

Ortalama farksal gelişim algoritması ile bilineer sistem kimliklendirme

Burhanettin DURMUŞ*

*Kütahya Dumlupınar Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü,
Evliya Çelebi Yerleşkesi, Kütahya.*

*Geliş Tarihi (Received Date): 10.08.2018
Kabul Tarihi (Accepted Date): 21.11.2018*

Özet

Bu çalışmada, bilineer sistem kimliklendirme problemi için ortalama farksal gelişim (average differential evolution-ADE) algoritması önerilmiştir. Doğrusal olmayan sisteme ait parametrelerin ADE tabanlı bilineer model üzerinden kestirimi gerçekleştirilmiştir. Bilinmeyen sistem çıkışı ile bilineer model çıkışı arasındaki Ortalama Karese Hata (Mean Square Error, MSE) performans ölçütü olarak kullanılmıştır. Önerilen algoritmanın performansı, hem farklı sezgisel algoritmaların kullanıldığı benzetim çalışmaları ile hem de literatürde rapor edilmiş diğer metotlar ile karşılaştırılmıştır. Karşılaştırmalı sonuçlarda, ADE tabanlı modelleme ile hata değerlerinin azaldığı ve hesaplanan parametre değerlerinin doğruluk oranının arttığı görülmüştür. Hızlı bir yakınsama ile global çözüme ulaşma kabiliyetine sahip olan ADE algoritması, parametre kestirimi uygulamaları için etkin bir araç olarak kullanılabilir.

Anahtar kelimeler: *Bilineer sistem, sistem kimliklendirme, sezgisel algoritmalar.*

Bilinear system identification with average differential evolution algorithm

Abstract

In this paper, average differential evolution (ADE) algorithm is proposed for bilinear system identification problem. The parameters of the nonlinear system were estimated using ADE based bilinear model. The mean square error (MSE) between the unknown system output and the bilinear model output is used as the performance criterion. The performance of the proposed algorithm is compared with both the simulation studies

* Burhanettin DURMUŞ, burhanettin.durmus@dpu.edu.tr, <http://orcid.org/0000-0002-8225-3313>

using different heuristic algorithms and other methods reported in the literature. The comparative results show that the ADE based modeling reduces the error values and increases the accuracy of the calculated parameter values. The ADE algorithm, which has global convergence capability with fast convergence, can be used as an effective tool for parameter estimation applications.

Keywords: *Bilinear system, system identification, meta-heuristic algorithms.*

1. Giriş

Sistem kimliklendirme problemi, deneysel bir veri setine göre sistemin giriş-çıkışını gözlemleyerek dinamik bir sistem modelini kestirmek olarak tanımlanmaktadır [1]. Bu tanıma göre, sistem davranışını belirleyen karakteristik modelin belirlenmesi gerekmektedir. Dolayısıyla sistem davranışını temsil edebilecek karakterizasyonun başarımı, sisteme ait parametrelerin doğru bir şekilde kestirimine bağlıdır.

Sistem kimliklendirme uygulamaları uzun süredir ilgi çeken bir konu olmuş ve akademik çalışmaların devam ettiği bir alan olarak görülmektedir [1-4]. Bu çalışmalar incelendiğinde genel olarak bilinmeyen sistem parametreleri bir kestirim algoritması üzerinden hesaplanmaktadır. Özellikle, filtrelerin bir sistem kimliklendirme aracı olarak modellendiği çalışmalar ön plana çıkmaktadır [2, 4-6]. Genel olarak doğrusal modelleme yapısında, geleneksel metotlar yerine sezgisel algoritmaların parametre kestiriminde kullanılması, doğrusal sistem kimliklendirme prosesinin etkinliğini arttırmaktadır. Bu modelleme yapılarında kimliklendirme problemi, aynı giriş işaretinin uygulanması sonucu bilinmeyen sistem ile model filtre çıkışı arasındaki hatanın minimize edilmesi şeklinde bir optimizasyon problemi olarak ele alınmaktadır. Söz konusu hata değerinin minimizasyonu ile sisteme ait parametre değerlerine ulaşılacak istenmektedir [6-8].

Öte yandan kontrol, ekonomi, biyoloji gibi birçok alandaki gerçek hayat problemlerinde, sistem modellerinin doğrusal olmayan davranış şekillerine sahip olduğu görülmektedir [9-13]. Doğrusal olmayan sistemlerin doğrusal modeller ile modellenmesi etkin bir kimliklendirme sağlayamamaktadır. Bu bağlamda, pratik sistemleri modelleme kabiliyetleri sebebiyle bilineer sistemlere olan ilgi artmaktadır [12, 14-16]. Bundan dolayı bu çalışmada sezgisel hesaplama tabanlı bilineer modelleme konusuna odaklanılmıştır.

ADE algoritması, yeni önerilen sezgisel algoritmalarından biri olup hızlı yakınsama oranına sahip bir optimizasyon aracı olarak sunulmaktadır [17]. Popülasyondaki bireylerin ortalama değerleri üzerinden yeni birey üretiminin gerçekleştirildiği bu sezgisel algortmada, kümülatif bir çözüm gelişimi sağlanmaktadır. Bu çalışmada, bilineer sistem kimliklendirme problemi için ADE algoritması önerilmiştir. Bilinmeyen sistem parametreleri, bilineer model kullanılarak önerilen algoritma üzerinden kestirilmiştir. Önerilen algoritmanın etkinliğini göstermek için Armoni Arama Algoritması (Harmony Search Algorithm, HS) [18], Yerçekimsel Arama Algoritması (Gravitational Search Algorithm, GSA) [19] ve Yüklü Sistem Arama Algoritması (Charged System Search, CSS) [20] algoritmaları ile de aynı modellemeler yapılmıştır. Ayrıca ele alınan örnek modellere ait literatürde yer alan diğer metotlar ile de

karşılaştırmalar gerçekleştirilmiştir. Karşılaştırmalarda performans ölçütü olarak MSE değerleri kullanılmıştır.

2. Bilineer sistem kimliklendirme

Özyineli doğrusal olmayan sistemlerin bir sınıfı olan bilineer sistemler mühendislik, ekonomi ve biyoloji gibi birçok alanda kullanılan modelleme türüdür. Basit ve doğrusal bir sistem formuna benzer yapıları sayesinde bu modelleme türü, parametre kestirimi uygulamalarında büyük ilgi görmektedirler [9, 12, 13, 21].

