

Güncel Evrimsel Algoritmalarla IIR Filtre Tasarımları

Design of IIR Digital Filters Using The Current Evolutionary Algorithm

Yiğit Çağatay Kuyu ve *Fahri Vatansever

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

Özet

Filtreleme uygulamalarında sonsuz dürtü cevaplı (IIR) filtrelerle elde edilen performanslar, bunlara göre daha yüksek dereceden sonlu dürtü cevaplı (FIR) filtrelerle sağlanabilmektedir. Ancak, IIR filtre tasarım problemleri, çoklu yerel minimum noktalarına sahip olması ve bu filtrelerin kompleks yapıları sebebiyle, geleneksel metotlar ve klasik algoritmalar ile çözülmesi oldukça zordur. Gerçekleştirilen çalışmada, Chebyshev Tip I ve Tip II IIR filtrelerin tasarımları, güncel algoritmalarından olan geri-izleme arama optimizasyon (BS), elektromanyetik alan optimizasyon (EFO) ve girdap arama (VS) algoritmaları ile yapılmış ve klasik tasarım yöntemleriyle karşılaştırılmıştır. Algoritmaların performansları; en iyi ile ortalama çözüm ve yakınsama hızı açısından birbiriyle kıyaslanmakta, ayrıca kullanılan kriterlere bağlı olarak hangi algoritmanın hangi tasarım problemi üzerinde en iyi performansı sergileyeceği hakkında kapsamlı bir analiz sunulmaktadır.

Anahtar kelimeler: Evrimsel algoritma, IIR filtre, Chebyshev filtre.

Abstract

In the filtering applications, the high-order FIR filters can show similar performances as compared to performances achieved by the low-order IIR filters. However, IIR filter design problems are difficult to solve for conventional methods and classical algorithms due to having multiple local optimum points and complex structures of these filters. In this work, the current algorithms, which are backtracking search optimization (BS), electromagnetic field optimization (EFO) and vortex search (VS) algorithms, have been implemented and compared to the classical design methods for the designs of Chebyshev type I and type II IIR filters. The performances of the algorithms have been compared to each other in terms of the best solution, the average solution and the convergence speed. This work also provides a comprehensive review about which the algorithm could perform best on which the design problem depend on the criteria used.

Key words: Evolutionary algorithm, IIR filter, Chebyshev filter.

1. Giriş

Günümüzde teknolojinin ilerlemesiyle sayısal filtreler, sayısal sinyal işlemede sıklıkla kullanılan yapılar arasında yer almaktadır. Sayısal filtreler temel olarak iki grupta sınıflandırılabilirler: sonlu darbe cevaplı (FIR) ve sonsuz darbe cevaplı (IIR) filtreler. Sayısal sinyal işleme uygulamalarında, IIR filtreler, FIR filtrelerden daha başarılı performans verebilmektedirler[1,2]. Diğer yandan, FIR filtreler genellikle IIR filtrelerin yakaladığı

*Corresponding author: Address: Faculty of Engineering, Department of Electrical-Electronics Engineering, Uludag University, 16059, Bursa TURKEY. E-mail address: fahriv@uludag.edu.tr, Phone: +902242940905

performansı daha yüksek dereceden filtre tasarımlarlarında yakalayabilmektedirler. Buna rağmen, IIR filtrelerin çok modlu ve lineer olmayan hata yüzeyine sahip olması, IIR filtre tasarımlarını zorlaştırmaktadır [3,4]. Bundan dolayı daha güçlü tasarım tekniklerine ihtiyaç duyulmaktadır.

