ALGORITMO DE EVOLUÇÃO DIFERENCIAL MULTIOBJETIVO APLICADO À OTIMIZAÇÃO DE FILTROS RECURSIVOS - IIR

DIOGO S. DE LUCENA[1], MARIANA T. SEBASTIANI[2], LEANDRO S. COELHO[1].

1. *Departamento de Engenharia Elétrica, Universidade Federal do Paraná*
UFPR – Setor de Tecnologia/PPGEE – Caixa Postal 19011 - 81531-970 – Curitiba – PR
E-mails: diogo.lucena90@gmail.com, leandro.coelho@pucpr.br

2. *Departamento de Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina*
UFSC/CTC/DAS/PPGEAS - Sala 214 - Caixa Postal 476 - 88040-900 – Florianópolis – SC
E-mail: marianasebastiani@globo.com

Abstract— Digital filters are highly used in many fields of engineering. With the increased use of digital components, digital filters have become more usual, especially the infinite impulse response filters, which are the recursive response filters. The design of such filters proves to be a complicated task because that their coefficients are determined in the real domain, while its response is analyzed in the frequency domain. Many classical methods are known, and are presented here, for the design of these filters, but all with different limitations. Currently, many evolutionary algorithms are applied to the design of filters, however, mostly aimed only treating the magnitude error problem. This project aims to apply the optimization technique known as Differential Evolution to optimize not only the magnitude of error, but also linear phase error.

Keywords—Evolutionary computation and bioinspired computation, digital filters, multiobjective optimization.

Resumo— Filtros digitais são altamente utilizados em diferentes áreas da engenharia. Com uma maior utilização de componentes digitais, os filtros digitais se tornaram mais usuais, principalmente os filtros de resposta de impulso infinito, que são filtros de resposta recursiva. O projeto de tais filtros se mostra uma tarefa complicada, pois seus coeficientes são determinados no domínio dos reais, enquanto sua resposta é analisada no domínio das frequências. Muitos métodos clássicos são conhecidos, como os métodos de Butterworth e Chebyshev, para o projeto destes filtros, porém todos com a presença de alguma limitação diferente. Atualmente, muitos algoritmos evolutivos são aplicados para o projeto de filtros, porém, em sua maioria, tem como foco apenas o tratamento do problema de erro de magnitude. Este artigo busca aplicar a técnica de otimização evolução diferencial com abordagem multiobjetivo para otimizar, não somente o erro de magnitude, mas também o erro de fase linear.

Palavras-chave—Computação evolucionária e computação bioinspirada, filtros digitais, otimização multiobjetivo.

1 Introdução

Filtros digitais são muito utilizados em processamento digital de sinais para diversas aplicações, entre as quais se incluem: processamento de voz e de dados de comunicação; processamento de imagens e de vídeo; sonares; radares; exploração de óleo; eletrônica; dentre diversas outras, como mostrado por Das e Konar (2007).

Quando se projetam filtros digitais, espera-se definir uma faixa de frequência pela qual a parte do sinal de entrada presente nesta será removida e aquela fora desta faixa ficará intacta. Desta forma, o projeto de filtros é geralmente feito no domínio das frequências, especificando esta, então chamada, frequência de corte, também mostrado por Das e Konar (2007).

Diferentes formas de projeto de filtros já foram introduzidas no decorrer dos anos, buscando otimizar os coeficientes do filtro a fim de obter uma resposta mais precisa. Cientistas e engenheiros de todas as áreas frequentemente têm que resolver problemas de busca e otimização, que significa encontrar a solução com melhor resultado ao problema, respeitando certas restrições e flexibilidades do mesmo, como dito por Yu e Xinjie (2007).

Além disso, muitos problemas do mundo real envolvem a otimização simultânea de muitas funções objetivo. Geralmente, estas funções não podem ser comparadas entre si e frequentemente possuem objetivos conflitantes e competitivos, ou seja, melhorar um objetivo vai, muitas vezes, significar piorar o outro. Isto resulta em um grupo de soluções que podem ser consideradas equivalentes quando nenhuma informação externa é dada em relação à importância de cada objetivo.