Bilineer sistemde genel olarak giriş-çıkış arasındaki ilişki aşağıdaki gibi tanımlanır:

$$y(n) = \sum_{i=1}^{K_a} a_i y(n-i) + \sum_{j=0}^{K_b} b_j x(n-i) + \sum_{i=1}^{K_{cy}} \sum_{j=0}^{K_{cx}} c_{i,j} y(n-i)x(n-j) \quad (1)$$

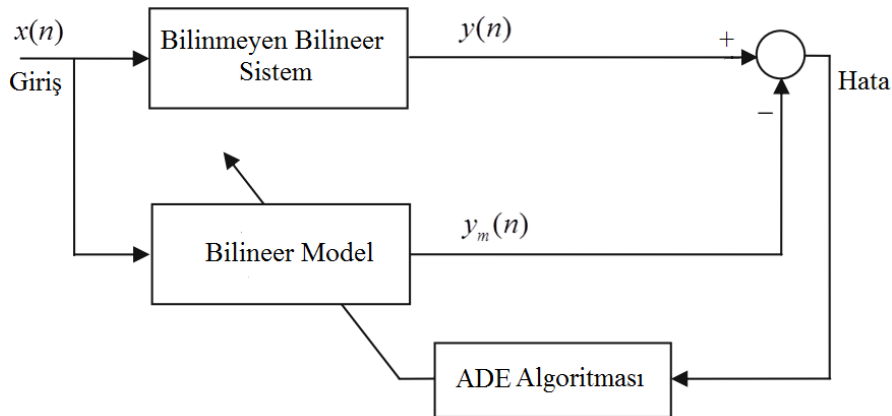
Burada $y(n)$ sistem çıkışını, $x(n)$ sistem girişini, a , b , c sistem katsayılarını temsil eder. K_a , K_b , K_{cy} ve K_{cx} ise sistemin polinom derecesidir.

$$y_m(n) = \sum_{i=1}^{K_a} a'_i y(n-i) + \sum_{j=0}^{K_b} b'_j x(n-i) + \sum_{i=1}^{K_{cy}} \sum_{j=0}^{K_{cx}} c'_{i,j} y_m(n-i)x(n-j) \quad (2)$$

Parametre kestiriminin temel amacı, sistem parametrelerini (2) denkleminde tanımlanan bilineer model üzerinden tahmin etmektir. Diğer bir deyişle model parametre değerlerini gerçek parametre değerlerine yaklaştırmaktır. Dolayısıyla problem, gerçek sisteme ait çıkış ile bilineer model sisteme ait çıkış arasındaki hatanın minimize edilmesi olarak bir optimizasyon problemine dönüşmektedir. Bu durumda, optimizasyon probleminin amaç fonksiyonu (3) denkleminde tanımlanan hata kareleri ortalaması (MSE) olarak tanımlanmaktadır [13-16].

$$MSE = \frac{1}{N} \sum_{i=1}^N [y(i) - y_m(i)]^2 \quad (3)$$

Burada N , sisteme ve modele uygulanan giriş işaret dizisindeki örnek sayısıdır.



Şekil 1. Bilineer sistem kimliklendirme blok diyagramı.

Şekil 1 'de şematik gösterimi verilen bilineer sistem kimliklendirme probleminde, bilinmeyen sistem parametreleri bilineer model üzerinden izlenmek suretiyle kestirilmektedir. Hem sisteme hem de modele aynı giriş uygulanarak elde edilen çıkışlar arasındaki fark yani hata değeri minimize edilmektedir. Sezgisel algoritmanın buradaki görevi; (3) denklemde tanımlanan hata değerini minimize edecek uygun katsayı değerlerini üretmektir. Hesaplama sonunda en düşük hatayı üreten değerler çözüm olarak kabul edilir.

3. ADE algoritması

ADE algoritması, diğer evrimsel hesaplama algoritmaları gibi çözüm geliştirme sürecinde sezgisel ve rassal değişimleri ve seçimleri kullanan, popülasyon tabanlı yeni önerilmiş bir sezgisel algoritmadır [17]. Hızlı bir yakınsama davranışı gösteren bu sezgisel algoritma, problemlerinin çözümünde kayda değer sonuçlar üretmektedir. Mevcut popülasyondaki çözüm değerlerinin ortalaması kullanıldığından, aday çözüm üretimi aşamasında kümülatif bir gelişim sağlanmaktadır.

ADE algoritması ilklendirme, değerlendirme, aday çözüm geliştirme, sınır aşımı ve kısıt kontrolü, seçim ve sonlandırma olmak üzere altı işlem basamağına sahiptir. Bu işlem basamakları aşağıda verilen alt başlıklarda anlatılmıştır.

3.1. İlklendirme

İlklendirme adımında, belirlenen parametre sınırlarına bağlı olarak rastgele üretilen ve olası çözüm değerlerini içeren başlangıç havuzu oluşturulur. Her bir çözüm vektörü bir parametrik dizi şeklinde tanımlanmakta olup aşağıdaki şekilde üretilmektedir:

$$x_{i,G}^j = x_{i,L} + rand \cdot (x_{i,U} - x_{i,L}) \quad i = 1, 2, \dots, PS \quad ve \quad j = 1, 2, \dots, D \quad (4)$$

Burada, x çözüm vektör setini, PS popülasyondaki birey sayısını, D her bir bireydeki değişken sayısını, $x_{i,U}$ ve $x_{i,L}$ değişkenlerin alt ve üst sınır değerlerini, $rand$ [0, 1] aralığında rastgele sayıyı, G ise kuşak sayısını temsil eder.

3.2. Değerlendirme

Bu aşamada, üretilen çözüm vektörlerinin uygunluk değerleri belirlenir. Uygunluk değerleri aslında çözüm vektörlerinin kalitesini temsil etmektedir. Bundan dolayı, her bir çözüm vektörü için hesaplanan amaç fonksiyon değeri, o vektörün uygunluk değeri olarak alınır. Optimizasyon problemi maksimizasyon problemi ise fonksiyon değeri en yüksek olan, minimizasyon problemi ise fonksiyon değeri en düşük olan vektör, uygunluk değeri en iyi olan vektör olarak kabul edilir.

