetro a , o qual decai linearmente de 2 à 0 ao decorrer das iterações. As equações (7) a (11) que descrevem o mecanismo de movimentação dos lobos são apresentadas a seguir:

$$A_i = a(2r_i - 1); i = \alpha, \beta, \delta \quad (7)$$

$$C_j = 2r_j; j = \alpha, \beta, \delta \quad (8)$$

$$D_i = |C_i X_i - X|; i = \alpha, \beta, \delta \quad (9)$$

$$R_i = X_i - A_i D_i; i = \alpha, \beta, \delta \quad (10)$$

$$X^{t+1} = \frac{R_\alpha + R_\beta + R_\delta}{3} \quad (11)$$

onde, r_w é um número aleatório com distribuição uniforme no intervalo $[0; 1]$ e $w = \{i, j\}$.

As fases de busca do algoritmo são tendenciadas pelo parâmetro a . Enquanto $a > 1$, ou seja, enquanto o número de iterações realizadas pelo algoritmo é inferior à metade das iterações que realizará, o enxame tende a focalizar-se em uma busca global. Analogamente, enquanto $0 < a < 1$, a tendência é de busca local. A Fig. 4 mostra o pseudocódigo do GWO.

```

Pseudo código GWO
Define parâmetros do algoritmo
Inicializa população aleatória
Para cada lobo do grupo faça:
  Calcula fitness
Fim para
Seleciona líderes ( $\alpha, \beta, \delta$ )
 $a < 2$ 
Para i de 2 até Max n. Iterações faça:
  Atualiza posições subgrupo  $\omega$  (equações 7:11)
  Para cada lobo do subgrupo  $\omega$  faça:
    Calcula fitness
  Fim para
  Seleciona líderes ( $\alpha, \beta, \delta$ )
  Atualiza parâmetro  $a$ 
Fim para
Solução ótima  $<- X(\alpha)$ 
  
```

Fig. 4. Pseudocódigo GWO.

V. ANÁLISE DE DINÂMICA INVERSA

A dinâmica inversa estuda as cargas e os momentos aplicados as juntas do corpo humano, tais cargas podem ser obtidas por modelos físico-matemáticos aplicados ao corpo humano que simplificam a representação do aparelho locomotor. A análise destas cargas é muito importante na busca de melhorar a eficiência do movimento do corpo humano tanto como a proteção do aparelho locomotor. A determinação de cargas intersegmentais para dados de movimento e forças de reação do solo vem sendo uma ferramenta muito utilizada na análise biomecânica de movimento, neste caso a caminhada [2].

Com o propósito de remover ruídos causados pela segunda derivada que ocorre em termos inerciais das equações de

movimento é necessária a aplicação de um filtro passa baixa para o processo.

Quatro frequências de corte dos três segmentos do modelo musculoesquelético 2-D (Fig. 5) coxa, canela e pé, mais a plataforma de força, serão determinados pelas técnicas de otimização aplicadas ao processo. Dados filtrados serão utilizados como referência para análise de dinâmica inversa 2-D padrão. Os dados da análise de dinâmica inversa de movimento foram simulados à uma amostragem de 200 fps e 0.5mm de ruído inserido para simulação de erros de digitalização.

Como critério de otimização, será feita a minimização do valor da função objetivo $BestM$ (12), que é a normalização do conjunto dos erros médios quadráticos (RMS) da diferença entre os resultados da dinâmica inversa e os momentos para cada uma das três juntas intersegmentais (quadril, joelho e calcanhar) do modelo 2-D já conhecidos do processo.

$$BestM = \frac{norm[M_{quadril}, M_{joelho}, M_{calcanhar}]}{\sqrt[3]{3}} \quad (12)$$

onde, $M_{quadril}$, M_{joelho} e $M_{calcanhar}$ são respectivamente os erros RMS das ligações quadril, joelho e calcanhar dos segmentos do modelo musculoesquelético 2-D, e $norm$ é a função de normalização aplicada aos momentos.