Tratando de algoritmos multiobjetivos dois conceitos são utilizados: dominância e Pareto-ótimo, que são amplamente discutidos em Deb *et. al*, 2002.

O problema de otimização de filtros de resposta ao impulso infinito (IIR do inglês *impulse infinite response*), se encaixa neste problema de otimização multiobjetivo. Suas equações, funções de minimização e restrições serão apresentadas neste artigo.

2 Problema de filtros IIR

Um dos principais problemas encontrados no projeto de filtros digitais é encontrar parâmetros para este filtro de forma a ter a frequência de corte bem definida e brusca, pois nos filtros reais, frequências próximas a de corte que deviam passar sem alteração sofrem pequenas distorções, enquanto aquelas que deviam ser removidas são apenas suavizadas. A equação (1) descreve matematicamente um filtro IIR,

como mostrado por Etter, Hicks e Cho (1982), Shynk (1989) e Tang *et al* (1998),

$$H(z)=\frac{b_m z^m+b_{m-1}z^{m-1}+\ldots+b_1 z+b_0}{z^n+a_{n-1}z^{n-1}+\ldots+a_1 z+a_0} \quad (1)$$

Em que a_i com $i = 0, 1, 2, \ldots, n$ e b_j com $j = 0,1,2, \ldots, m$ são os coeficientes do filtro, sendo n a ordem em que $n \geq m$. Desta forma, o número de coeficientes a ser definido é $N = n + m + 1$.

Mello (2012) mostra que 4 tipos de projeto de filtros são largamente utilizados na prática: Butterworth, Chebyshev I, Chebyshev II e Elíptico. A fim de avaliar a qualidade dos resultados obtidos neste trabalho, estes serão comparados com os valores encontrados utilizando estes quatro métodos clássicos de projeto. Para o projeto destes filtros foi utilizada a toolbox do *Matlab*® chamada *FDAtool*®, sempre com filtros de ordem 4.

Os objetivos propostos no problema de otimização desse trabalho são o erro mínimo da resposta de magnitude e o erro mínimo de resposta da fase linear. Tais objetivos são explicados em mais detalhes nas seções seguintes.

2.1 Erro de resposta de magnitude

A busca pelo erro de resposta de magnitude mínimo tem como objetivo, para o caso dos filtros passa baixa, fazer com que as frequências menores do que a frequência de corte definida sejam alteradas o menos possível e as acima da dela sejam ao máximo eliminadas. Para o cálculo do erro de resposta de magnitude, é usada a diferença absoluta entre a resposta em magnitude do filtro ideal e a do filtro obtido. Os cálculos são feitos no domínio da frequência com esta sendo normalizada entre 0 e 1. A equação (2) mostra esta diferença para a banda passante, enquanto a equação (3) para a banda de corte no caso de um filtro passa baixa.

$$H_p(\omega)=\begin{cases}|H(e^{j\omega})|-1, & \omega<\omega_c \\ 0, & \omega\geq\omega_c\end{cases} \quad (2)$$

Tal que ω é a frequência, com $0\leq\omega\leq1$, $H_p(\omega)$ é o erro de resposta de magnitude na banda passante na frequência ω e ω_c é a frequência de corte, com $0<\omega_c<1$.

$$H_c(\omega)=\begin{cases}0, & \omega<\omega_c \\ |H(e^{j\omega})|-0, & \omega\geq\omega_c\end{cases} \quad (3)$$

Tal que $H_c(\omega)$ é o erro de resposta de magnitude na banda de corte na frequência ω. A figura 1 mostra um exemplo para um filtro passa baixa com $\omega_c=0,3$.

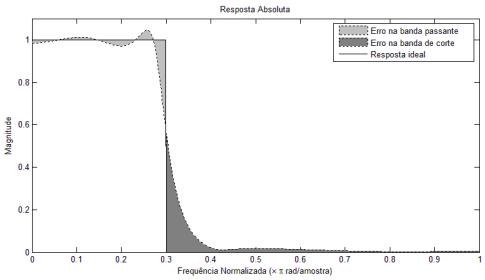

Figura 1. Exemplo de erro de resposta absoluta

Desta forma, a primeira função a ser minimizada é apresentada na equação (4).

$$\min f_1=\frac{0,5}{\omega_c}\sum H_p(\omega)+\frac{0,5}{(1-\omega_c)}\sum H_c(\omega) \quad (4)$$

Com ω assumindo valores de 0 a 1. Os coeficientes multiplicando os somatórios tem como objetivo atribuir mesmo peso para o erro da banda passante assim como o da banda de corte.