3.3. Aday çözüm geliştirme

Bu aşamada, sonraki kuşaklar için aday çözüm vektörleri oluşturulur. ADE algoritması, mevcut kuşaktaki en iyi bireyi ve bireylerin ortalamasını kullanan bir mutasyon operatörü ile aday vektör üretimini gerçekleştirir. İlk olarak, mevcut kuşak için ortalama vektörü \bar{A}_G hesaplanır. Bu vektör o kuşakta bulunan tüm vektörlerin ortalaması şeklinde aşağıdaki denklem ile hesaplanır:

$$\vec{A}_G = \frac{1}{PS} \sum_{i=1}^{PS} \vec{x}_{i,G} = (1/PS) * (\vec{x}_{1,G} + \vec{x}_{2,G} + \dots + x_{PS,G}) \quad (5)$$

Burada, \vec{A}_G vektörü G . kuşaktaki ortalama vektörü, PS popülasyondaki çözüm vektör sayısı, \vec{x}_i ise çözüm vektörleridir.

Mevcut kuşağa ait ortalama vektör hesaplandıktan sonra popülasyonda bulunan her bir çözüm vektörü için aday çözümü üretimine geçilir. Aday vektör üretimi ilk olarak mutasyon vektörü, akabinde ise çaprazlama işlemi ile gerçekleştirilir. Mutasyon vektörü, hedef vektörün ortalama vektör ile farkının alınması ve en iyi vektöre yaklaşması şeklinde aşağıdaki denklem kullanılarak üretilir.

$$\vec{u}_{i,G+1} = \vec{x}_{best,G} + \gamma \cdot rand_i[-1,1].[\vec{A}_G - \vec{x}_{i,G}] \quad (6)$$

Burada, $\vec{u}_{i,G+1}$ vektörü mutasyon vektörünü, $\vec{x}_{best,G}$ vektörü G . kuşaktaki en iyi vektörü, \vec{A}_G vektörü G . kuşaktaki ortalama vektörü, $\vec{x}_{i,G}$ vektörü hedef vektörü, γ ölçekleme faktörünü ve $rand_i[-1,1]$ ise $[-1, 1]$ aralığındaki rastgele sayıları temsil eder.

Yukarıda tanımlanan mutasyon işlemi ile hedef vektörün mevcut kuşaktaki ortalama vektör ile farkı alınmaktadır. Yani, hedef vektörü oluşturan parametrik değerler, mevcut kuşaktaki vektörlere ait parametrik değerlerin ortalamasına yaklaştırılır. Böylece mevcut kuşaktaki çözüm vektörlerine ait genetik kazanım, ortalama bir öğrenme mirası şeklinde yeni vektör üretimine aktarılmaktadır. Daha kümülatif bir vektörel değişim oluşmaktadır. Bu fark vektörüne kuşağın en iyi vektörünü ilave etmek sureti ile de, en iyi çözüme doğru bir yönelim gerçekleşmektedir. Böylece global çözüme yakın değerlere hızlı bir yakınsama sağlanmaktadır.

Aday çözümü üreten son adım ise çaprazlama işlemidir. Bu işlem basamağında mutasyon vektörü $\vec{u}_{i,G+1}$ ile hedef vektör $\vec{x}_{i,G}$ arasında C_r olasılıklı parametre bazında bir çaprazlama işlemi gerçekleştirilir. Çözümlere ait her bir parametre için gerçekleştirilen bu işlem sonunda sonraki kuşağa bir aday vektör elde edilmektedir. Çaprazlama işlemi aşağıdaki denklem ile gerçekleştirilir:

$$\hat{x}_{i,G+1}^j = \begin{cases} \text{Eğer } rand_j[0,1] \leq C_r \text{ ise} & u_{i,G+1}^j \\ \text{Aksi durumda} & x_{i,G}^j \end{cases} \quad (7)$$

3.4. Sınır aşımı ve kısıt kontrolü

Önceki aşamada üretilen aday çözümün parametrik elemanları, ilklendirme aşamasında tanımlanan sınırların dışına çıkabilmektedirler. Bu aşamada, sınır aşan ilgili parametrik değerler en yakın sınır değerine çekilirler. Ayrıca, kısıtlı optimizasyon problemlerinde tanımlı kısıtlara uyulup uyulmadığının kontrolü denetlenir. Zira üretilen aday vektör kısıtları sağlamıyor olabilir. ADE algoritmasında, kısıt aşımını engellemek için literatürde yaygın bir şekilde kullanılan ceza fonksiyonu tanımlanmaktadır. Ceza fonksiyonu ilaveli amaç fonksiyonu aşağıdaki şekilde ifade edilir:

$$f_p(x) = f(x) + \mu \sum_{i=1}^M \tau_k [g_i(x)]^2 \begin{cases} \text{Eğer } g_i(x) \text{ kısıtı aşılmış ise,} & \tau_k = 1 \\ \text{Aksi durumda} & \tau_k = 0 \end{cases} \quad (8)$$

Burada, $f(x)$ probleme ait amaç fonksiyonunu, $g_i(x)$ i . eşitsizlik kısıtını, μ ise ceza katsayısını temsil eder. Kısıt aşımı gerçekleştiğinde çözüm vektörüne ait ceza puanı artmakta ve bu da uygunluk değerini düşürmektedir. Uygunluk değeri düşen vektöründe sonraki kuşaklara aktarımı zorlaşmaktadır. Böylece kısıt aşımına sebebiyet veren çözümlerin üretilmesi engellenmektedir.

3.5. Seçim

Üretim aşamasında elde edilen aday çözümün bir sonraki kuşağa aktarılmasına bu işlem basamağında karar verilir. Karar verme işlemi $\bar{x}_{i,G}$ hedef ve $\hat{x}_{i,G+1}$ aday vektörlerin uygunluk değerleri üzerinden gerçekleştirilir. Buradaki amaç; aday çözümlerin uygunluklarını maksimumlaştırmaktır. Aşağıdaki denklemde ifade edildiği gibi uygunluğu daha iyi olan vektör sonraki kuşağa aktarılır.

$$\bar{x}_{i,G+1} = \begin{cases} \hat{x}_{i,G+1} & \text{Eğer } f(\hat{x}_{i,G+1}) > f(\bar{x}_{i,G}) \\ \bar{x}_{i,G} & \text{Aksi durumda} \end{cases} \quad (9)$$

Burada, $f(\hat{x}_{i,G+1})$ ve $f(\bar{x}_{i,G})$ sırası ile aday ve hedef vektörlerin uygunluk değerlerini temsil eder.

