

# DİFERANSİYEL GELİŞİM ALGORİTMASI KULLANILARAK ADAPTİF LİNEER TOPLAYICI TASARIMI

Nurhan KARABOĞA<sup>1</sup>

Canan Aslıhan KOYUNCU<sup>2</sup>

Elektrik Elektronik Mühendisliği Bölümü  
Mühendislik Fakültesi  
Erciyes Üniversitesi, 38090, Talas, Kayseri

e-posta<sup>1</sup>: nurhan\_k@erciyes.edu.tr

e-posta<sup>2</sup>: 1030215054@erciyes.edu.tr

*Anahtar sözcükler: Diferansiyel Gelişim Algoritması, Adaptif Lineer Toplayıcı*

## ÖZET

*This paper has been oriented to bring new theoretical and practical experience with using of new type optimization algorithm in the filters design. It presents the application of Differential Evolution (DE), an Evolutionary Computation method, for the optimization of adaptive FIR filter weights. The new method requires few control variables, is robust and easy to use. While obtaining the noisy desired signal, to optimize the Adaptive Linear Combiner's weights, different error functions are used. These error functions are Mean Square Error function, Mean Absolute Error function and Least Mean Squared Error function. Our purpose is comparing these functions' performances for our problem and determining which function is appropriate. The simulation is written in Matlab environment.*

## 1. GİRİŞ

Sayısal sistemler güvenilir ve ucuz olmalarının yanı sıra diğer bazı avantajlara da sahiptir. Sayısal sistemlerdeki işaretlerin, sadece belirli değerleri aldıkları için bozulmalardan etkilenmeleri de analog işaretlere göre daha azdır. Sayısal süzgeç ayırık zamanlı işaretleri filtrelemek için kullanılan sayısal bir sistemdir. Sayısal süzgeçler, işaretin örneklenmiş değerleri üzerinde nümerik hesaplamaları yapabilmek için sayısal işlemcileri kullanmaktadırlar. Süzgeç tasarımcıları, farklı bir çok süzgeç yapısını, örneğin, Sonlu Darbe Yanıtlı (SDY) ve Sonsuz Darbeli Yanıtlı (SzDY) süzgeçler, bunların doğrudan ya da kaskat şekilleri veya kafes yapıları vs., incelemektedirler. Çoğu süzgeç tasarım prosedüründe tasarımcı, önce bir süzgeç yapısını ardından süzgeç katsayılarını ve sonra da seçilen süzgecin ihtiyaçları yerine getirip getirmediğini belirlemektedir [1]. Bu prosedür eğer dikkatlice gerçekleştirilirse, tatmin edici sonuçlar alınabilir. Çoğu tasarım algoritmaları daha az karmaşık yapılar üzerinde yoğunlaştığı için seçilen süzgeç yapısı yine de optimalliği sağlamayabilir [2]. Giriş verisinin istatistiksel karakteristiği zamana bağlı olarak değişirse ya da giriş verisi hakkındaki gereken

bilgi yeterli olmazsa adaptif süzgeçlere ihtiyaç duyulmaktadır [3]. Adaptif SDY süzgeçlerinin, adaptif lineer toplayıcıların, tasarlanması için bir çok metod vardır. Bilgisayarlar, performanslarının gelişimi ile evrimsel algoritmalara dayanan optimizasyon metodlarının yeni bir sınıfını kullanarak bazı teknik problemlerin çözümüne izin vermektedir. Bu çalışmada, Diferansiyel Gelişim algoritması olarak adlandırılan algoritma, SDY süzgeç tasarımı çözümü için kullanılmıştır.

DG algoritması ilk olarak 1995 yılında K. Price tarafından ortaya konmuştur [4]. DG algoritması basit ama güçlü popülasyon tabanlı bir algoritmadır. Özellikle bütünüyle düzenlenmiş uzaya tanımlı ve gerçek değerli tasarı parametrelerini içeren fonksiyonları, küresel olarak optimize etmek amacıyla kullanılan bir direkt araştırma algoritmasıdır [5]. Bu çalışmada Adaptif Lineer Toplayıcı için, DG algoritmasına dayalı bir metod sunulmaktadır. Gürültü olduğunda, Adaptif Lineer Toplayıcının ağırlıkları adaptif olarak ayarlanmaktadır.