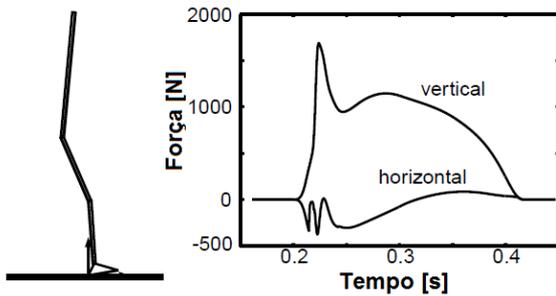


Fig. 5. Modelo 2-D musculoesquelético à esquerda, gráfico de simulação de forças de reação do solo à direita.

VI. RESULTADOS E DISCUSSÃO

Os testes foram executados no seguinte sistema, sistema operacional Windows 8.1 64-bit, CPU core i7-4700HQ 2.4 GHz, RAM 8 GB e no ambiente computacional MATLAB® R2014a. Foram utilizados os três métodos de otimização, ED, BA e GWO, com filtros de primeira, segunda e terceira ordem para cada técnica.

Para o algoritmo da ED, foram realizados 30 experimentos, 4 dimensões, com um total de 80 iterações e uma população inicial de 20 indivíduos. Os parâmetros de controle da ED utilizados foram 0.8 e 0.95 respectivamente para a taxa de cruzamento e fator de perturbação. A estratégia utilizada pela ED neste experimento foi “DE/local-to-best/1”. O tempo médio para cada experimento foi de aproximadamente 8 minutos.

Para o algoritmo BA, foram realizados 30 experimentos, 4 dimensões, com um total de 100 iterações e uma população inicial de 20 morcegos. Parâmetro de controle do algoritmo foram usados 0.5 para sonoridade e taxa de pulsação, a frequência mínima de 0 e máxima de 2. O tempo médio para cada experimento foi de aproximadamente 9 minutos.

No algoritmo GWO, foram realizados 30 experimentos, 4 dimensões, com um total de 80 iterações e uma população inicial de 20 agentes de busca. O tempo médio para cada experimento foi de 7 minutos.

As Tabela I, II e III, mostram os resultados obtidos com os experimentos.

TABELA I. TABELA COM OS VALORES DE 1ª ORDEM

1ª ordem	ED	GWO	BA
Frequência Coxa	13.4465	15.7712	16.3974
Frequência Canela	13.7231	14.0636	10.8013
Frequência Pé	12.1231	12.874	11.8499
Frequência Plataforma de Força	15.2855	15.3132	13.609
Média	4.5516	4.5539	7.4197
Desvio Padrão	0.1049	0.11341	1.8649
$BestM$	4.2645	4.3336	4.5279

TABELA II. TABELA COM OS VALORES DE 2ª ORDEM

2ª ordem	ED	GWO	BA
Frequência Coxa	18.7155	14.7917	17.7691
Frequência Canela	18.0813	17.0931	17.6183
Frequência Pé	19.2135	15.8606	20.6868
Frequência Plataforma de Força	17.007	15.7875	16.9214
Média	2.841	2.8112	5.1941
Desvio Padrão	0.074523	0.086351	1.5539
$BestM$	2.6908	2.5871	2.8564

TABELA III. TABELA COM OS VALORES DE 3ª ORDEM

3ª ordem	ED	GWO	BA
Frequência Coxa	19.1271	18.0638	14.271
Frequência Canela	16.0711	15.966	21.4142
Frequência Pé	21.4951	20.4702	19.3293
Frequência Plataforma de Força	18.0102	17.3135	17.7932
Média	3.0934	3.059	5.2365
Desvio Padrão	0.10286	0.088632	1.4012
$BestM$	2.8667	2.8807	3.0095

Tomando como referência os valores da Tabela IV [2], com o uso de um filtro passa-baixa de segunda ordem, após realizadas 10 simulações, foram encontrados os seguintes valores das médias das frequências de corte com seus respectivos desvios padrões. E a partir das frequências de corte foram obtidos os momentos das juntas intersegmentais, que foram normalizados com o propósito de comparação de desempenho.

