

Evrimsel Algoritma Tabanlı FIR Filtre Tasarım Simülatörü

The FIR Filter Simulator based on Evolutionary Algorithm

¹Yigit Cagatay Kuyu, ¹Nedim Aktan Yalcin, *¹Fahri Vatansever

*¹Faculty of Engineering, Electrical-Electronics Engineering Dept., Uludag University, Turkey

Özet

Evrimsel hesaplama algoritmalarının gelişmesiyle, uygulama alanları da genişlemiştir. Bu alanlardan birisi de elektronik mühendisliğindeki filtre tasarımlarıdır. Gerçekleştirilen çalışmada FIR filtrelerin tasarımları için - eğitim amaçlı da kullanılabilecek - bir simülatör geliştirilmiştir. Kullanıcı dostu simülatör ile parametreleri (filtrenin türü, derecesi, kesim frekansı/frekansları) girilen filtre hem standart yöntemlerle hem de seçilen evrimsel yöntemlerle (genetik algoritma, diferansiyel gelişim algoritması, parçacık sürü optimizasyon algoritması vb.) tasarlanabilmektedir. Elde edilen karşılaştırmalı sonuçlar hem sayısal (filtre katsayıları vb.) hem de grafiksel (cevap eğrileri: genlik, faz, birim dürtü, birim basamak vb., sıfır-kutup dağılımı vb.) olarak kullanıcıya sunulabilmektedir. Böylece parametrelerin ve yöntemlerin etkisi/verimliliği açıkça görülebilmektedir.

Anahtar kelimeler: FIR, simülatör, evrimsel algoritma.

Abstract

Associated with development in evolutionary computation algorithms, theirs application fields have also extended. One of these fields are design filters in electronic engineering. In this study, a simulator has been developed to be used also for educational purposes for the design of FIR filters. With user friendly simulator, parameters (type of filter, degree of filter, cut off frequency) that entered the filter can be design both evolutionary methods (genetic algorithm, differential evolution, particle swarm optimization etc.) and standard methods. The obtained comparative results can be presented to the user as both numeric (filter coefficient etc.) and graphical (response curves: amplitude, phase, unit impulse, unit step response etc., zero-pole distribution etc.). In this way, effect and productivity of parameters and methods can be clearly seen.

Keywords: FIR, simulator, evolutionary algorithm.

1. Giriş

Elektrik-elektronik mühendisliğindeki temel işlemlerden birisi de filtreleme işlemidir. Uygulama alanlarına özel filtre tasarımları yapmak için farklı yöntemler mevcuttur. Ancak ideale yakın filtreler gerçeklemek için hesaplama/tasarım aşamaları artmakta/zorlaşmaktadır. Bu nedenle klasik tasarım yöntemlerinin yanında artık evrimsel algoritmalar da kullanılmaya başlanmıştır.

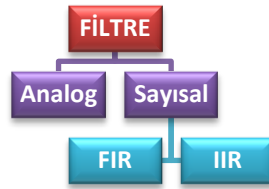
Evrimsel algoritmalar olan genetik algoritmalar (GA) ile sayısal filtre tasarımına yönelik çalışmalar 1980'li yıllarda başlamış ve günümüze kadar da geliştirilerek kullanımı artmıştır[1-3].

Sayısal filtre tasarımı diğer evrimsel algoritmalar olan diferansiyel gelişim algoritması (DE)[3-6] ve parçacık sürü optimizasyonu algoritması (PSO)[7-9] da sıklıkla kullanılmıştır.

Bu çalışmadaki simülatör ile kullanıcının seçimine bağlı olarak belirlediği optimizasyon algoritması veya standart yöntemler yardımıyla istenilen derece ve türdeki filtreler gerçekleştirilebilmekte; birçok cevapları ve değerleri elde edilebilmekte; karşılaştırmalı tasarımlar yapılabilmektedir; frekans faz ve genlik cevaplarını çizdirebilmektedir. Böylece kullanıcı dostu ve eğitim amaçlı da kullanılabilecek simülatör ile sayısal ve grafiksel olarak elde edilen sonuçlarla filtre tasarımları kolay ve etkin bir şekilde gerçekleştirilebilmektedir.