Bölüm 2, 3, 4'te Adaptif Lineer Toplayıcıların, DG Algoritmasının ve Hata Fonksiyonlarının kısa bir özeti verilmektedir. Bölüm 5'te Adaptif Lineer Toplayıcının tasarımı için DE algoritmasının nasıl uygulandığı gösterilmekte ve simülasyon sonuçları sunulmaktadır.

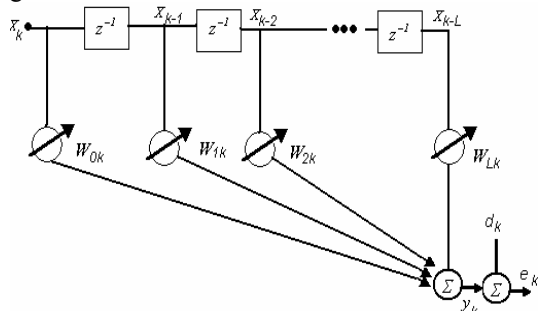
## 2. ADAPTİF LİNEER TOPLAYICILAR

Adaptif Lineer Toplayıcılar, otomatik ayar yapabilen sayısal süzgeçlerdir. Bu süzgeçler giriş işaretlerine göre değişebilirler. Adaptif Lineer Toplayıcılar, değişken işaret durumlarına karşılık süzgeç karakteristiklerinin değişimini gerektiren uygulamalarda kullanılırlar [6].

Adaptif Lineer Toplayıcılarda iki giriş gerekmektedir: işaret ve referans giriş. Bir adaptif süzgeç kendi katsayılarını, ağırlıklarını, yenileme kabiliyetine sahiptir. Yeni katsayılar bir katsayı üreticinden süzgece gönderilir. Katsayı üretici gelen bir sinyale

cevaben katsayıları değiştiren adaptif bir algoritmadır. Çoğu uygulamalarda katsayı üreticinin amacı süzgeç katsayılarını referans girişe eşleştirmektir, bundan dolayı adaptif süzgeç referans girişi işaretten çıkarılabilir. Referans giriş işareti değiştiğinden katsayılar da bunu yakalamak için değişmek zorundadır, bu çalışma prensibinden dolayı da adaptif süzgeç olarak adlandırılmaktadır.

Adaptif Linear Toplayıcılar, gürültü giderme, lineer kestirim, sistem modelleme, zaman gecikmesi kestirimi, parametre kestirimi ve adaptif kontrolü içeren uygulamaların çoğunda kullanılmaktadırlar. Aynı zamanda, adaptif süzgeçlemede de adaptif lineer toplayıcılar kullanılmaktadır. Şekil-1'de, SDY tek girişli bir adaptif lineer toplayıcının yapısı gösterilmektedir.



Şekil-1 Tek girişli bir adaptif lineer toplayıcı

SDY süzgeç, L+1 uzunluğundadır, katsayıları da  $w_k$  ile gösterilmektedir.  $w_k$  katsayılarının ayarlanması çevreye bağımlı olabilir ve katsayılar zamanla değişiyor olabilir. Şekil-1'de,  $x_k$  adaptif SDY süzgecinin giriş işareti,  $y_k$  çıkış işareti,  $d_k$  arzu edilen işaret ve  $e_k$  hata işaretidir.