3.6. Sonlandırma

Yukarıda tanımlanan beş aşama sonlandırma kriteri sağlanıncaya kadar sürdürülür. Kuşak sayısı genelde sonlandırma kriteri olarak alınır. Hesaplama öncesinde tanımlanan kuşak sayısına ulaşıldığında hesaplama durdurulur ve o ana kadar elde edilen en iyi vektör çözüm olarak kabul edilir.

4. Benzetim sonuçları

Bu bölümde, ADE tabanlı parametre kimliklendirme modeli tasarlanmış, Şekil 1 'de gösterilen kimliklendirme şeması üzerinden bilinmeyen sistem parametreleri hesaplanmıştır. Benzetim çalışmaları literatürde yer alan üç örnek uygulama üzerinden, ADE algoritması ile HS [18], GSA [19] ve CSS [20] algoritmalarının temel versiyonları kullanılarak gerçekleştirilmiştir. Kullanılan algoritmalara ait parametre değerleri ilgili çalışmalarda belirtilen değerlere göre sırası ile şu şekilde seçilmiştir: HS için $HMS = 5$, $HMCR = 0.8$, $PAR_{\min} = 0.1$, $PAR_{\max} = 0.9$; GSA için $G_0 = 100$, $\alpha = 10$; CSS için $k_a = 0.5$, $k_v = 0.5$, $k_t = 0.8$. ADE algoritması için $C_r = 0.85$, $\gamma = 2$.

Her bir uygulama için algoritmalar ile 30 bağımsız çalıştırma gerçekleştirilmiş ve sonuçlar tablosal olarak sunulmuştur. Adil bir karşılaştırma yapmak için; popülasyon sayısı 20 olarak belirlenmiş, sistem girişine Gaussian dağılımlı beyaz gürültülü sinyal uygulanmıştır [15]. İlaveten, benzetim çalışmalarından elde edilen sonuçlar literatürde

rapor edilmiş diğer metotlar ile de karşılaştırılmıştır. Performans ölçütü olarak (3) denkleminde tanımlanan MSE değerleri kullanılmıştır.

4.1. Uygulama-1

Çalışmanın ilk örneğinde, (10) denkleminde tanımlanan bilineer sistem ele alınmıştır [16, 22]. Bilineer sistemin ve modelin giriş-çıkış arasındaki bağıntısı aşağıdaki şekilde tanımlanmaktadır:

$$y(n) = 0.325y(n-1) - 0.4x(n-1) + 0.56x(n-3) - x(n-4) - 0.65y(n-2)x(n-2) + 0.065y(n-3)x(n-5) + 0.065y(n-5)x(n-2) \quad (10)$$

$$y_m(n) = a'_1y_m(n-1) + b'_1x(n-1) + b'_3x(n-3) + b'_4x(n-4) + c'_{2,2}y_m(n-2)x(n-2) + c'_{3,5}y_m(n-3)x(n-5) + c'_{5,2}y_m(n-5)x(n-2) \quad (11)$$

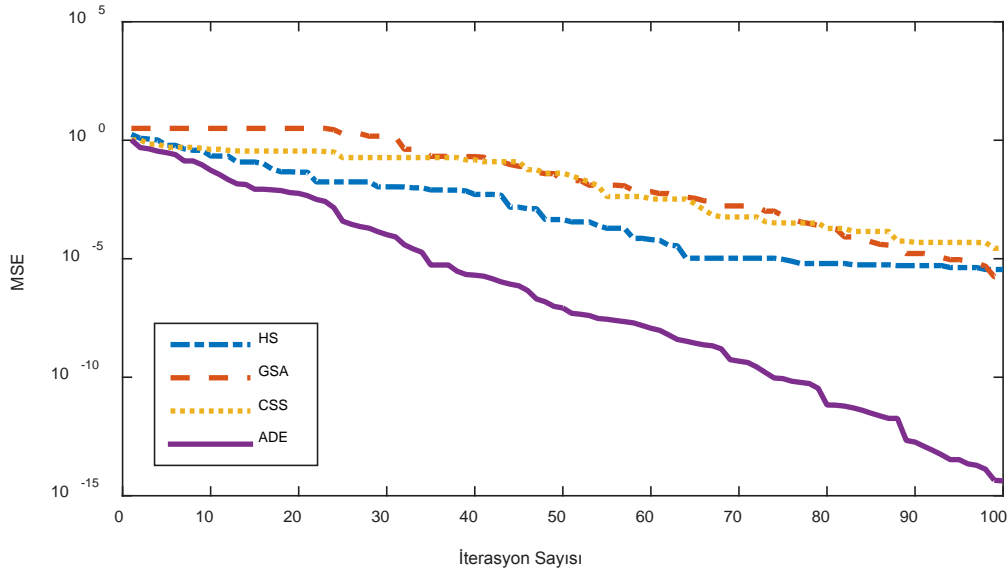
Burada, $y(n)$ sistem çıkışını, $y_m(n)$ model sistem çıkışını, $x(n)$ sistem girişini, $[a, b, c]$ katsayıları ise sistem parametrelerini temsil eder.

Tablo 1’de, önerilen ADE algoritması ve diğer sezgiseller ile 30 bağımsız çalıştırma sonucunda elde edilen en iyi, ortalama, en kötü hata değerleri ile standart sapma değerlerini içeren benzetim sonuçları verilmiştir. Bu tablodaki sonuçlara göre; ADE ‘nin diğer metotlara göre hem en iyi hem de ortalama hata değerlerinde daha düşük MSE değerlerini ürettiği görülmektedir. Ayrıca Şekil 2’de gösterilen hesaplama sürecine ait yakınsama eğrileri incelendiğinde, ADE algoritmasının hızlı bir yakınsama gerçekleştirdiği, yaklaşık 30 iterasyon sonunda MSE değerinin 10^{-5} seviyelerine düştüğü görülmektedir.