2. Sayısal Filtreler

İşaret ve görüntü işleme alanındaki en önemli işlemlerden birisi de filtrelemedir. Filtreler, yapıları itibarıyla analog ve sayısal filtreler olarak ikiye ayrılmaktadırlar. Analog filtrelerde işlemler devre elemanlarıyla gerçekleştirirken sayısal filtrelerde ise yazılımsal olarak yapılmaktadır. Sayısal filtreler de kendi içlerinde sonlu dürtü cevaplı (FIR) ve sonsuz dürtü cevaplı (IIR) olarak iki grupta toplanabilmektedirler (Şekil 1).

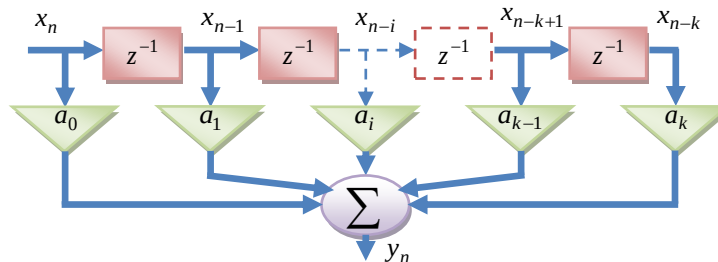


Şekil 1. Filtrelerin genel gruplandırılması

FIR filtrelerin IIR filtrelere göre başlıca avantajları doğrusal fazlı frekans cevapları ve sabit grup gecikmeleridir. Bir FIR filtrenin çıkışı

$$y_n = \sum_{i=0}^k a_i \cdot x_{n-i} \quad (1)$$

eşitliğiyle verilmektedir (Şekil 2). Filtrenin katsayıları, merkez katsayıya göre simetrik veya asimetrik olabilmektedir. Buna göre de dört filtre tipi oluşabilmektedir. Ayrıca bu filtre katsayılarının hesaplanması da genellikle basit ve doğrusaldır. İdeal FIR filtreler için normalize edilmemiş filtre katsayılarının hesaplanmasında kullanılan denklemler Tablo 1'de özetlenmiştir [10-13].



Şekil 2. FIR filtre temel yapısı

Tablo 1. İdeal FIR filtre katsayıları

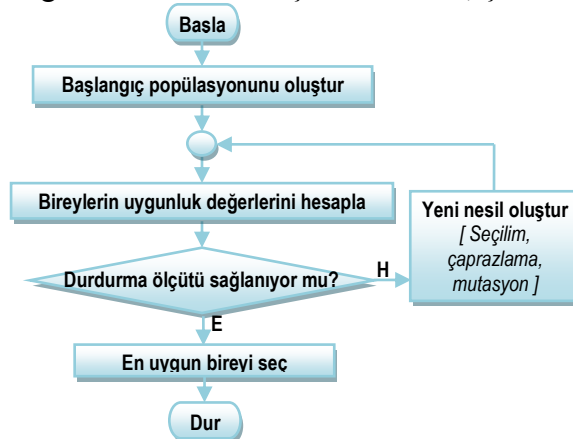
Filtre	Merkez katsayısı	Diğer katsayılar
<i>Alçak geçiren</i>	$h[0] = \frac{\omega_c}{\pi}$	$h[n] = \frac{\sin(n\omega_c)}{n\pi}$
<i>Yüksek geçiren</i>	$h[0] = 1 - \frac{\omega_c}{\pi}$	$h[n] = -\frac{\sin(n\omega_c)}{n\pi}$
<i>Bant geçiren</i>	$h[0] = \frac{\omega_{c2} - \omega_{c1}}{\pi}$	$h[n] = \frac{\sin(n\omega_{c2}) - \sin(n\omega_{c1})}{n\pi}$
<i>Bant durduran</i>	$h[0] = 1 - \frac{\omega_{c2} - \omega_{c1}}{\pi}$	$h[n] = \frac{\sin(n\omega_{c1}) - \sin(n\omega_{c2})}{n\pi}$

Sayısal filtre katsayılarını elde etmek için birçok yöntemler geliştirilmiştir. Tasarımdaki temel amaç, çıkışın ideale yakın olması olduğundan; bu durum filtrenin derecesini arttırmakta, dolayısıyla da hesaplamadaki toplama ve çarpma işlemlerini fazlalaştırmaktadır. Filtre tasarımıdaki hesaplama sayılarını/aşamalarını azaltmak için son yıllarda evrimsel algoritmalar da faydalanılmaktadır.