### 3. DİFERANSİYEL GELİŞİM ALGORİTMASI

Diferansiyel gelişim algoritması küresel optimizasyon için basit, ama güçlü popülasyon tabanlı bir algoritmadır [7]. Aerodinamik şekillerin optimizasyonunda, IIR süzgeç tasarımında, Sınır ağlarının öğrenmesi gibi çoğu uygulamada dinçliği ve gücü gösterilmiştir [8-10]. DG'nin basit yapısı, kullanım kolaylığı, hızı ve dinçliği en önemli avantajları arasında yer almaktadır. DG'nin önemli parametreleri: NP (Number of Population-popülasyon büyüklüğü), CR (Crossover Rate-çaprazlama sabiti), F (Scaling Factor-ölçekleme faktörü) olarak sayılabilir. D parametresini içeren bir optimizasyon, D boyutlu bir vektör ile gösterilebilir. DG'de, NP adet çözüm vektörünün bir popülasyonu başlangıçta rasgele meydana getirilir. Bu popülasyon mutasyon, çaprazlama ve seleksiyon operatörleri uygulanarak başarılı bir şekilde geliştirilir [11-12]. DG algoritmasının temel adımları aşağıdaki gibi verilebilir:

*Başlangıç popülasyonunun oluşturulması*

*Değerlendirme*

**Repeat**

*Mutasyon*

*Rekombinasyon*

*Değerlendirme*

*Seleksiyon*

**Until** (*durdurma kriteri sağlanıncaya kadar*)

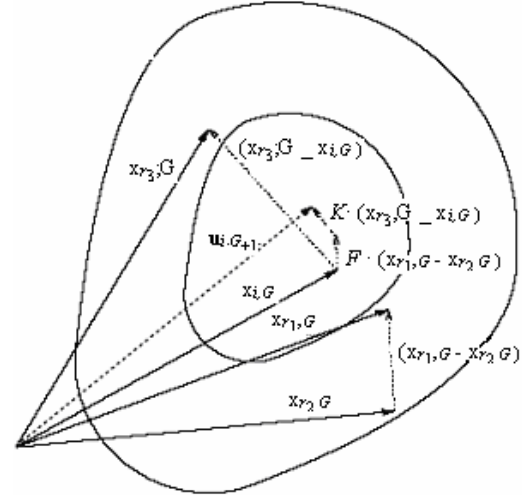
DG, diğer Evrimsel Algoritmalarından mutasyon ve rekombinasyon aşamalarında farklılık göstermektedir. DG popülasyonu karıştırmak için, çözüm vektörleri arasındaki ağırlıklandırılmış farkları kullanmaktadır.

$$u_{i,G+1} = x_{i,G} + K \cdot (x_{r3,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G})$$

Rasgele seçilen  $r_1; r_2; r_3 \in \{1, 2, \dots, n\}$ ;

$$r_1 \neq r_2 \neq r_3 \neq i \quad (1)$$

Bu çalışmada kullanılan DG, DG/current-to-rand/1 (Eşitlik 1) olarak bilinir ve rotasyonel olarak değişimsizdir (invariant) [13]. Bir Diferansiyel Gelişim Algoritması popülasyonu başlangıç parametre sınırları içerisinde rasgele başlatılır. Her bir G jenerasyonunda, popülasyon pertürbasyona uğrar. Birbirinden farklı üç birey ya da  $x$  ile gösterilen çözüm vektörleri popülasyondan rasgele seçilir. K katsayısı  $x_{r3,G}$  ve mevcut birey  $x_{i,G}$  arasında meydana gelen birleşmenin seviyesini temsil eder. F katsayısı ise  $x_{r1,G} - x_{r2,G}$  vektör farkından ileri gelen adım büyüklüğünün ölçeklenmesini gösterir.



Şekil-2 DG algoritmasında mevcut çözümden rastgele yeni bir çözümün üretilmesi

Şekil-2, yeni bir aday çözümün gelişimi ile ilgili vektörler arasındaki ilişkiyi göstermektedir. Tipik olarak tek amaçlı durumlarda, eğer yeni birey  $u_{i,G+1}$  halen seçili olan birey  $x_{i,G}$ 'den daha iyiyse bu durumda hali hazırdaki birey yeni olanla yer değiştirir. Algoritma  $i$  üzerinden 1'den  $n$ 'e kadar tekrarlanır. Burada  $n$ , popülasyonun büyüklüğüdür.