Günümüzde, evrimsel algoritmalar optimizasyon problemlerinde öne çıkan güçlü tekniklerin başında gelmektedir [5,6]. Genel olarak evrimsel algoritmalar, olası çözüm kümesini rastsal olarak oluşturmakta ve ürettiği olası çözümlerin uygunluk fonksiyonu değerlerini hesaplamaktadırlar. Daha sonra kendilerine özgü hesaplama teknikleri kullanarak yeni bir çözüm kümesi üretmektedirler. Oluşturdukları yeni çözüm kümesi değerlerinin uygunluk fonksiyonu değerlerini kullanarak çözümlerini geliştirmekte ve ön tanımlı durdurma kriterine ulaştıklarında işlemlerini sonlandırmaktadırlar. IIR filtre tasarımlarında, literatürde, temel evrimsel algoritmalar olan genetik algoritma (GA) [7], parçacık sürü optimizasyon algoritması (PSO) [8] ve diferansiyel gelişim algoritması (DE) [9] ile birçok çalışma mevcuttur. GA, sıklıkla sayısal IIR filtre kullanılmaktadır [10-12]. Çalışma [13, 14]' te PSO, sayısal IIR filtre tasarımlarına uygulanmakta ve GA'dan birçok tasarımda daha iyi performans gösterdiği gözlemlenmektedir. Storn, çalışmasında [15], DE'nin IIR filtre tasarımına verimli bir şekilde uygulanabileceğini göstermektedir. Diğer yandan, bu algoritmaların güçlü yapılarına rağmen, bazı rastsal işlem adımlarına sahip olmaları, global çözümlere ulaşmalarını daima sağlayamamaktadır. Bu yeni ve güncel algoritmalarla ihtiyaca yol açmaktadır.

Bu çalışmada, güncel evrimsel algoritmalar olan, elektromanyetik alan optimizasyon algoritması (EFO) [16], geri-izleme arama optimizasyon algoritması (BS) [17], girdap arama algoritması (VS) [18] kullanılarak, Chebyshev Tip I ve Tip II filtre tasarımları yapılmaktadır. Her bir tasarım problemi için algoritmaların bulduğu katsayılar tablolar halinde verilmekte, hata değerleri ve ortalama işlem süreleri analiz edilmektedir. Bunun yanında, buldukları en iyi çözümlere yakınsama grafikleri ve frekans cevabı grafikleri verilerek algoritmaların örnek tasarımlar üzerinde davranışları detaylı olarak incelenmektedir.

2. Problemin Tanımı

Sayısal IIR filtreler

$$y(k) + \sum_{j=1}^N b_j y(k-j) = \sum_{j=0}^M a_j x(k-j) \quad (1)$$

fark denkleminde tanımlanabilmektedirler[19]. Burada a ve b , filtre katsayılarını; $N (\geq M)$, filtrenin derecesini belirtmektedir. Filtrenin genel transfer fonksiyonu Denklem 2'deki gibi yazılabilmektedir.

$$H(z) = \frac{A(z)}{1+B(z)} = \frac{\sum_{j=0}^L a_j z^{-j}}{1+\sum_{j=1}^M b_j z^{-j}} \quad (2)$$

Pay, $\{a_j\}_{j=0}^L$ ve payda $\{b_j\}_{j=1}^M$ katsayılarından oluşan katsayılar vektörü Denklem 3'teki gibi tanımlanabilmektedir.

$$K = [a_0, \dots, a_L, b_1, \dots, b_M] \quad (3)$$

Ortalama karesel hata fonksiyonu aracılığıyla, minimize edilecek hedef fonksiyon $J(K)$, Denklem 4'te verilmektedir.

$$J(K) = \frac{1}{L} \sum_{k=1}^L e(k)^2 \quad (4)$$

$e(k) = d(k) - y(k)$ ile ifade edilmekte; $d(k)$; filtrenin istenen cevabını, L ; örnek sayısını, J ; minimize edeceğimiz hata fonksiyonunu temsil etmektedir.