3. Evrimsel Algoritmalar

3.1. Genetik algoritmalar

Genetik algoritmalar, 1975 yıllarında John Holland tarafından geliştirilerek kullanılmıştır. Doğada evrimsel sürece dayanan bu algortmada; başlangıç popülasyonu, her nesilde doğal seçim ve tekrar türetme ile geliştirilmekte ve en iyi birey, en uygun (optimum) çözüm olarak ortaya çıkmaktadır. Genetik algoritmalar; birçok alandaki optimizasyon problemlerinin çözümünde, deneysel ve pratik uygulamalarda, sınıflandırma problemlerinde, bilgi sistemlerinde vb. kullanılmaktadır. Genetik algortmadaki temel işlem adımları, Şekil 3'te özetlenmektedir[14-16].



Şekil 3. GA'daki genel işleyiş adımları

3.2. Diferansiyel gelişim algoritması

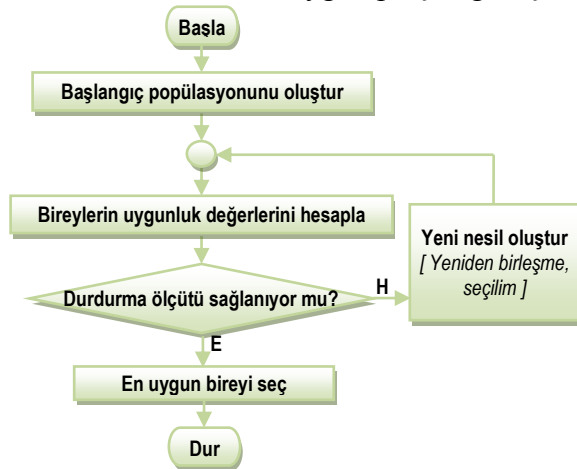
Price ve Storn tarafından 1995 yılında geliştirilmiş olan diferansiyel gelişim algoritması, temel olarak 4 evreden (başlatma, mutasyon, yeniden birleşme ve seçim) oluşan işleyişin adımları aşağıda özetlenmektedir (Şekil 4)[16-18]:

- i. Başlatma: Tüm parametre uzayını kapsayacak şekilde olası çözümlerin, rastgele bir popülasyon vektörü oluşturulur.
- ii. Mutasyon: Hedef vektör dışında, üç vektör seçilerek bu vektörler mutasyon çarpanı yardımıyla mutasyona uğrattılır.
- iii. Yeniden birleşme: Önceki nesilden gelen en iyi çözümleri birleştirilir.
- iv. Seçilim: Yeni popülasyona seçilecek bireyler belirlenir.

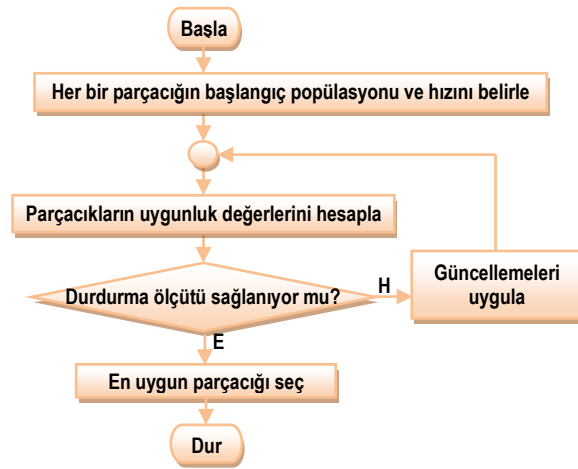
3.3. Parçacık sürü optimizasyonu algoritması

Parçacık sürü optimizasyonu algoritması, sürü halinde hareket eden balıklar ve böceklerden esinlenerek Kenedy ve Eberhart tarafından geliştirilmiş bir optimizasyon yöntemidir. Bu algoritmanın çözüm aşamaları aşağıdaki gibi özetlenebilir (Şekil 5) [16, 19]:

- i. Parçacıkların başlangıç pozisyonu ve hız bilgisini oluştur.
- ii. Parçacıkların uygunluk değeri hesapla.
- iii. Durdurma ölçütü sınaması
 - a. Sağlanmıyorsa:
 - Parçacıkları uygunluk değerlerine göre seç.
 - Popülasyondaki yerel ve global en iyi parçacık değerlerini güncelle.
 - Parçacık hızını ve parçacık pozisyonunu güncelle.
 - (ii) adıma dön.
 - b. Sağlanıyorsa:
 - En uygun parçacığı seç.