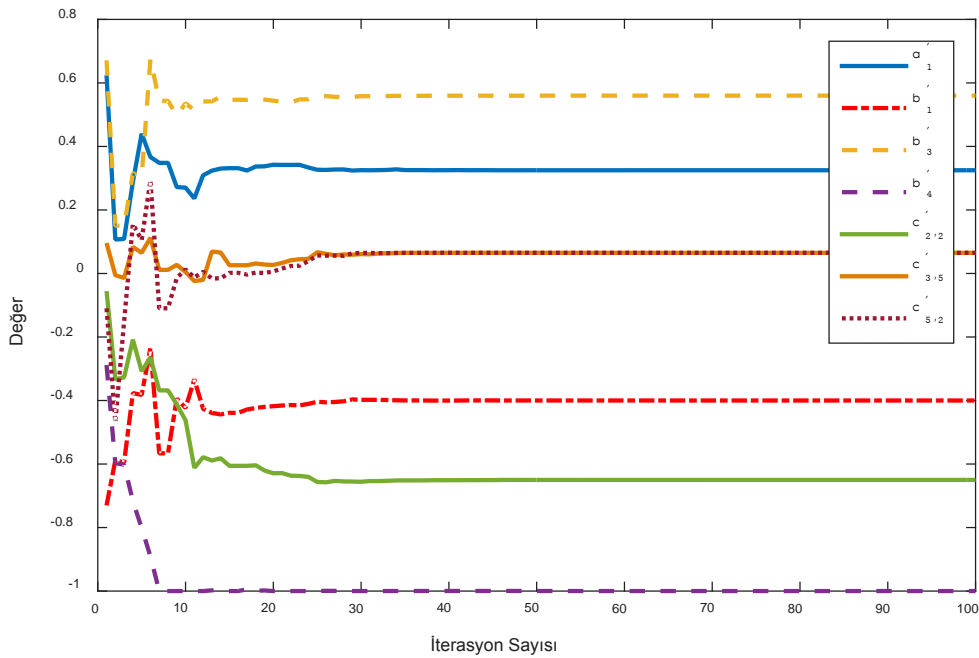
Tablo 1. Uygulama-1’e ait hesaplanan hata değerleri.

Algoritmalar	En iyi	Ortalama	En kötü	Standart sapma
HS	3.51E-06	2.90E-04	4.50E-03	8.26E-04
GSA	1.72E-06	7.30E-03	4.25E-02	1.28E-02
CSS	2.75E-05	3.37E-02	3.41E-01	6.96E-02
ADE	4.22E-15	3.96E-08	1.02E-06	1.87E-07

ADE ile elde edilen sonuçların literatür sonuçları ile birlikte sunulduğu Tablo 2’deki veriler karşılaştırıldığında da, benzer sonuç ifadesi desteklenebilir. ADE algoritması ile daha düşük seviyede hata oranı ile hedef sisteme ait parametre değerlerinin hesaplandığı görülmektedir. Ayrıca hesaplama süresi boyunca model parametrelerinin gelişimi Şekil 3’de gösterilmiş olup, hesaplama sonunda parametrelerin gerçek değerlerine ulaştığı izlenmektedir.



Şekil 2. Uygulama-1'e ait yakınsama eğrileri.



Şekil 3. Uygulama-1'e ait parametre gelişimi.

4.2. Uygulama-2

Bu örnek; bilinear sistem kimliklendirme çalışmalarını konu alan çalışmalarda yer alan örneklerden biridir [14-16] ve aşağıdaki gibi tanımlanmaktadır:

$$y(n+1) = 0.898y(n-1) + 0.284x(n-1) - 0.106y(n)x(n) \quad (12)$$

$$y_m(n+1) = a'_1 y_m(n-1) + b'_1 x(n-1) + c'_{0,0} y_m(n)x(n) \quad (13)$$

Burada, $y(n)$ sistem çıkışını, $y_m(n)$ model sistem çıkışını, $x(n)$ sistem girişini, $[a, b, c]$ katsayıları ise sistem parametrelerini temsil eder.

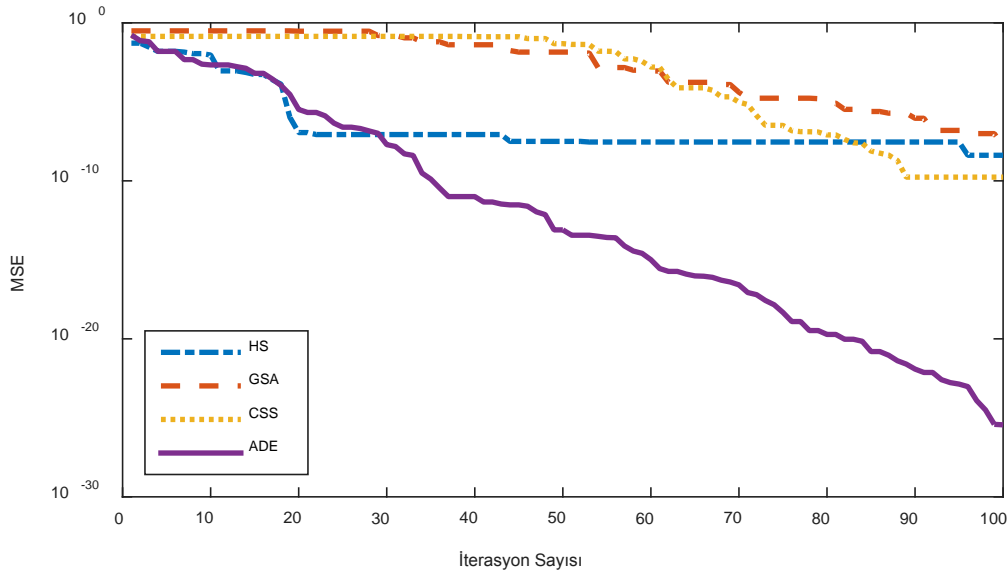
Tablo 2. Uygulama-1'e ait hesaplanan parametre değerleri.

Algoritmalar	a'_1	b'_1	b'_3	b'_4	$c'_{2,2}$	$c'_{3,5}$	$c'_{5,2}$	MSE	MSE(dB)
Gerçek değer	0.325	-0.400	0.560	-1.000	-0.650	0.065	0.065	-	Sonsuz
GA[22]	0.326	-0.394	0.567	-1.009	-0.612	0.003	0.044	5.60E-05	-42.51
DE[22]	0.325	-0.401	0.559	-0.997	-0.656	-0.016	0.060	1.50E-05	-42.23
PSO[16]	0.298	-0.384	0.529	-0.956	-0.548	0.098	0.024	7.43E-05	-41.29
DE[16]	0.326	-0.396	0.560	-0.998	-0.648	0.066	0.067	1.96E-06	-57.07
CSA[16]	0.324	-0.399	0.559	-0.999	-0.649	0.065	0.065	6.44E-09	-81.91
SFLA[16]	0.320	-0.399	0.556	-0.993	-0.630	0.063	0.047	1.01E-08	-79.95
LSFLA[16]	0.325	-0.399	0.559	-1.000	-0.650	0.064	0.065	6.39E-11	-101.94
ADE	0.325	-0.399	0.559	-1.000	-0.650	0.064	0.065	4.22E-15	-143.74