### 4. TASARIMDA KULLANILAN HATA FONKSİYONLARI

Adaptif Lineer Toplayıcı tasarımında, yaygın olarak kullanılan hata fonksiyonları aşağıda tanımlanmıştır.

**Ortalama Karesel Hata (MSE):** MSE hata fonksiyonu giriş işaretinin, arzu edilen işaretten çıkarılarak karelerinin alınması ve ortalamasının kullanılması fikrine dayalıdır.

$$MSE=1/N \sum_{n=1}^N [d(n) - y(n)]^2 \quad (2)$$

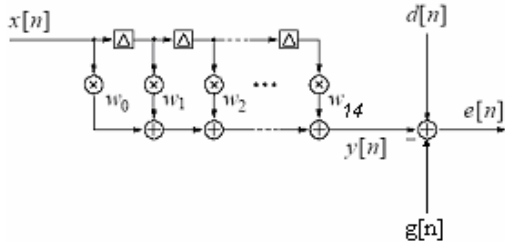
**En Küçük Karesel Hata (LMS):** En Küçük Karesel Hata fonksiyonu, arzu edilen işaretten tasarlanan süzgecin çıkışının çıkarılarak elde edilmiş olan hataların kareleri alınarak toplanması ve toplamın karekökünün minimize edilmesi amacını taşımaktadır.

$$LMS=1/N \left\{ \sum_{n=1}^N [d(n) - y(n)]^2 \right\}^{1/2} \quad (3)$$

**Ortalama Mutlak Hata (MAE):** Ortalama Mutlak Hata fonksiyonu arzu edilen işaret ile süzgeç çıkışı arasındaki farkın, mutlak değerler toplamının en küçük yapılmasını amaçlamaktadır.

$$MAE=1/N \sum_{n=1}^N |d(n)-y(n)| \quad (4)$$

## 5. SİMÜLASYON SONUÇLARI



Şekil-3 15 ağırlıklı adaptif lineer toplayıcı

DG algoritmasında, adaptif süzgecin uygun değerini iyileştirmek için MAE, LMS ve MSE hata fonksiyonları kullanılmıştır. Şekil 3'te, 15 ağırlıklı tek girişli adaptif lineer toplayıcı gösterilmektedir. Giriş ve arzu edilen işaret birim zamanda 5 örnek ile örneklenmiştir. Bu çalışmada kullanılan işaretler, Eşitlik 5'te verilmektedir.

$$d(n)=2\cos(2\pi n/5) \quad n=1,2,\dots,50$$

$$x(n)=\sin(2\pi n/5) \quad (5)$$

$$d_1(n)=d(n)+g(n)$$

Eşitlik 5'te verilen  $d(n)$  işaretine, aynı zamanda  $g(n)$  işareti eklenmiştir.  $g(n)$  normal dağılımlı rasgele bir

gürültü işaretidir ve  $d_1(n)$ , SNR değeri 10dB. olan bir işaretir. Bu çalışmada kullanılan MAE, LMS ve MSE sifıra sürülecek hata fonksiyonları, y izlenecek gerçek işaret ve  $d_1$  arzu edilen işaretir. Tablo-1'de, bu çalışmada kullanılan DG'nin kontrol parametreleri verilmiştir.

Tablo-1. DG'nin kontrol parametreleri

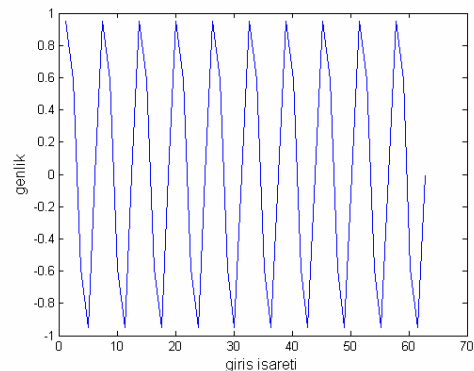
Popülasyon büyüklüğü	100
Çaprazlama oranı	0.8
Birleşme faktörü	0.2
Ölçekleme faktörü	0.2
Jenerasyon sayısı	20000

Adaptif Lineer Toplayıcılar için MAE, LMS ve MSE hata fonksiyonları kullanıldığında elde edilen ağırlıklar Tablo-2'de verilmiştir.