Şekil 4. DE'deki genel işleyiş adımları

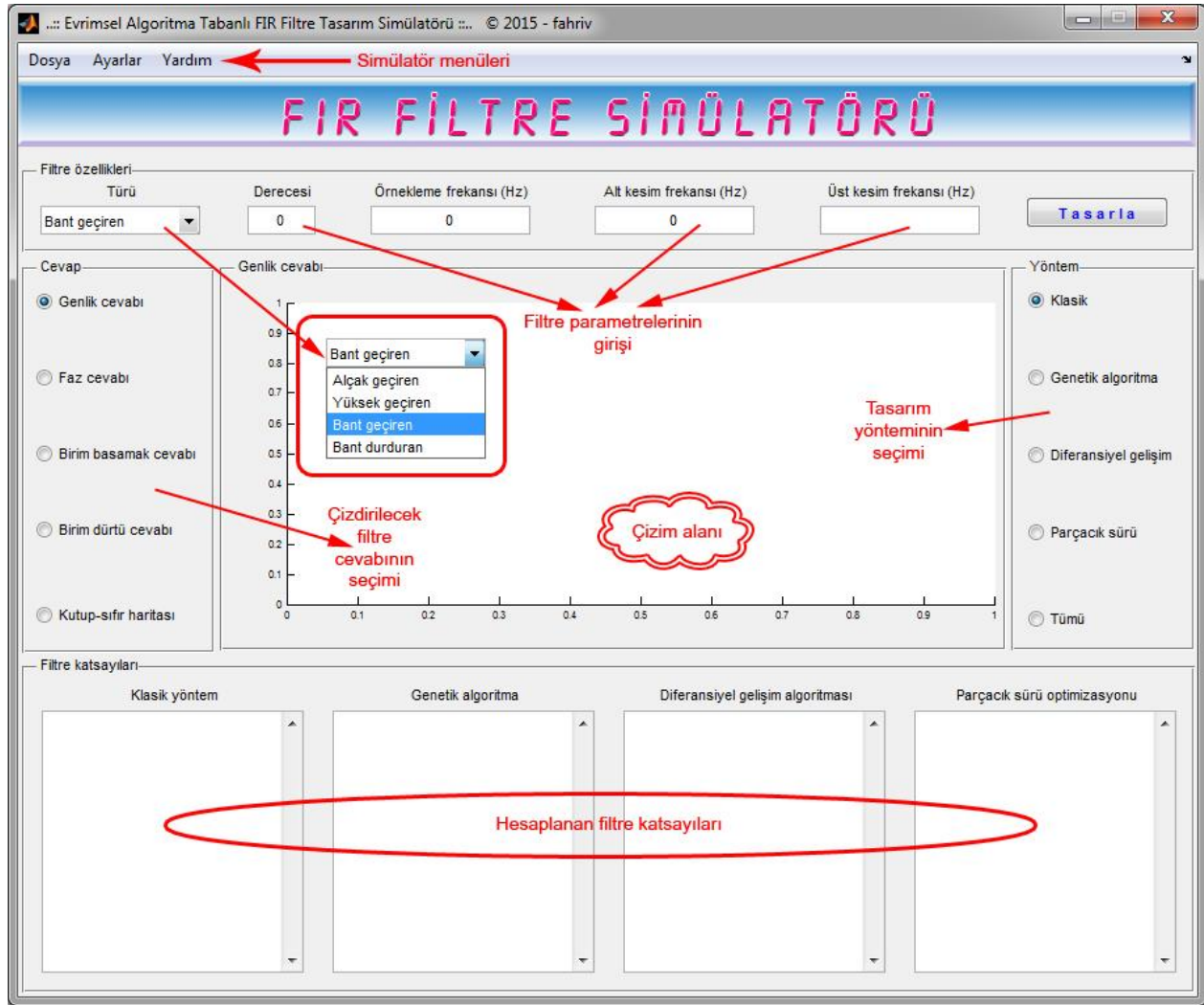


Şekil 5. PSO'daki genel işleyiş adımları

4. Tasarlanan Simülâtör ve Uygulamaları

MATLAB[20] kullanılarak geliştirilen simülâtöre ait ana ekran görüntüsü Şekil 6'da verilmektedir. Tasarlanacak filtre türü (alçak geçiren, yüksek geçiren, bant geçiren, bant durduran) seçilip parametreleri (derecesi, kesim veya alt/üst kesim frekansı ile örnekleme frekansı) girildikten sonra simülâtör dört yöntemle (klasik, GA, DE, PSO) katsayıları hesaplamaktadır. Ayrıca seçilen yönteme ve cevap türüne (genlik, faz, birim basamak, birim dürtü cevapları ve sıfır-kutup haritası) göre çizimler bireysel veya karşılaştırmalı olarak