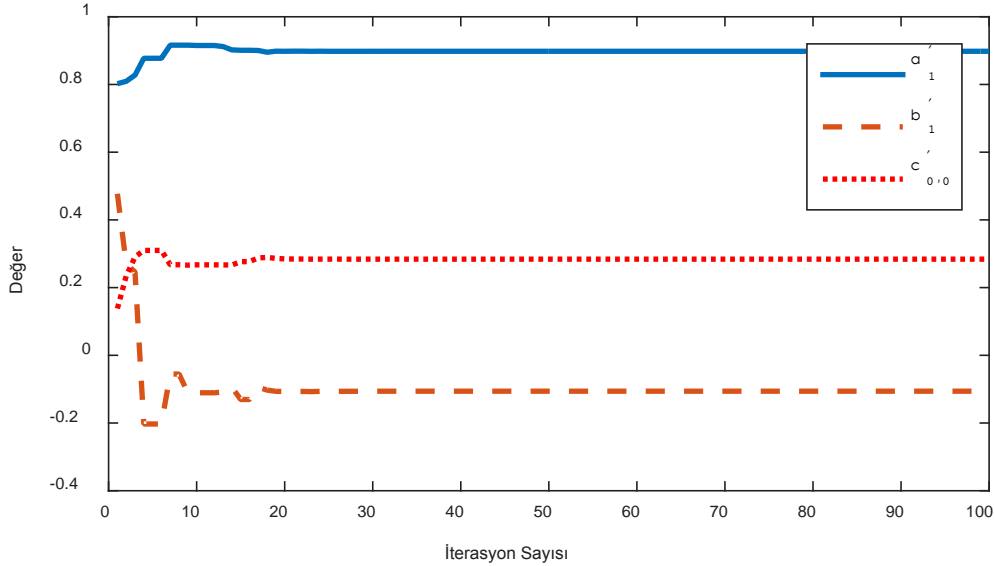
ADE algoritması ve diğer sezgisellere ait 30 deneme sonucunda elde edilen istatistiksel sonuçlar Tablo 3'de sunulmuştur. Bu tablodan görüldüğü gibi ADE algoritması ile hem en iyi hem de ortalama hata değerleri sırasıyla $3.60E-26$ ve $6.02E-18$ ile en düşük seviyede elde edilmiştir. Şekil 4'de sunulan yakınsama eğrilerine bakıldığında da ADE algoritmasının hızlı bir yakınsama yaptığı ve 35 iterasyon sonunda MSE değerinin 10^{-10} seviyesinin altına düştüğü görülmektedir.

Tablo 3. Uygulama-2'ye ait hesaplanan hata değerleri.

Algoritmalar	En iyi	Ortalama	En kötü	Standart sapma
HS	4.16E-09	3.81E-07	2.25E-06	4.91E-07
GSA	1.60E-08	2.05E-05	2.71E-04	6.65E-05
CSS	1.71E-10	3.02E-05	7.42E-04	1.35E-04
ADE	3.60E-26	6.02E-18	1.80E-16	3.28E-18



Şekil 4. Uygulama-2'ye ait yakınsama eğrileri.



Şekil 5. Uygulama-2'ye ait parametre gelişimi.

Tablo 4'de, ADE ve literatürde rapor edilmiş diğer çalışmalara ait sonuçlar verilmiştir. Bu tablodan, ADE algoritmasının diğer metotlara göre sistem parametrelerini daha doğru bir şekilde hesapladığı ve gerçek değerlere ulaştığı sonucu çıkmaktadır. Ayrıca bu sonucu hesaplama sürecine ait parametrelerin değişimini gösteren Şekil 5'de desteklemektedir.

Tablo 4. Uygulama-2'ye ait hesaplanan parametre değerleri.

Algoritmalar	a'_1	b'_1	$c'_{0,0}$	MSE	MSE(dB)
Gerçek değer	0.898	-0.106	0.284	-	Sonsuz
LDW-PSO [15]	0.898	-0.106	0.284	8.70E-11	-100.60
NDW-PSO [15]	0.898	-0.106	0.283	2.30E-13	-126.38
DIW-PSO [15]	0.898	-0.106	0.283	1.10E-14	-139.58
APSO [15]	0.898	-0.106	0.283	5.20E-22	-212.84
GA [14]	0.897	-0.107	0.284	5.00E-06	-53.01
PSO [16]	0.890	-0.125	0.284	6.42E-07	-61.92
DE [16]	0.891	-0.106	0.283	2.31E-06	-56.36
CSA [16]	0.894	-0.112	0.283	1.37E-09	-88.63
SFLA [16]	0.887	-0.114	0.287	7.17E-10	-91.44
LSFLA [16]	0.898	-0.106	0.279	4.70E-11	-103.27
ADE	0.898	-0.106	0.284	3.60E-26	-254.43

4.3. Uygulama-3

Bu örnekte, önceki örneklerden farklı olarak bileer sistemin kimliklendirmesi için Sonlu Tepki Cevabı (Finite Impulse Response, FIR) modeli kullanılmıştır [23]. Bileer sistemin ve FIR modelin giriş-çıkış arasındaki bağıntısı aşağıdaki şekilde tanımlanmaktadır:

$$y(n) = 0.25y(n-1) - 0.5y(n-1)x(n) + 0.05y(n-1)x(n-1) - 0.5x(n) + 0.5x(n-1) \quad (14)$$

$$y_m(n) = a'_0 x(n) + a'_1 x(n-1) \quad (15)$$

Burada, $y(n)$ sistem çıkışı, $y_m(n)$ model sistem çıkışı, $x(n)$ sistem girişini, $[a'_0, a'_1]$ katsayıları ise FIR model parametrelerini temsil eder.