3. Evrimsel Algoritmalar

3.1. Elektromanyetik Alan Optimizasyon Algoritması

Elektromanyetik alan optimizasyon algoritması (EFO) [16], elektromanyetik alandaki pozitif ve negatif parçacıkların (elektromagnet) arasındaki itme ve çekme kuvvetlerinden esinlenerek ortaya atılmıştır. EFO, popülasyon tabanlı bir algoritma olup, çözüm vektörü elektromagnetlerin bir grubunu temsil etmektedir (elektromanyetik parçacık). Çözüm vektöründeki her bir değişken ise elektromanyetik parçacıklardaki elektromagnetlerle ifade edilmektedir. Algoritmanın işlem adımları aşağıdaki gibi özetlenebilmektedir:

- i. Elektromanyetik parçacık popülasyonunu rastgele olarak üret ve her bir parçacığın uygunluk değerini hesapla.
- ii. Parçacıkları, uygunluk değerine göre üç alana ayır. Bunlar; en iyi uygunluk değerine sahip parçacıklar için pozitif alan; en kötü uygunluk değerine sahip parçacıklar için negatif alan; kalan parçacıklar için nötr alan.
- iii. Pozitif alandaki parçacıkları pozitif polarite, negatif alandaki parçacıkları negatif polarite ve nötr alandaki parçacıkları sıfıra yakın negatif polariteyle gruplandır.
- iv. Parçacıkların yeni pozisyonlarını Denklem 5'e göre hesapla.

$$EMP_j^{New} = EMP_j^{Kj} + ((\alpha \cdot r) \cdot (EMP_j^{Pj} - EMP_j^{Kj})) - (r \cdot (EMP_j^{Nj} - EMP_j^{Kj})) \quad (5)$$

Denklem 5'te EMP elektromanyetik parçacığı, r , $[0,1]$ arası üretilen rastgele değeri, j değişken indeksini, K nötr alandaki, P pozitif alandaki ve N negatif alandaki rassal indeksleri belirtmektedir.

- v. Üretilen yeni parçacık, popülasyondaki en kötü parçacıktan daha iyi ise, kötü parçacığı popülasyondan çıkart ve bulunan parçacığı uygun polariteyle gruplandır. İşlemlere durdurma kriteri sağlanıncaya kadar devam et.

3.2. Geri-izleme Arama Optimizasyon Algoritması

Geri-izleme arama optimizasyon algoritması, Civicoglu tarafından geliştirilen, işlem adımları; başlangıç, seleksiyon-1, mutasyon, çaprazlama ve seleksiyon-2 olarak beş temel yapıdan oluşan

popülasyon tabanlı, iteratif bir algoritmadır [17]. Başlangıç adımında Denklem 6 kullanılarak olası çözüm kümesi oluşturulmaktadır.

$$Pop_{i,j} \sim U(\text{alt sınır}_j, \text{üst sınır}_j) \quad i = 1, 2, \dots, N \text{ ve } j = 1, 2, \dots, D \quad (6)$$

altsınır ve üstsınır problemin alt ve üst sınırlarını, N ve D sırasıyla popülasyon sayısı ve problemin boyutunu ifade etmektedir. Bu algoritma, hafızada tutacağı bir eski popülasyon ($oldPop$) üreterek, bunu Denklem 7’de verilen mutasyon işleminde kullanmaktadır.

$$Mutant = P + F(oldP - P) \quad (7)$$

P , güncel popülasyon; F , $(oldP - P)$ arama doğrultu matrisinin büyüklüğünü kontrol etmektedir. Çaprazlama işlemi aracılığıyla deneme popülasyonu (T) oluşturulmaktadır. Bu popülasyonun başlangıç değeri $Mutant$ olarak alınmaktadır. Seleksiyon-2 aşamasına geçildiğinde, P_i ve T_i değerleri, P_i ’leri güncellemek amacıyla kullanılmaktadır. Eğer P ’nin en iyi değeri P_{best} , algoritmanın o ana kadarki bulduğu global minimum değerinden daha iyiye, en iyi çözüm değeri P_{best} olarak güncellenmektedir. Bu işlem adımları durdurma kriteri sağlanıncaya kadar devam etmektedir[17].