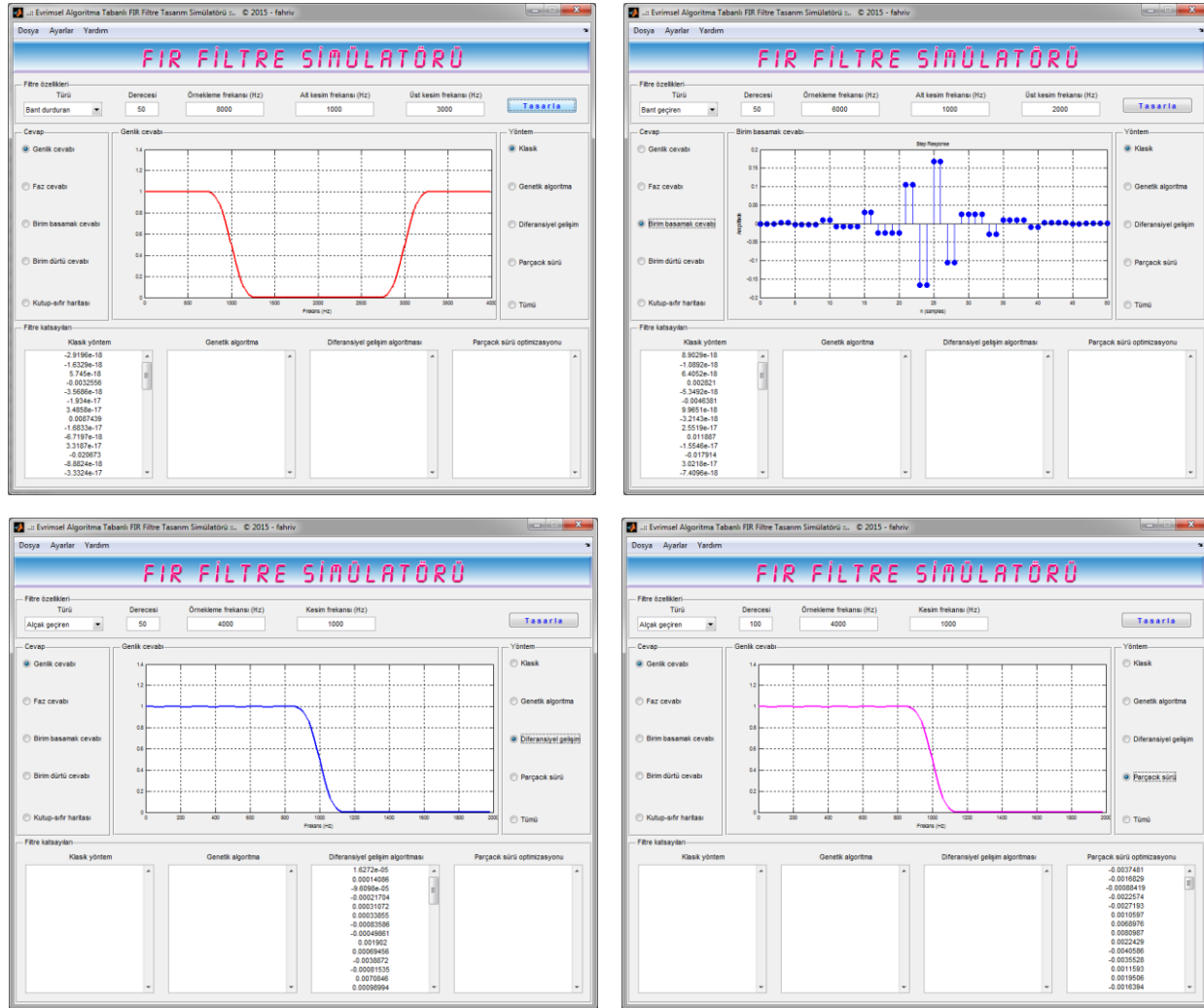
gerçekleştirilebilmektedir. Bunun yanında simülatör menüleriyle de sonuçlar değişik biçimlerde kaydedilebilmekte veya yazdırılabilmektedir. Menü ve seçeneklerinin görevleri Tablo 2'de özetlenmektedir. Şekil 7'de ise bazı örnek simülasyon ekran görüntüleri verilmektedir.