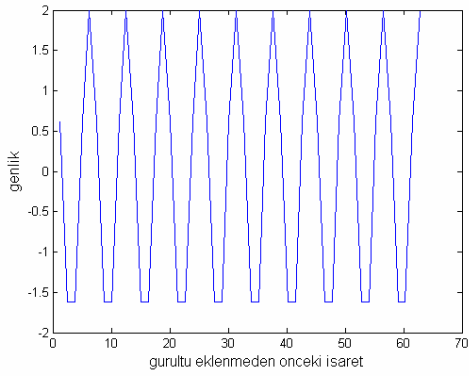
Tablo-2. MAE, LMS ve MSE hata fonksiyonları kullanıldığında Adaptif Lineer Toplayıcı için elde edilen ağırlıklar

Ağırlıklar	Elde Edilen Ağırlıklar		
	MAE	LMS	MSE
$w_0$	0.3893	0.2265	0.2115
$w_1$	-0.0950	-0.9679	-0.0214
$w_2$	0.4523	-0.2007	-1.2133
$w_3$	0.8859	1.0126	1.8632
$w_4$	0.4383	0.8830	1.1143
$w_5$	0.3041	-0.2301	0.7383
$w_6$	-0.0046	1.3069	0.8456
$w_7$	0.1377	-0.2368	1.6929
$w_8$	0.9226	1.0239	0.9356
$w_9$	0.6957	0.7194	0.8868
$w_{10}$	0.7897	0.7222	1.0466
$w_{11}$	0.6015	1.3647	0.7508
$w_{12}$	0.2947	-0.1921	0.9741
$w_{13}$	-0.0339	0.5619	-0.0638
$w_{14}$	0.8265	0.0610	1.0089

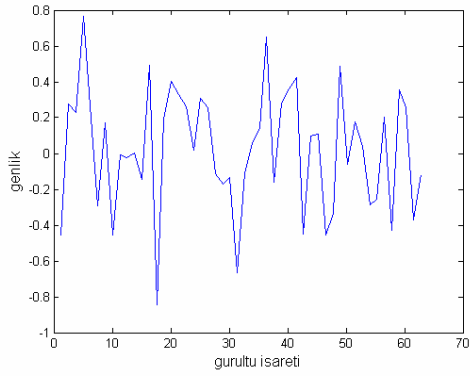
Süzgeç tasarımı için, DG algoritması kullanılarak çıkışla arzu edilen işaret arasındaki hata minimize edilene kadar süzgeç parametreleri ayarlanmıştır.



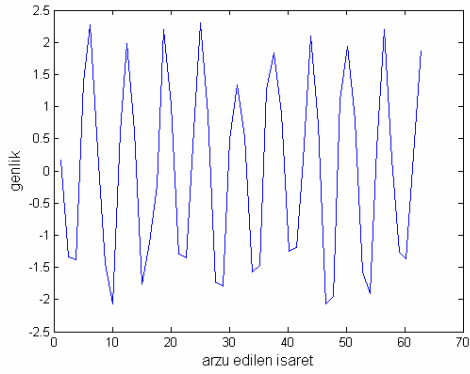
Şekil-4 Giriş işareti,  $x(n)$



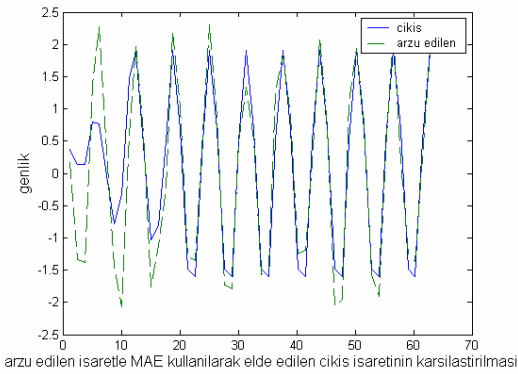
Şekil-5 Gürültü eklenmeden önceki işaret,  $d(n)$