3.3. Girdap Arama Algoritması

Akışkanların oluşturduğu girdap örüntüsünden esinlenerek oluşturulan girdap arama algoritması (VS) [18] adaptif adım küçültme yöntemi kullanarak olası çözümlere yaklaşmaktadır. 2 boyutlu bir optimizasyon problemi ele alındığında, ilgili problem algoritma tarafından iç içe geçmiş çemberler olarak modellenenmektedir. Çemberin merkezi Denklem 8’deki gibi hesaplanmaktadır.

$$\mu_0 = \frac{\text{üst sınır} + \text{alt sınır}}{2} \quad (8)$$

üst sınır ve alt sınır problemin üst ve alt limitlerini, μ_0 ise en dış girdap çemberinin merkezini temsil etmektedir. Başlangıçta rassal olarak Gauss dağılımı kullanarak μ_0 merkezi etrafında $C_t(s)$ komşu çözümleri üretilir. t iterasyon indeksinde, $t = 0$ başlangıç konumunda $C_0(s) = \{s_1, s_2 \dots s_n\}$ $n = 1, 2, \dots, M$ başlangıç olası çözüm kümesi elde edilir. Bu çözüm kümesinde, çözümlerin problemin sınırları içerisinde bulunduğuna emin olmak amacıyla VS algoritması sınır kontrolü yapmaktadır. Daha sonra, çemberin başlangıç merkezi μ_0 ‘ı güncellemek için en iyi çözüm $s' = C_0(s)$ uygunluk fonksiyonu değerlerine göre saptanır ve s_{best} olarak kaydedilir. s_{best} bir sonraki iterasyonda yeni merkez olarak atanır ve bu yeni merkez etrafında Gauss dağılımı kullanılarak içteki çemberin yarıçapı azaltılır. Böylece algoritmanın yerel arama gücü, azaltılan yarıçap ile artmış olmaktadır. Algoritma yukarıda belirtilen süreçleri ön tanımlı durdurma kriteri sağlanıncaya kadar devam etmektedir[18].

4. Uygulamalar

Çalışmada uygulama örnekleri olarak Chebyshev Tip I alçak geçiren ve Chebyshev Tip II yüksek geçiren filtrelerin BS, EFO ve VS algoritmalarıyla tasarımları verilmektedir. Filtrelere ilişkin başlangıç şartları, Tablo 1’de gösterilmektedir. Eşit kıyaslama açısından, tüm algoritmalar için, başlangıç çözüm sayıları 80 ve maksimum fonksiyon hesaplama sayıları 100.000 olarak alınmıştır. Algoritmaların optimal parametre değerleri literatürdeki makalelerden alınmaktadır [16-18]. Her bir algoritma 10 bağımsız çalıştırmaya tabi tutulmakta, algoritmaların buldukları en iyi hata, ortalama hata, standart sapma, ortalama çalıştırılma zamanı ve -3dB noktasındaki kesim frekansları verilmektedir. Ayrıca algoritmaların buldukları en düşük hataya göre yakınsama grafikleri karşılaştırmalı olarak çizdirilmektedir.

Tablo 1. Örnek tasarımlara ilişkin özellikler/parametreler

Özellik/Parametre	Örnek tasarım 1	Örnek tasarım 2
Fitre türü	Chebyshev Tip I	Chebyshev Tip II
Fitre tipi	Alçak geçiren filtre	Yüksek geçiren filtre
Örnekleme frekansı	10 kHz	10 kHz
Geçirme bandı kesim frekansı	2 kHz	2.2 kHz
Durdurma bandı kesim frekansı	2.2 kHz	2 kHz
Geçirme bandı dalgalanması	0.1 dB	0.1 dB
Durdurma bandı zayıflaması	30 dB	30 dB

4.1. Alçak Geçiren Filtre Tasarımı

İlgili algoritmalar kullanılarak tasarlanan Chebyshev Tip I alçak geçiren filtrenin katsayıları Tablo 2’de; tasarlanan filtreye ilişkin hata, ortalama işlem zamanı ve kesim frekansı değerleri Tablo 3’te verilmektedir. Algoritmaların yakınsama grafiği Şekil 1’de, frekans cevabı grafiği Şekil 2’de gösterilmektedir.