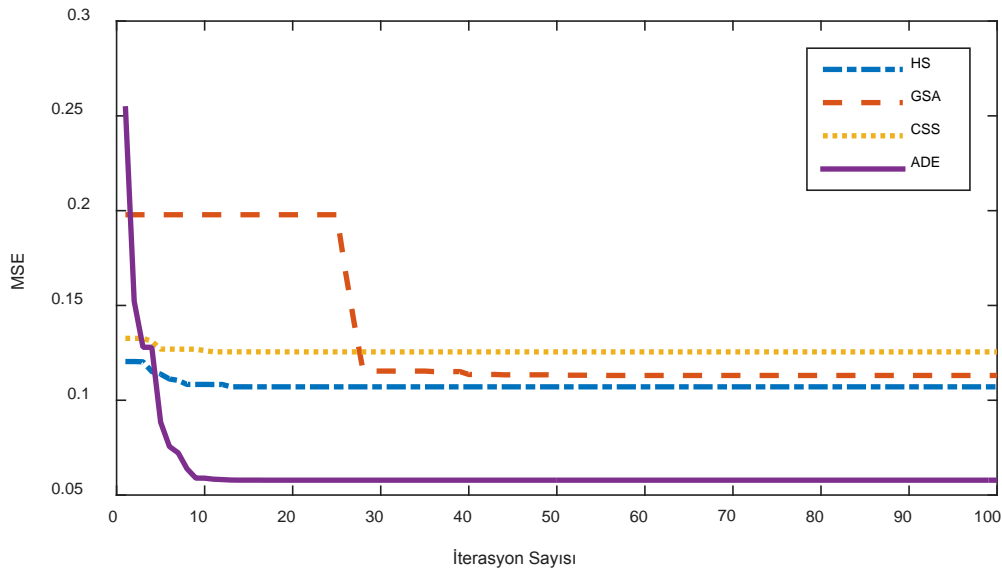
Uygulama-3 için, ADE algoritması ve diğer sezgisellere ait 30 deneme sonucundan elde edilen istatistiksel sonuçlar Tablo 5’de, literatürdeki sonuçların karşılaştırmaları ise Tablo 6’da sunulmuştur. Bu tablolardan görüldüğü gibi ADE algoritması en iyi hata değerini $5.78E-02$ seviyesinde hesaplamıştır. Şekil 6’da sunulan yakınsama eğrilerine bakıldığında da ADE algoritmasının hızlı bir yakınsama yaptığı görülmektedir. Hesaplama süresince FIR model parametrelerinin gelişimi Şekil 7’de gösterilmiştir.

Tablo 5. Uygulama-3’e ait hesaplanan hata değerleri.

Algoritmalar	En iyi	Ortalama	En kötü	Standart sapma
HS	1.07E-01	1.17E-01	1.41E-01	3.78E-09
GSA	1.13E-01	1.53E-01	1.55E-01	5.37E-09
CSS	1.25E-01	1.60E-01	1.82E-01	4.17E-07
ADE	5.78E-02	7.24E-02	9.24E-02	5.76E-10

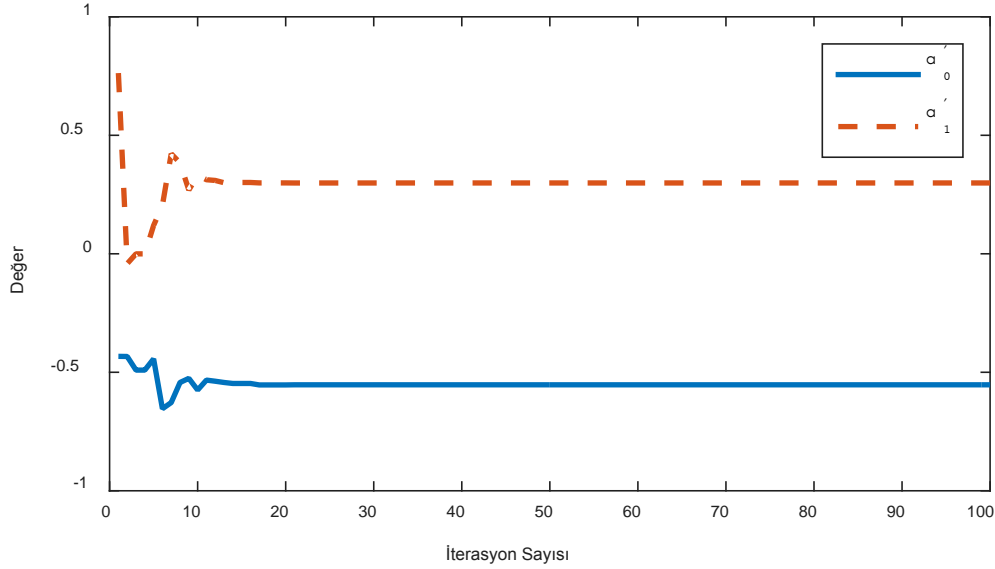
Tablo 6. Uygulama-3’e ait en iyi hata değerleri.

Algoritmalar	MSE	MSE(dB)
RLS [23]	1.31E-01	-8.82
DEA [23]	1.31E-01	-8.82
HS	1.07E-01	-9.70
GSA	1.13E-01	-9.46
CSS	1.25E-01	-9.03
ADE	5.78E-02	-12.38



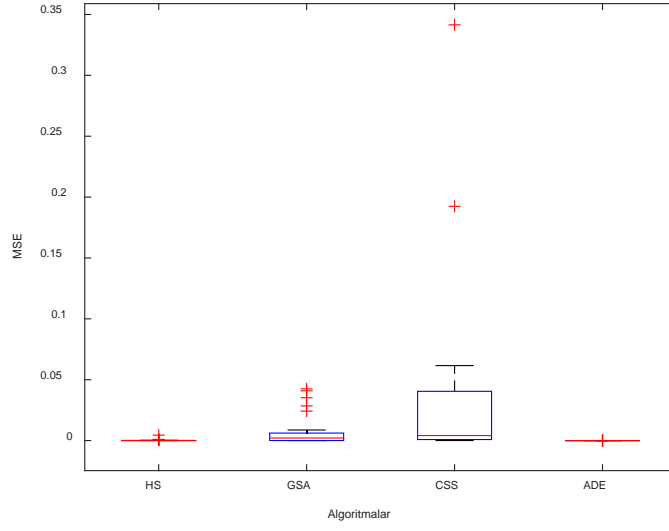
Şekil 6. Uygulama-3’e ait yakınsama eğrileri.