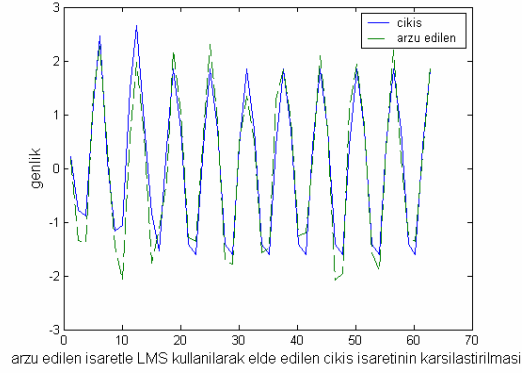
Şekil-6 Gürültü işareti,  $g(n)$



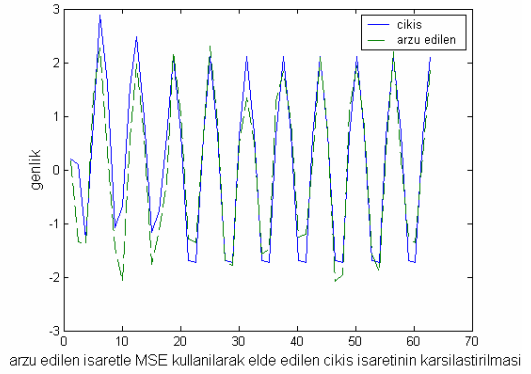
Şekil-7 Arzu edilen işaret,  $d_1(n)$



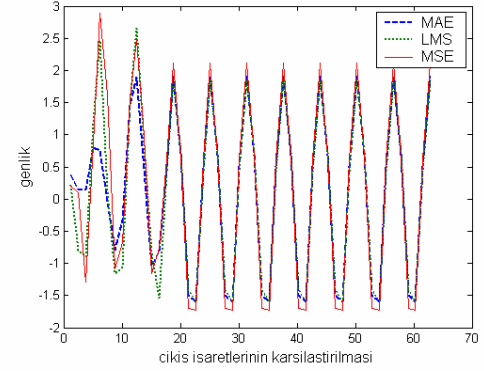
(a)



(b)

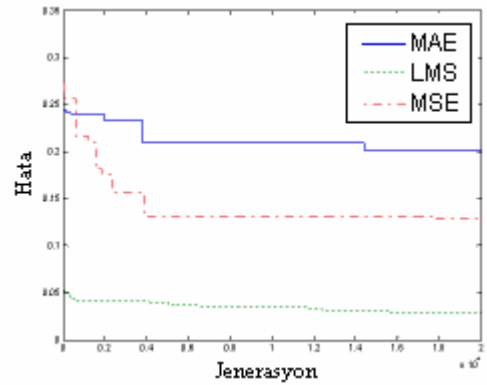


(c)



(d)

Şekil-8 (a) MAE (b) LMS (c) MSE ile elde edilen çıkış işaretleri (d) bu işaretlerin karşılaştırılması