Tablo 2. Örnek IIR filtre tasarımı 1 için pay ve payda katsayıları

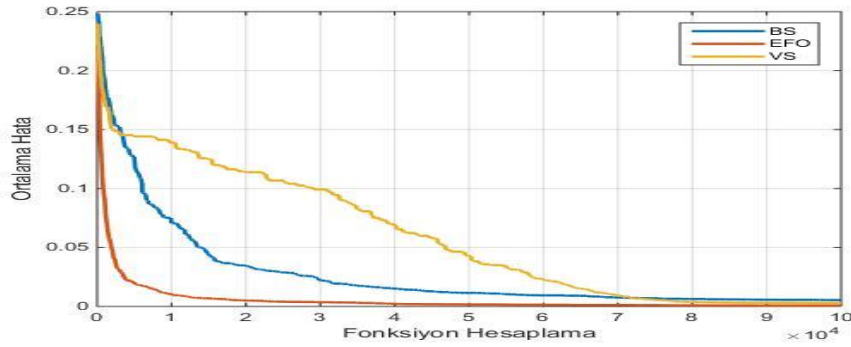
No.	Pay Katsayıları			Payda Katsayıları		
	BS	EFO	VS	BS	EFO	VS
1	4.9264e-5	5.0541e-5	2.3401e-5	1.0000	1.0000	1.0000
2	5.0604e-4	7.220e-5	0.0005	-5.9492	-5.8447	-5.6961
3	0.0004	0.0004	0.0027	18.2526	18.1474	18.0435
4	0.0010	0.0010	0.0015	-38.4594	-38.6214	-38.4022
5	0.0033	0.0299	0.0252	60.7322	60.9535	60.7837
6	0.0500	0.0499	0.0418	-74.1295	-74.2712	-74.1026
7	0.0500	0.0403	0.0448	71.3614	71.3537	71.3494
8	0.0500	0.0499	0.0433	-54.2153	-54.1824	-54.5306
9	0.0030	0.0299	0.0276	32.5348	32.6495	32.9495
10	0.0010	0.0015	0.0067	-15.2725	-15.3909	-15.4815
11	0.0004	0.0004	0.0031	5.5655	5.6628	5.3778
12	0.0007	7.3249e-5	0.0003	-1.5077	-1.5142	-1.2405
13	4.6435e-5	4.5763e-5	4.3039e-5	0.2544	0.2694	0.1515

Tablo 3’ten görülebileceği üzere, EFO algoritması kıyaslanan algoritmalar arasından, en iyi ve ortalama hata değerleri açısından en başarılı performansı göstermektedir. Çalıştırılma zamanı kıyaslandığında ise, bu algoritma en yavaş algoritma olarak göze çarpmaktadır. BS algoritması aynı örnek tasarım için 68 sn ile en hızlı zaman süresinde ilgili filtreyi tasarlamakta fakat bu

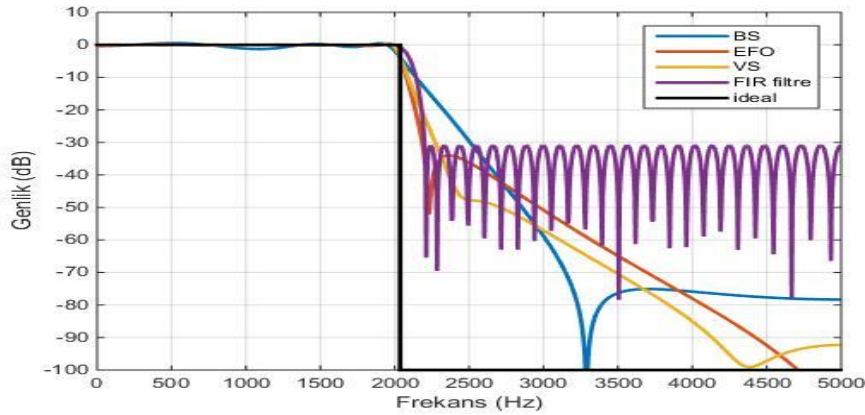
algoritma ulaştığı hata değerleri açısından en kötü performansı sergileyen algoritma olmaktadır. VS algoritması ise 72 sn ile BS algoritmasına yakın değerlerde filtreyi tasarlayabilmekte bulunduğu 0.0029 ortalama hata ile BS'den daha iyi bir performans göstermektedir. Şekil 1'den görülebileceği üzere, EFO algoritması kendi çözümlerine diğer algoritmalarından daha hızlı bir yakınsama göstermektedir ve ulaştığı çözüm algoritmalar arasındaki en iyi çözüm olmaktadır. Diğer yandan, Şekil 2'de, EFO algoritmasının durdurma bandında başarılı bir şekilde sönümleme yapabildiği görülebilmektedir.