2.2 Erro de resposta de fase linear

A não linearidade da resposta de fase implica em diferentes atrasos no sinal para diferentes frequências. Tal diferença traz distorções no sinal de saída conforme as frequências do sinal de entrada e o quão não linear é a resposta do filtro. Ambas as banda passante e banda de transição são consideradas quando se calcula o erro de resposta de fase linear como mostra Yu e Xinjie (2007). Os pontos de amostragem são igualmente distribuídos na banda de passagem e de transição. Então a sequência de diferenças de fase de primeira ordem $\{\theta_1, \theta_2, \theta_3, \ldots, \theta_{P_n+T_n}\}$, onde $\theta_i = \angle H(\omega_i)$ são calculadas conforme a equação (5).

$$\Delta Fase = \{\Delta\theta_1, \Delta\theta_2, \Delta\theta_3, \ldots, \Delta\theta_{P_n+T_n-1}\} \quad (5)$$

De forma que $\Delta\theta_i = \theta_{i+1} - \theta_i$; P_n e T_n são as frequências de amostragem na banda passante e na banda transitória, respectivamente.

Agora, conforme a equação (6), o erro de resposta de fase linear pode ser calculado, sendo esta a segunda equação a ser minimizada.

$$\min f_2 = var\{\Delta\theta_i/\theta_i \in (bp \cup bt)\} \quad (6)$$

Tal que var é um operador de cálculo de variância, *bp* significa banda passante e *bt* banda transitória. Se f_2 atinge seu melhor valor, ou seja, $f_2 = 0$ representa que todos os elementos de $\Delta Fase$ são exatamente iguais.

2.3 Critério de estabilidade

A única restrição presente para o projeto de filtros IIR é a estabilidade do filtro. Smith (2012) apresenta duas formas de testar esta característica. A primeira, e mais comum, é a verificação do posicionamento dos polos, em que um filtro é dito estável se

os módulos de todos os polos forem menores do que 1. Porém, Smith (2012) apresenta um método mais rápido e eficaz de fazer essa verificação, chamado Durbin, e ele será utilizado neste trabalho.

2.4 Formulação do problema de projeto de filtros IIR

O problema de projeto de filtros digitais pode ser formulado conforme a equação de otimização multi-objetivo da equação (7),

$$\begin{aligned} &\min f = \{f_1(x), f_2(x)\} \\ &\text{sujeito a } x \in X \text{ e a estabilidade do filtro} \\ &\text{objetivo 1: } f_1(x)\text{-erro de resposta de magnitude} \\ &\text{objetivo 2: } f_2(x)\text{-erro de resposta de fase linear} \end{aligned} \quad (7)$$

em que $X \subset (R^N)$ denota o espaço de soluções com N sendo o número de variáveis, em que $N = n + m + 1$ representando os coeficientes do filtro; $x = \{x_1, x_2, x_3, \dots, x_{M+N}\} \in R^N$ é o vetor das variáveis de decisão; $f: X \to R^2$ representa as duas funções objetivos para mapear de N dimensões do espaço das variáveis para as 2 dimensões das funções objetivo $f = \{f_1(x), f_2(x)\}$.

3 DEMO-F

Robic e Filipic (2005), criaram o DEMO (Evolução Diferencial para Otimização Multiobjetivo, do inglês *Differential Evolution for Multiobjective Optimization*) que faz uso das vantagens da evolução diferencial (ED) combinada com os mecanismos de classificação do NSGA-II (Algoritmo Genético por Ordenação de Não-Dominância II, do inglês *Nondominated Sorting Genetic Algorithm II*) proposto por Deb *et al.* (1994) que usa classificação por frentes e distância de multidão para classificação dos indivíduos. Três versões do algoritmo são apresentadas por Robic e Filipic (2005): DEMO/Parent, DEMO/Mais próximo/Decisão e DEMO/Mais próximo/Objetivo. Elas se diferem entre si pela forma em que a seleção dos indivíduos que permanecem para a próxima geração é feita em cada uma (seleção feita antes do uso dos mecanismos do NSGA-II).