Şekil 6. Tasarlanan simülatörün ana ekranı

Tablo 2. Tasarlanan simülatörün menüleri ve görevleri

Menü	Ekran görüntüsü	Görevi
Dosya	Dosya Ayarlar Ya Yeni Ctrl+Y Kaydet Ctrl+K Yazdır Ctrl+Z Çıkış Ctrl+9	- Yeni simülasyonlar gerçekleştirme - Simülasyon sonuçlarını farklı biçimlerde kaydetme - Simülasyon sonuçlarını yazdırma - Programdan çıkış
Ayarlar	Ayarlar Yardım Genetik algoritma Ctrl+G Diferansiyel gelişim algoritması Ctrl+D Parçacık sürü optimizasyon algoritması Ctrl+P	- Genetik algoritma parametrelerini ayarlama - Diferansiyel gelişim algoritması parametrelerini ayarlama - Parçacık sürü optimizasyon algoritması parametrelerini ayarlama
Yardım	Yardım Yardım konuları Hakkında	- Yardım konularını inceleme - Simülatör hakkında bilgilere erişme



Şekil 7. Bazı örnek simülasyon ekran görüntüleri

5. Sonuçlar

Gerçekleştirilen çalışmada; hem klasik hem de evrimsel algoritmalarından genetik algoritma, diferansiyel gelişim algoritması ve parçacık sürü optimizasyonu algoritmasıyla FIR filtre tasarımları yapan simülör tasarlanmıştır. Eğitim amaçlı da kullanılabilecek simülör ile FIR filtrelerin tasarımları ve analizleri kolay ve etkin bir şekilde gerçekleştirilebilmektedir. Ayrıca evrimsel algoritmaların bu alandaki performansları, karşılaştırmalı analizlerle gözlemlenebilmektedir.

Kaynaklar

- [1] Williams T, Miller WC. Genetic algorithms for the design of digital filters. Proceedings of IEEE ICSP 2004;4:9–12.

- [2] Boudjelaba K, Ros F, Chikouche D. An efficient hybrid genetic algorithm to design finite impulse response filters. *Expert Systems with Applications* 2014;41:5917-5937.
- [3] Karaboga N, Cetinkaya B. Performance comparison of genetic and differential evolution algorithms for digital FIR filter design. *Lecture Notes in Computer Science* 2005;3261:482-488.
- [4] Storn R. Differential evolution design of an IIR-filter. *Proceedings of IEEE International Conference on Evolutionary Computation* 1996;268 – 273.
- [5] Karaboga N, Cetinkaya B. Design of digital FIR filters using differential evolution algorithm. *Circuits System Signal Processing* 2006;25:649-660.
- [6] Karaboga N, Cetinkaya B. Efficient design of fixed point digital FIR filters by using differential evolution algorithm . *Lecture Notes in Computer Science* 2005;3512:812-819.
- [7] Ababneh JI, Bataineh MH. Linear phase FIR filter design using particle swarm optimization and genetic algorithms. *Digital Signal Processing* 2008;18:657-668.
- [8] Singh R, Arya SK. Optimization of IIR digital filters using particle swarm optimization. *International Conference on Communication, Information & Computing Technology* 2012;1-7.
- [9] Najjarzadeh M, Ayatollahi A. FIR digital filters design: Particle swarm optimization utilizing LMS and minimax strategies. *IEEE International Symposium on Signal Processing and Information Technology* 2008;129-132.
- [10] Winder S. *Analog and digital filter design*. 2nd ed. USA:Newnes;2002.
- [11] Chen W-K(Ed.). *The circuits and filters handbook*. 2nd ed. USA:CRC Press;2003.
- [12] Ingle VK, Proakis JG. *Digital signal processing using MATLAB, v.4*. Boston:PWS Publishing Company;1997.
- [13] Thede L. *Practical analog and digital filter design*. Boston:Artech House Publication;2004.
- [14] Holland JH. *Adaption in natural and artificial systems*. Cambridge:MIT Press;1975.
- [15] Goldberg DE. *Genetic algorithms in search optimization and machine learning*. Boston:Addison Wesley;1989.
- [16] Simon D. *Evolutionary optimization algorithms*. New Jersey:John Wiley & Sons;2013.
- [17] Storn R, Price K. *Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report 1995:TR-95-012.
- [18] Qing A. *Differential evolution: Fundamentals and applications in electrical Engineering*. Singapore:Wiley-IEEE Press;2009.
- [19] Kennedy J, Eberhart RC. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks* 1995;4:1942-1948.
- [20] MATLAB. The MathWorks Inc., 2007.