Tablo 3. Örnek IIR filtre tasarımı 1 için algoritmalar tarafından 10 bağımsız çalıştırmada bulunan değerler

	BS	EFO	VS
Minimum en iyi hata	0.0040	0.0002	0.0004
Ortalama hata	0.0053	0.0007	0.0029
Standart Sapma	0.0017	0.0003	0.0011
Ortalama zaman (sn)	68	105	72
Kesim frekansları (Hz)	2021	2031	2026



Şekil 1. Örnek IIR filtre tasarımı 1 için algoritmaların yakınsama grafiği



Şekil 2. Örnek IIR filtre tasarımı 1 için frekans cevabı grafiği

4.2. Yüksek Geçiren Filtre Tasarımı

İlgili algoritmalar kullanılarak tasarlanan Chebyshev Tip II yüksek geçiren filtrenin katsayıları Tablo 4'te; tasarlanan filtreye ilişkin hata, ortalama işlem zamanı ve kesim frekansı değerleri Tablo 5'te verilmektedir. Algoritmaların yakınsama grafiği Şekil 3'te ve frekans cevabı grafiği Şekil 4'te gösterilmektedir. Bulunan hatalar ve ortalama zaman açısından, Tablo 5'e bakılacak

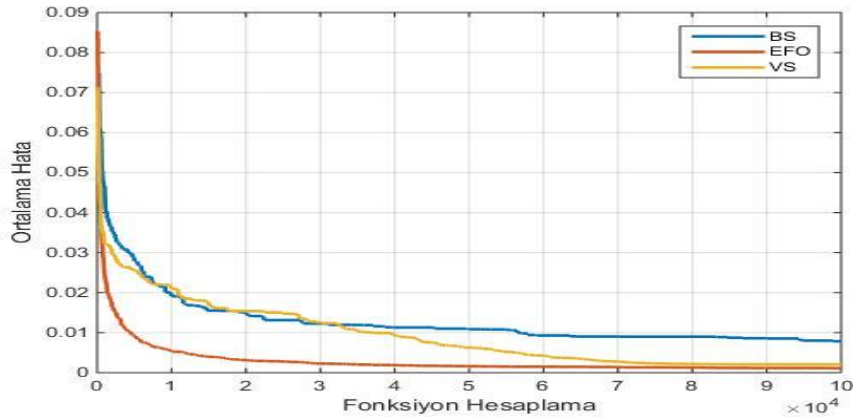
olursa, EFO algoritması alçak geçiren filtre tasarımı gösterdiği performansa paralel olarak, ortalama ve en iyi hatalar açısından kıyaslanan algoritmalarından daha iyi sonuçlara ulaşmaktadır fakat bu algoritma zaman açısından yüksek geçiren filtreyi en yavaş şekilde tasarlamaktadır. Diğer yandan, BS algoritması alçak geçiren filtre tasarımı olduğu gibi yüksek hatayla filtre tasarımı yaptığından her iki örnek tasarımı için diğer algoritmalarla kıyaslanınca tercih edilebilir bir algoritma olmamaktadır. Bunun yanında hız açısından algoritmalar kıyaslandığında, BS algoritmasının en hızlı tasarımları yapan algoritma olduğu görülmektedir. Şekil 3'te algoritmaların yakınsama grafiklerine bakılacak olursa, EFO algoritmasının yüksek yakınsama hızı gösterdiği ve diğer algoritmalarından daha önce kendi çözümüne yakınsadığı görülmektedir. Algoritmaların frekans cevapları grafikleri Şekil 4' ten görülebileceği üzere, BS ve EFO algoritmaları, filtreyi tasarladıkları düşük hata farklarından dolayı birbirine yakın frekans cevabı grafiği vermektedir. Aynı grafikten klasik FIR filtre tasarımının algoritmalarla nazaran durdurma bandında daha iyi bastırma yaptığı söylenebilmektedir. Bunun sebebi, aynı tasarım değişkenlerinde, evrimsel algoritmaların IIR filtre tasarımını 12. dereceden yapabilirken, FIR filtre bu tasarıma 90. dereceden ulaşabilmesi olarak gösterilebilmektedir.