Şekil-9 MAE, MSE ve LMS hatalarının gelişimi

Şekil-3'teki DG algoritmasına ait olan sonuçlar Tablo 1'deki parametre değerleri kullanılarak elde edilmiştir. Şekil-4, 5, 6, 7'de gösterildiği gibi giriş işareti bir sinüs işareti ve arzu edilen işaret de normal dağılımlı rasgele bir gürültü işareti ile bir kosinüs işaretinin birleşimidir. Şekil-8, MSE, MAE ve LMS hata fonksiyonları ile birlikte adaptif lineer toplayıcı çıkışında elde edilen işaretleri göstermektedir. Bu üç hata fonksiyonu ile alınan çıkışların birbirine çok yakın olduğu şekillerden görülmektedir. Buradan hata fonksiyonlarının benzer performanslar gösterdiği anlaşılmaktadır. Şekil-9'da, jenerasyona göre hata fonksiyonlarının performanslarını görülmektedir. Şekil-8(a), (b), (c), (d) ve Şekil-9'da görüldüğü gibi en uygun hata fonksiyonu LMS'dir.

## 6. SONUÇ

Bu çalışmada DG algoritması, farklı hata fonksiyonları kullanılarak adaptif lineer toplayıcı tasarımına uygulanmıştır. Simülasyon sonuçlarından da görüldüğü gibi, MAE, MSE ve LMS hata fonksiyonları gürültü olsa bile, adaptif lineer toplayıcı tasarımında etkin bir şekilde kullanılabilir. DG algoritması ile MAE, MSE ve LMS hata fonksiyonları ayrı ayrı kullanıldığında tasarlanan adaptif lineer toplayıcı kısa bir süre sonra arzu edilen işareti hemen yakalamıştır. Adaptif lineer toplayıcı tasarımında hataların gelişimi değerlendirildiğinde en iyi performansı LMS hata fonksiyonu göstermektedir.

## KAYNAKLAR

- [1] Nurhan Karaboğa, Canan Aslıhan Koyuncu: Optimal Determination of Adaptive Linear Combiner Weights Using Differential Evolution Algorithm. INISTA, Yıldız Technical University, Istanbul (2005),71-74
- [2] Lohmann R, Gorne T, Schneider M and Warstat M: Digital Systems and Structure Evolution, Prepr. 94<sup>th</sup> Conv. Audio Eng. Soc., D6-2, Berlin (1993)
- [3] Adem Kalinli, Nurhan Karaboga: A New Method for Adaptive IIR Filter Design Based on Tabu Search Algorithm, AEÜ, 2004.
- [4] Storn, R. - Price, K.: Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Vol. 11. Kluwer Academic Publishers (1997) 341 – 359
- [5] Derviş Karaboğa, Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayınevi, İstanbul, 2004.
- [6] Allan V. Oppenheim: Adaptive Signal Processing. Prentice Hall Signal Processing Series, Prentice Hall (1989) 22-23
- [7] Price K. V.: Differential Evolution: a Fast and Simple Numerical Optimizer. In: Smith, M., Lee, M., Keller, J., Yen., J. (eds.): Biennial Conference of the North American Fuzzy Information Processing

Society, NAFIPS. IEEE Press, New York (1996) 524-527

- [8] Ilonen, J., Kamarainen, J. -K., Lampinen, J.: Differential Evolution Training Algorithm for Feed-Forward Neural Networks. In: Neural Processing Letters Vol. 7, No. 1 (2003) 93-105
- [9] Storn, R.: Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay. In: Proceedings IEEE International Conference on Evolutionary Computation ICEC'96. IEEE Press, New York (1996) 268 – 273
- [10] Rogalsky, T., Derksen, R.W. and Kocabiyik, S.: Differential Evolution in Aerodynamic Optimization. In: Proceedings of the 46<sup>th</sup> Annual Conference of the Canadian Aeronautics and Space Institute. (1999) 29-36
- [11] Dervis Karaboga, Selcuk Okdem,: A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. Turk J Elec Engin, Vol.12, No.1,(2004)
- [12] Nurhan Karaboga,: Digital IIR Filter Design Using Differential Evolution Algorithm. EURASIP Applied Signal Processing (2005) No.8, 1269-1276.
- [13] Price, K V.: An Introduction to Differential Evolution. In: Come, D., Dorigo, M., and Glover, F. (eds): New Ideas in Optimization McGraw-Hill, London UK (1999) 79-108