O ED é um dos mais poderosos algoritmos de otimização de parâmetros reais em uso corrente, surgiu como uma forma muito competitiva de computação evolutiva há mais de uma década. O primeiro artigo sobre o assunto foi escrito por Storn e Price em 1995 (STORN; PRICE, 1995), onde a versão original do algoritmo com abordagem mono-objetivo pode ser encontrado.

Além da versão original, variações melhoradas do algoritmo foram criadas, tais como: SaDE criada por Qin, Huang e Suganthan (2009); jDE apresentado por Brest *et al.* (2006); DE com vizinhança global e local (DEGL) apresentado por Das *et al.* (2009); JADE desenvolvido por Zhang e Sanderson (2009); entre outros.

Já o primeiro algoritmo evolutivo multiobjetivo usando o conceito de dominância foi o NSGA (Algoritmo Genético por Ordenação de Não-Dominância, do inglês *Nondominated Sorting Genetic Algorithm*) proposto por Srinivas e Deb (1994). Posteriormente em Deb *et al.* (2002), é apresentado o NSGA-II, que é uma versão melhorada do NSGA. No NSGA-II é incorporada a ordenação por não-dominação para o processo de seleção, em que a população da geração atual e a nova população gerada com as operações de cruzamento e mutação são colocadas juntas e inicia-se o processo de classificação por frentes.

No DEMO/Parent, após o uso do tradicional DE/aleatório/1/binário para mutação e cruzamento dos indivíduos da população, cada vetor experimental é comparado com o vetor base usado para criá-lo através do conceito de dominância apresentado anteriormente, satisfazendo uma das condições:

1. Vetor experimental domina o vetor base: o vetor base é descartado;
2. Vetor experimental é dominado pelo vetor base: o vetor experimental é descartado;
3. Vetor experimental não domina o vetor base e nem é dominado pelo mesmo: os dois vetores ficam na população.

As outras duas versões do algoritmo DEMO se baseiam na mesma estratégia apresentada, a diferença é que, no caso do DEMO/Mais Próximo/Decisão, a comparação é feita não mais entre o vetor experimental e o vetor base, mas entre o vetor experimental e o indivíduo mais similar a ele no espaço de decisões (domínio das variáveis).

Já no caso do DEMO/Mais Próximo/Objetivo a comparação é feita entre o vetor experimental e o mais similar a ele no espaço dos objetivos como mostrado por Robic e Filipic (2005).

O fluxograma de funcionamento do DEMO é apresentado na figura 2

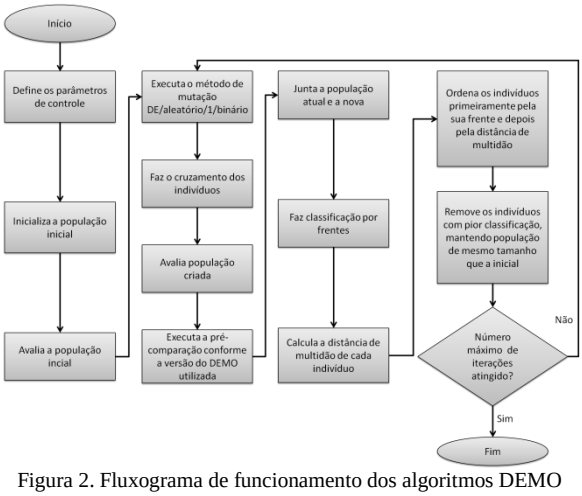

Figura 2. Fluxograma de funcionamento dos algoritmos DEMO

Foi então definida uma variação do DEMO com a utilização de um valor *F*, que é o fator de escala no processo de mutação, igual a um número aleatório baseado na curva normal, com média 0,5 e desvio padrão de 0,2, como o utilizado no algoritmo SaDE de Qin e Huang (2009). O novo algoritmo é chamada então de DEMO-F. A validação do algoritmo foi feita em quatros funções que foram introduzidas por

Zitzler, Deb e Thiele (2000) e são frequentemente utilizadas na literatura como em Xue, Sanderson e Graves (2003) e em Robic e Filipic (2005).