Tablo 4. Örnek IIR filtre tasarımı 2 için pay ve payda katsayıları

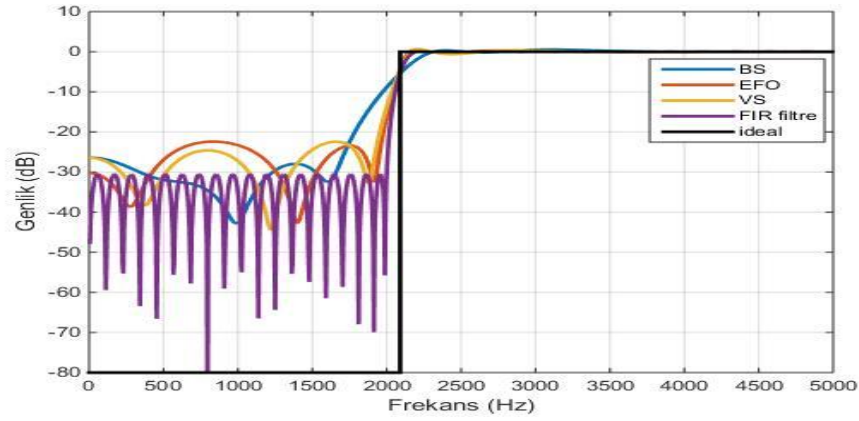
No.	Pay Katsayıları			Payda Katsayıları		
	BS	EFO	VS	BS	EFO	VS
1	0.1430	0.1990	0.1699	1.0000	1.0000	1.0000
2	-1.4951	-1.4666	-1.4214	-5.0000	-4.9334	-4.9935
3	5.4460	5.2037	5.2358	13.0000	13.0073	13.0201
4	-13.2830	-13.0450	-13.1356	-26.0000	-25.8997	-25.9076
5	24.5777	24.4029	24.5302	35.5000	35.6741	35.6428
6	-35.5000	-35.1641	-35.2866	-40.0000	-39.9528	-39.8966
7	40.0000	39.6305	39.7458	34.5000	34.5130	34.6019
8	-35.2103	-35.0748	-35.2313	-23.5364	-23.6589	-23.8295
9	24.2639	24.4029	24.5302	13.5000	13.3388	13.4396
10	-13.0813	-13.1394	-13.2452	-5.9628	-5.5478	-5.7236
11	5.3705	5.3123	5.3619	1.5947	1.6628	1.6565
12	-1.5000	-1.4816	-1.4799	-0.1000	-0.3902	-0.3946
13	0.1975	0.1843	0.1643	0.0240	0.0234	0.0879

Tablo 5. Örnek IIR filtre tasarımı 2 için algoritmalar tarafından 10 bağımsız çalıştırmada bulunan değerler