TABELA IV. TABELA COM OS VALORES DE REFERÊNCIA

2ª ordem	Referência
Frequência Coxa	15.1 ± 1.0
Frequência Canela	14.6 ± 1.6
Frequência Pé	16.5 ± 2.0
Frequência Plataforma de Força	15.1 ± 1.2
$BestM$	3.4361

As três técnicas apresentaram um desempenho superior em comparação com os valores de referência da Tabela IV, com exceção do filtro de primeira ordem, provando ser vantajoso o uso de técnicas de otimização para melhorar o desempenho de

um filtro passa-baixa aplicado a um processo de dinâmica inversa.

As três técnicas, em relação ao valor de função objetivo apresentaram valores relativamente próximos, independente da ordem do filtro aplicado.

Na aplicação do filtro de primeira ordem, a ED obteve melhor desempenho em relação às outras técnicas. Entretanto os resultados obtidos pelas três técnicas não foram melhores do que o valor de referência (Tabela IV), apresentando certa instabilidade que pode ser observada nas curvas de momento, principalmente na curva de momento da junta do quadril, Fig. 6.

No filtro de segunda ordem, o algoritmo GWO obteve um resultado melhor em relações às técnicas em competição, todas as técnicas obtiveram melhor desempenho em relação ao valor de referência (Tabela IV), e como os valores chegaram muito

próximos se torna um pouco difícil de perceber qual técnica foi melhor apenas observando as curvas de momentos intersegmentais, Fig. 7.

Para o filtro de terceira ordem, Fig. 8, por uma diferença relativamente pequena a ED novamente obteve o melhor resultado em relação a seus competidores. Os valores de momento das três técnicas foram menores, portanto melhores, que o valor de referência (Tabela IV), entretanto ainda não foram melhores que a aplicação do filtro de segunda ordem.

Provando que para esta aplicação o filtro de segunda ordem, dentre as três ordens competidas, seria a melhor opção, neste caso, como o GWO obteve melhor desempenho nesta categoria de filtro seria a técnica recomendada para o processo em questão.

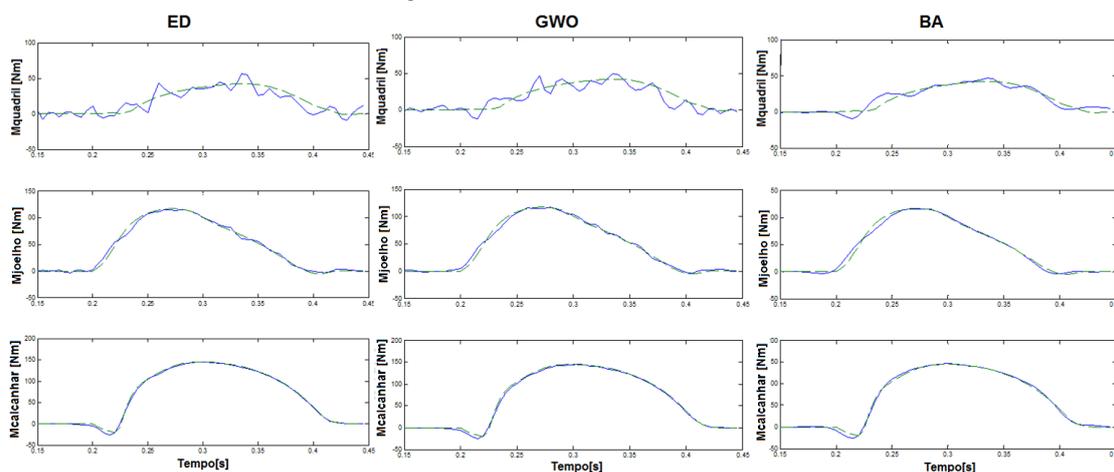


Fig. 6. Resultados de momentos intersegmentais com aplicação de um filtro de primeira ordem.