4 Resultados

Após todos os estudos desenvolvidos para definir o melhor algoritmo e os melhores parâmetros de controle para o mesmo, finalmente este foi aplicado no problema de projeto de filtros multiobjetivo. Para tanto, foi utilizado o algoritmo DEMO-F com *CR*, probabilidade de cruzamento, igual a 0,21 – definido de forma empírica, população de 100 indivíduos e número máximo de iterações de 2000.

A ordem do filtro a ser otimizado é 4 e o algoritmo busca um filtro passa-baixa com frequência de corte de 0,3. É importante ressaltar que a decisão pelo filtro passa-baixa foi feita devido ao filtro ser, em geral, o mais utilizado em aplicações reais. O algoritmo tem o mesmo funcionamento e, possívelmente, o mesmo nível de resultado para os outros tipos de filtros como passa-alta, passa-banda e corta-banda, porém tal hipótese não será provada neste trabalho.

A tabela 1 mostra uma comparação percentual dos filtros encontrados através do algoritmo DEMO-F em 10 execuções, seguindo os parâmetros anteriormente especificados, com os filtros obtidos através dos projetos clássicos. Entretanto, uma análise estatística mais aprofundada é necessária para provar a consistência da qualidade dos resultados do DEMO-F. Na tablea 1, a primeira e a segunda coluna mostram, respectivamente, as soluções do algoritmo DEMO-F que obtiveram melhores resultados que aqueles obtidos com o uso das técnicas tradicionais de projeto de filtro para os objetivos 1 e 2. Já nas últimas duas colunas, é mostrado a porcentagem das soluções em que o filtro projetado foi melhor que cada um dos filtros clássicos em ambos os objetivos e também os casos em que o os filtros clássicos se sobressaíram em ambos os objetivos comparativamente com o DEMO-F.

Tabela 1. Comparação percentual dos indivíduos encontrados pelo algoritmo DEMO-F em 10 runs (1000 indivíduos) em relação aos filtros de projeto clássico

Filtro	$f_{1_{projetado}} < f_{1_{filtro}}$	$f_{2_{projetado}} < f_{2_{filtro}}$	$f_{1,2_{projetado}} < f_{1,2_{filtro}}$	$f_{1,2_{projetado}} > f_{1,2_{filtro}}$
Butterworth	82%	100%	82%	0%
Chebyshev I	72%	100%	72%	0%
Chebyshev II	99%	100%	99%	0%
Elíptico	67%	100%	67%	0%

Através da tabela 1 é possível notar que poucos resultados do conjunto de soluções dados pelo DEMO-F tiveram desempenho inferior em qualquer um dos objetivos quando comparados àqueles obtidos pelo projeto de filtros clássicos. E, pela última coluna da tabela, é visto que, quando os resultados de DEMO-F são inferiores aos filtros de projeto clássico em algum dos objetivos, no outro objetivo ele é superior, desta forma, não é possível dizer que qualquer resultado obtido com o filtro de projeto clássico é melhor do que o resultado de DEMO-F sem um conhecimento prévio da importância de cada um dos objetivos. Em contrapartida, muitos dos indivíduos encontrados no conjunto solução do algoritmo DEMO-F são superiores aos valores dos filtros de projeto clássico em ambas as funções objetivo, o que faz esses resultados serem indiscutivelmente melhores, independente de qualquer conhecimento prévio.

Para os resultados apresentados no restante do trabalho, foi selecionada a melhor frente encontrada nas 10 execuções com relação à convergência e diversidade da mesma.

A figura 3 mostra a evolução dos indívuos em cada um dos objetivos através das iterações e como a frente de Pareto é obtida ao final das iterações.