	BS	EFO	VS
Minimum en iyi hata	0.0048	0.0008	0.0012
Ortalama hata	0.0078	0.0011	0.0020
Standart Sapma	0.0025	0.00025	0.0005
Ortalama zaman (sn)	77	229	149
Kesim frekansları (Hz)	2163	2104	2109



Şekil 3. Örnek IIR filtre tasarımı 2 için algoritmaların yakınsama grafiği



Şekil 4. Örnek IIR filtre tasarımı 2 için frekans cevabı grafiği

5. Sonuçlar

Bu çalışmada, güncel evrimsel algoritmalar olan BS, EFO ve VS kullanılarak Chebyshev Tip I ve Tip II filtreler tasarımları gerçekleştirilmiştir. Algoritmaların performansları; hatalar, yakınsama hızları ve işlem zamanı açısından analiz edilmiştir. Elde edilen sonuçlara göre, EFO algoritması her iki filtre tasarımı için hem ortalama hata hem de en iyi hata açısından en başarılı performansı göstermektedir. Fakat bu algoritma işlem zamanı açısından en yüksek zamana sahip olmaktadır. Diğer yandan, BS algoritması işlem zamanı açısından en hızlı algoritma olmasına rağmen, yüksek hatalar ile filtreleri tasarlamaktadır. Kullanılan algoritmaların seçilen kriterin önceliğine bağlı olarak (yakınsama hızı, işlem zamanı, hata) filtre tasarımlarında kolay ve etkin bir şekilde tercih edilebileceği gözlenmiştir.

Kaynaklar

[1] Proakis JG, Manolakis DG. Digital signal processing: principles, algorithms and applications. 4th ed. New Jersey: Pearson Prentice Hall; 2007.

- [2] Ingle V, Proakis JG. Digital signal processing using MATLAB. Cengage Learning; 2011.
- [3] Chauhan RS, Arya SK. An application of swarm intelligence for the design of IIR digital filters. *International Journal of Swarm Intelligence* 2013; 1: 3-18.
- [4] Antoniou A. Digital Filters: analysis, design and applications. New Delhi: McGraw Hill; 2005.
- [5] Michalewicz Z, Hinterding R. Evolutionary algorithms. In: Pedrycz W, editor. *Fuzzy evolutionary computation*, USA: Kluwer Academic; 1997, p. 3-31.
- [6] Yang XS. Efficiency analysis of swarm intelligence and randomization techniques. *Journal of Computational and Theoretical Nanoscience* 2012; 9: 189-198.
- [7] Holland, J. *Adaptation in natural and artificial systems*. USA: MIT Press; 1975.
- [8] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks* 1995; 1942-1948.
- [9] Storn R, Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *ICSI Technical Report TR-95-012* 1995.
- [10] Li JH, Yin FL. Genetic optimization algorithm for designing IIR digital filters. *Journal of China Institute of Communications* 1996; 17: 1-7.
- [11] Renders JM, Flasse SP. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part B* 1996; 26: 243-258.
- [12] Kaur R, Patterh MS, Dhillon JS. Digital IIR filter design using real coded genetic algorithm. *Information Technology and Computer Science* 2013; 27-35.
- [13] Krusienski DJ, Jenkins WK. Adaptive filtering via particle swarm optimization. In: *Proceedings of the Conference Record of the Asilomar Conference on Signals Systems and Computers* 2003; 571-575.
- [14] Gao Y, Li Y, Qian H. The design of IIR digital filter based on chaos particle swarm optimization algorithm. In: *Proceedings of the 2nd International Conference on Genetic and Evolutionary Computing* 2008; 303-306.
- [15] Storn R. Differential evolution design of an IIR-filter. In: *Proceedings of IEEE International Conference on Evolutionary Computation* 1996; 268-273.
- [16] Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DNA. Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation* 2016; 26: 8-22.
- [17] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation* 2013; 219: 8121-8144.
- [18] Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: vortex search algorithm. *Information Sciences* 2015; 293: 125-145.
- [19] Ifeachor EC, Jervis BW. *Digital signal processing, a practical approach*. UK: Pearson Education; 2002.