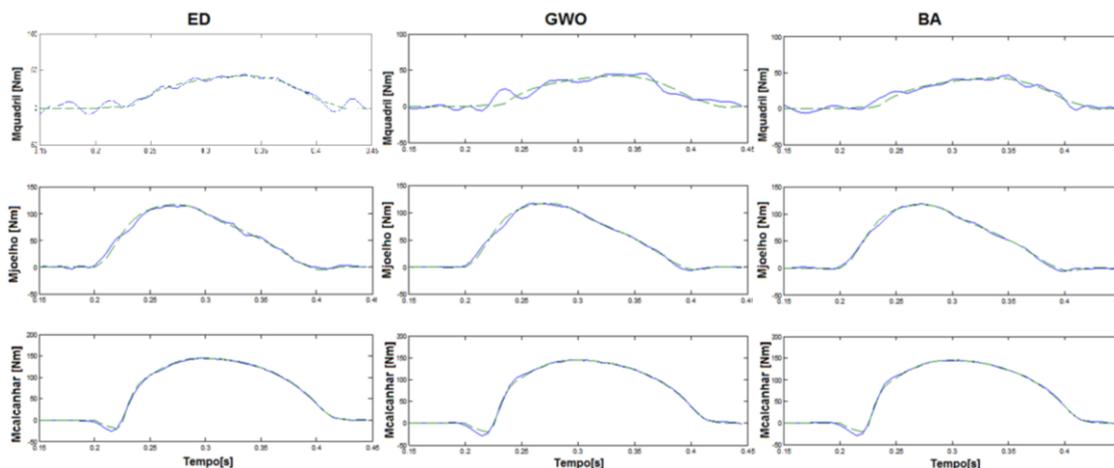


Fig. 7. Resultados de momentos intersegmentais com aplicação de um filtro de segunda ordem.

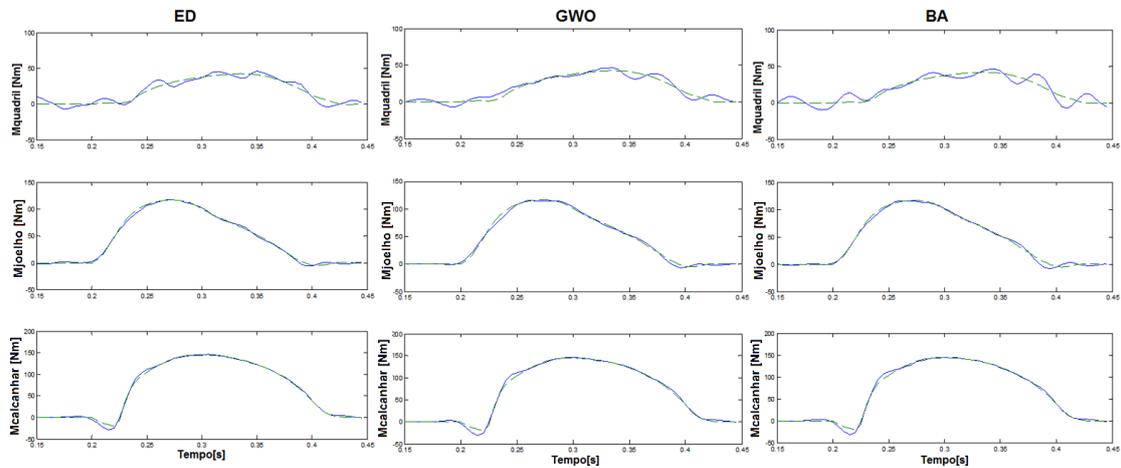


Fig. 8. Resultados de momentos intersegmentais com aplicação de um filtro de terceira ordem.

VII. CONCLUSÃO

Neste trabalho foi feita a competição entre três técnicas de otimização, ED, GWO e BA para um problema de filtro digital aplicado a um processo de dinâmica inversa. As três técnicas mostraram um desempenho superior ao valor de referência (Tabela IV), com exceção do filtro de primeira ordem, e foi visto que para esta aplicação a técnica GWO com um filtro de segunda ordem seria a melhor opção.

Para futuras referências, estender o trabalho para maiores ordens de filtro, pois apesar da técnica GWO ter obtido um resultado melhor para um filtro de segunda ordem a ED teve um melhor desempenho nas outras ordens de filtro.