Após a execução do algoritmo, três indivíduos foram escolhidos a fim de exemplificar a solução obtida. O primeiro, Projetado 1, é o que obteve o melhor resultado para o primeiro objetivo, já o Projetado 2 teve o melhor resultado para o segundo objetivo e o Projetado 3 obteve uma resposta intermediária em ambos os objetivos. A tabela 2 apresenta o resultado dos três filtros junto com os resultados dos filtros clássicos para os dois objetivos. Nota-se que o resultado obtido pelos filtros projetados são superiores aos filtros clássicos.

Tabela 2. Comparação de 3 filtros projetados com os filtros de projeto clássico

Filtro	f_1	f_2
Projetado 1	**30,833**	0,0015
Projetado 2	90,680	**0,00000**
Projetado 3	35,072	0,00012
Butterworth	69,032	0,01995
Chebyshev I	60,951	0,02013
Chebyshev II	118,010	0,03971
Elíptico	51,386	0,03994

A figura 4 apresenta as respostas de magnitude e de fase para o filtro Projetado 1. Nota-se que há oscilação tanto na banda de passagem como na banda de corte, porém sua amplitude é menor do que a apresentada pelo filtro Elíptico (que possui a mesma característica de resposta) e também a linearidade na resposta de fase é maior.

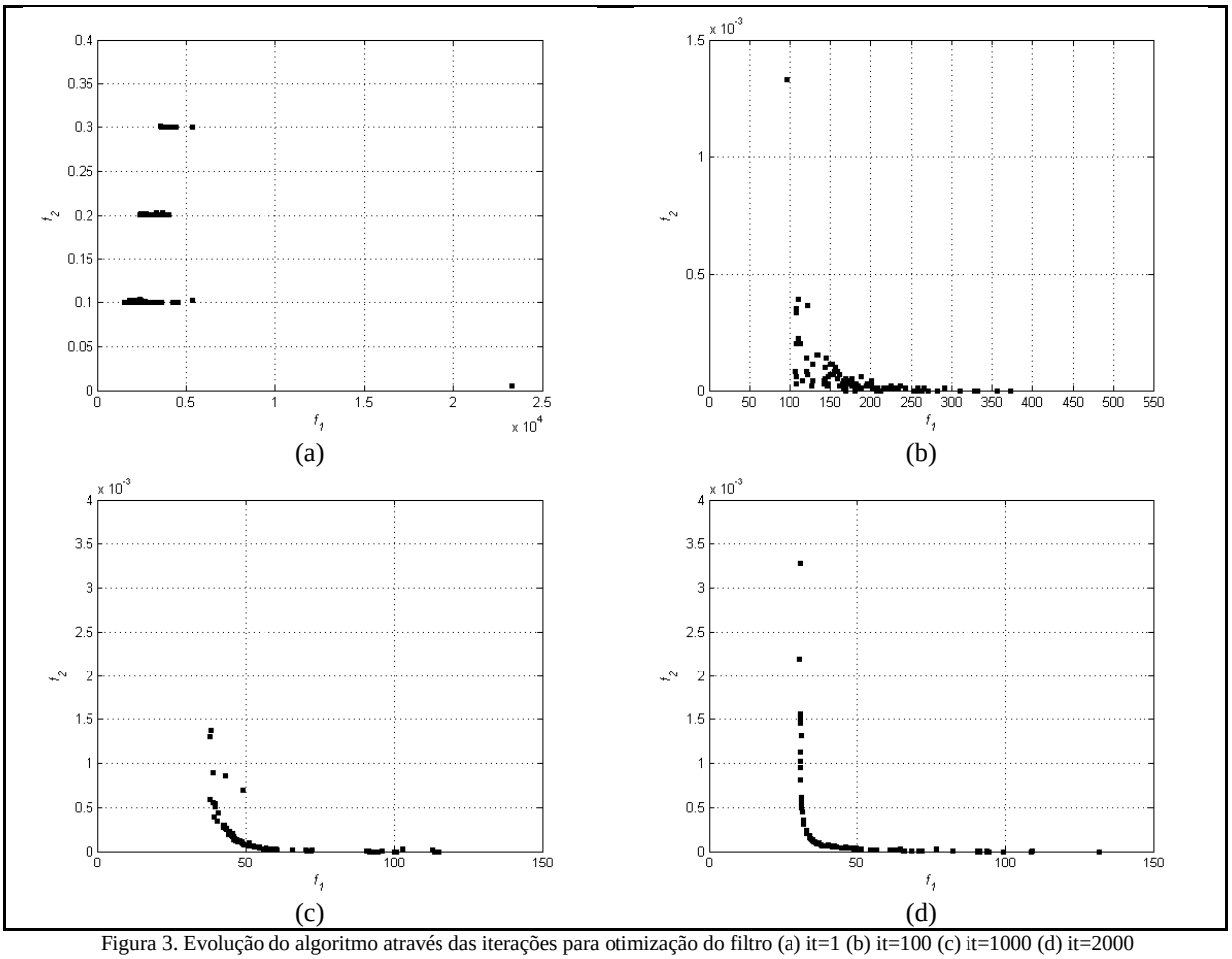

Figura 3. Evolução do algoritmo através das iterações para otimização do filtro (a) it=1 (b) it=100 (c) it=1000 (d) it=2000

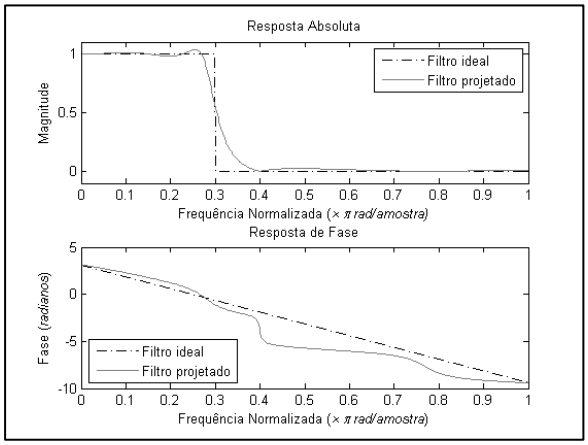

Figura 4. Repostas absoluta e de fase para o filtro Projetado 1

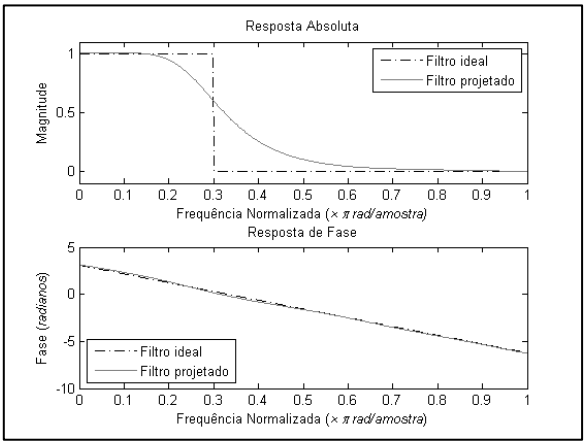

Figura 5. Repostas absoluta e de fase para o filtro Projetado 2

Já para o filtro Projetado 2 as respostas são apresentadas na figura 5. As respostas em banda passante e em banda de corte não oscilam, porém a fase transitória tem declive pouco íngreme, resposta que se assemelha à do filtro Butterworth, mas o erro em ambas as respostas é menor. Nota-se que a resposta de fase obtida é semelhante a reposta linear do filtro ideal.

Finalmente, para o filtro Projetado 3 as respostas são apresentadas na figura 6. A oscilação ocorre somente na banda de passagem e com amplitude baixa quando comparada com as respostas dos outros filtros. Este filtro possui uma resposta de fase mais linear do que a do filtro Projetado 1.

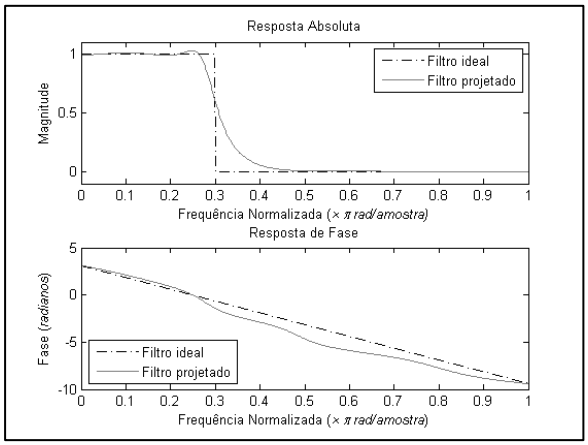

Figura 6. Repostas absoluta e de fase para o filtro Projetado 3

5 Conclusão

Os resultados alcançados com o uso do DEMO-F no problema multiobjetivo de otimização no projeto de filtros apresentaram valores satisfatórios, mostrando grande superioridade àqueles encontrados com os métodos clássicos de projeto de filtros. Uma análise estatística mais aprofundada é necessária para provar a consistência da convergência do algoritmo, porém para comportar tais resultados, outros tópicos necessitariam ser removidos deste trabalho, desta forma foi optado por deixar esta etapa como um trabalho futuro.

Futuros projetos consistem também na adição de um terceiro objetivo ao problema de otimização, em que além dos dois objetivos apresentados neste trabalho, também buscaria-se a minimização da ordem do filtro. Tal implementação tem grande impacto no método de otimização, pois todo o sistema de codificação dos parâmetros deve ser repensado uma vez que o tamanho de cada indivíduo seria alterado no passar das iterações.

Referências Bibliográficas

Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transaction on Evolutionary Computation, v.10, n.6, p.646–657.

Das, S.; Abraham, A.; Chakraborty, U.K.; Konar, A. (2009). Differential evolution using a neighborhood based mutation operator.IEEE Transactions on Evolutionary Computation, v.13, n.3, p.526–553.

Das, S.; Konar, A. (2007). A swarm intelligence approach to the synthesis of two-dimensional IIR filters. Science Direct – Engineering Application of Artificial Intelligence 20, p.1086-1096.

Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, v. 6, p. 182 – 197.

Etter, D.M.; Hicks, M.J; Cho, K.H. (1982). Recursive adaptive filter design using an adaptive genetic algorithm. IEEE Conference ASSP, p.635–638.

Mello, C. Técnicas de Projeto de Filtros. Universidade Federal de Pernambuco. Disponível em: <http://www.cin.ufpe.br/~cabm/pds/PDS_Aula05%20Projeto%20de%20Filtros.pdf>. Acesso em: 23 de janeiro de 2013.

Qin, A.K.; Huang, V.L.; Suganthan, P.N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization.IEEE Transactions on Evolutionary Computation, v.13, n.2, p.398–417.

Robic, T; Filipic, B. (2005). Differential Evolution for Multiobjective Optimization in Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization – EMO, p 520-533.

Shynk, J.J. (1989). Adaptive IIR filtering, IEEE ASSP Magnetism, v.6, n.2, p.4–21.

Smith, J. Introduction to Digital Filters with Audio Applications. Department of Music, Stanford University. Disponível em: <https://ccrma.stanford.edu/~jos/filters/>. Acesso em: 15 de janeiro de 2013.

Srinivas, N.; Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms.Evolutionary Computation, v. 2, p. 221 – 248.

Tang, K.S.; Man, K.F.; Kwong, S.; LIU, Z.F. (1998). Design and optimization of IIR filter structure using hierarchical genetic algorithms. IEEE Transactions on Electronics, v.45, n.3, p.481-487.

Yu, Y.; Xinjie, Y. (2007). Cooperative coevolutionary genetic algorithm for digital IIRfilter design. IEEE Transaction Electron, v.54, n.3, p.1311–1318.

Xue, F.; Sanderson, A.; Graves, R. (2003). Pareto-based multi-objective differential evolution. Proceedings of the 2003 Congress on Evolutionary Computation, v. 2, p. 862 - 869.

Zhang, J.; Sanderson, A.C. (2009). Adaptive differential evolution with optional external archive.IEEE Transactions on Evolutionary Computation., v.13, n.5, p.945–958.

Zitzler, E.; Deb, K.; Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, v. 8, p. 173 – 195.