Como também, é recomendado fazer o uso de técnicas multiobjectivo, como por exemplo MODE [20] uma variante multiobjectivo da ED, para minimizar não só os momentos intersegmentais, mas também as forças intersegmentais das juntas simuladas pelo modelo 2-D.

REFERÊNCIAS

- [1] A. Slowik. "Design of IIR digital filters with non-standard characteristics using differential evolution algorithm", *Bulletin of the polish academy of sciences technical sciences*, vol. 55, no. 4, pp 359-363, 2007.
- [2] A. J. Bogert, J. J. de Koning. "On optimal filtering for inverse dynamics analysis", *Proceedings of the ninth biennial conference of the canadian society for biomechanics*, Vancouver, Canada, pp. 214-215, January 1996.
- [3] K. Price, R. Storn. "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, 1995 [Online].
- [4] K. Price, R. Storn, J. Lampinen. "Differential evolution: a practical approach to global optimization". Berlin: © Springer-Verlag. pp. 538, 2005.
- [5] R. Storn. "Differential evolution design of an IIR-filter". *IEEE international conference on evolutionary computation*, pp. 268–273, 1996.
- [6] N. Goel, S. Singh. "Optimal IIR filter design using differential evolution algorithm", *International journal of computer applications*, vol. 93, no. 5, pp. 1- 13, May 2014.
- [7] S. Chattopadhyay, S. Sanyal, A. Chandra. "Design of FIR filter using differential evolution optimization & to study its effect as a pulse-shaping filter in a QPSK modulated system". *international journal of computer science and network security*, vol. 10, no. 1, pp. 313-321, January 2010.
- [8] W. Zhu, J. Fang, Y. Tang, W. Zhang, W. Du. (2012). "Digital IIR filters design using differential evolution algorithm with a controllable probabilistic population size". *Plos one* e40549, vol. 7, no. 7, pp. 1-9, July 2012.
- [9] S. Mirjalili, S. M. Mirjalili, A. Lewis. "Grey wolf optimizer". *advances in engineering software*, vol. 69, pp. 46–61, 2014.
- [10] S. Mirjalili. "How effective is the grey wolf optimizer in training multi-layer perceptrons". *Applied intelligence*, vol. 43, pp. 150–161, January 2015.
- [11] G. Komaki, V. Kayvanfar. "Grey wolf optimizer algorithm for the two-stage assembly flowshop scheduling problem with release time". *Journal of computational science*, vol. 8, pp. 109–120. 2015.
- [12] X. Song et. al. "Grey wolf optimizer for parameter estimation in surface waves". *Soil dynamics and earthquake engineering*, vol. 75, pp. 147–157, 2015.
- [13] O. Hasançebi, T. Teke, O. Pekcan. "A bat-inspired algorithm for structural optimization". *Computers and structures*, vol. 128, pp. 77–90. 2013.
- [14] A. Gandomi, X. Yang, A. Alavi, S. Talatahari. "Bat algorithm for constrained optimization tasks". *Neural computing and applications*, vol. 22, no. 6, pp. 1239–1255. 2013.
- [15] X. Yang, A. Gandomi. "Bat algorithm: a novel approach for global engineering optimization". *Engineering computations: international journal for computer-aided engineering and software*, vol. 29, no. 5, pp. 464-483, 2012.
- [16] X. Yang. "A new metaheuristic bat-inspired algorithm". *studies in computational intelligence*, vol. 284, pp. 65–74. 2010.
- [17] J. Zhang, A. Sanderson. "JADE : adaptive differential evolution with optional external archive". *IEEE transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [18] M. Mitchell. "An introduction to genetic algorithms". Massachusetts: MIT Press, pp 162, 1996.
- [19] D. Mayer, B. Kinghorn, A. Archer. "Differential evolution: an easy and efficient evolutionary algorithm for model optimization". *Agricultural systems*, vol. 83, no. 3, pp. 315-328, 2005.
- [20] G. Meza, S. Nieto, J. Sanchis, F. Blasco. "Controller tuning by means of multi-objective optimization algorithms: a global tuning framework". *IEEE transactions on control systems technology*, vol. 21, no. 2, pp. 445-458. March 2013.