Bilineer sisteme ait parametre kestiriminin yapıldığı örnek uygulamalarda, önerilen ADE algoritmasının hem HS, GSA, CSS gibi sezgisel algoritmalara hem de literatürde rapor edilmiş diğer metotlara kıyasla daha düşük MSE değerlerinde çözüm sunduğu görülmektedir. Bu durumda, bilineer sisteme ait gerçek parametre değerlerine daha yakın sonuçlar üretilmiş ve bilinmeyen sisteme ait kimliklendirme işleminde doğruluk oranı artırılmıştır. İlaveten ADE algoritmasının sahip olduğu yüksek yakınsama oranı, örneklere ait yakınsama eğrilerinde kendini göstermiştir. Az sayıdaki iterasyon ile global çözüm değerlerine ulaşılması, algoritmanın hesaplama gücünü ortaya koymaktadır.

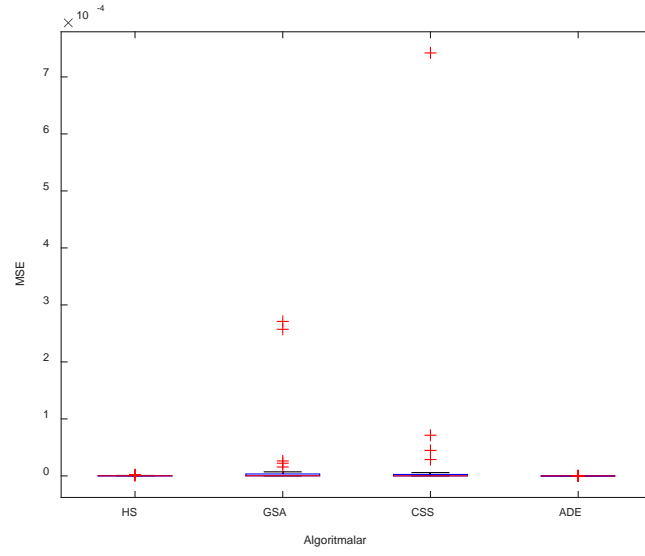


Şekil 7. Uygulama-3'e ait parametre gelişimi.

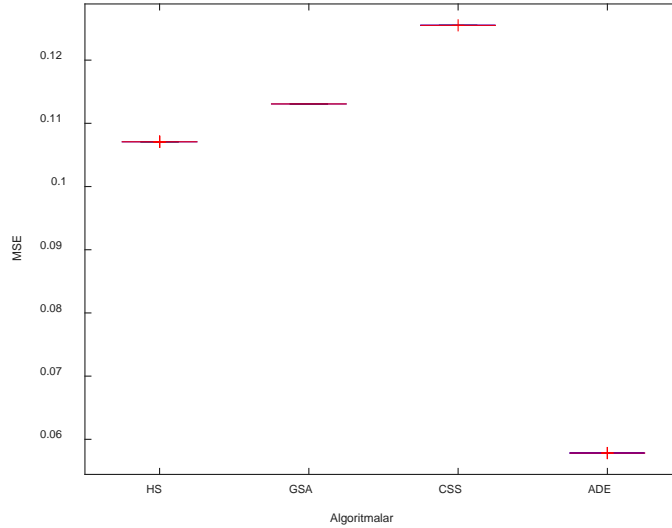
Şekil 8, 9 ve 10'da örnek uygulamalar için 30 denemeye ait kutu-grafik çizimleri gösterilmiştir. Şekiller incelendiğinde, ADE ve HS algoritmaları ile üretilen en iyi, en kötü ve ortanca çözümlerin diğer algoritmalara kıyasla daha dar bir aralıkta yer aldığı görülmektedir. Bu sonuç, üretilen çözümlerin birbirine yakın değerlerde olduğunu ve aday çözümlerin aynı çözüm noktasına yöneldiğini ifade etmektedir. Diğer bir deyişle, algoritmaların kararlılıklarını göstermektedir. Ayrıca, ADE algoritmasının yeni birey oluşturma aşamasında, mevcut popülasyondaki ortalama vektörünün kullanılması, bu sonucun elde edilmesinin doğal bir sonucu şeklinde yorumlanabilir.



Şekil 8. Uygulama-1'e ait kutu-grafik çizimi.



Şekil 9. Uygulama-2'ye ait kutu-grafik çizimi.



Şekil 10. Uygulama-3'e ait kutu-grafik çizimi.

5. Sonuçlar ve tartışma

ADE tabanlı bilinear sistem kimliklendirme probleminin gerçekleştirildiği bu çalışmada, bilinmeyen sistem parametreleri bilinear model üzerinden kestirilmiştir. Önerilen algoritmanın performansı hem farklı hesaplama stratejilerine sahip sezgisel algoritmalar ile hem de literatürde rapor edilmiş diğer metotlar ile karşılaştırılmıştır. Yapılan karşılaştırmalarda, önerilen algoritmanın bilinmeyen sistem parametrelerini daha doğru kestirdiği ve hata değerlerini azalttığı görülmüştür. Daha az sayıda iteratif hesaplamaya ulaşılması ve hızlı bir yakınsamanın gerçekleşmesi, önerilen algoritmanın hesapsal gücünü ortaya koymaktadır. Sonuç olarak, önerilen algoritma parametre kestiriminin yapıldığı diğer mühendislik uygulamaları için ümit vaat etmektedir.

Kaynaklar

- [1] Chinarro, D., **System engineering applied to fuenmayor karst aquifer and collins glacier**, Springer International Publishing, Switzerland, (2014).
- [2] Lin, J., Chen, C., Parameter estimation of chaotic systems by an oppositional seeker optimization algorithm, **Nonlinear Dynamics**, 76(1), 509-517, (2014).
- [3] Zhang, R.D., Lu, R.Q., Xue, A.K., Gao, F.R., Predictive functional control for linear systems under partial actuator faults and application on an injection molding batch process, **Industrial & Engineering Chemistry Research**, 53(2), 723-731, (2014).
- [4] Dai, C., Chen, W., Zhu, Y., Seeker optimization algorithm for digital IIR filter design, **IEEE Transactions on Industrial Electronics**, 57(5), 1710-1718, (2010).
- [5] Mostajabi, T., Poshtan, J., Mostajabi, Z., IIR model identification via evolutionary algorithms, **Artificial Intelligence Review**, 39, 1-15, (2013).

- [6] Luitel, B., Venayagamoorthy, G.K., Particle swarm optimization with quantum infusion for system identification, **Engineering Applications of Artificial Intelligence**, 23(5), 635-649, (2010).
- [7] Kumar, M., Kumar, T.K., Aggarwal, A., Adaptive infinite impulse response system identification using modified-interior search algorithm with Levy flight, **ISA Transactions**, 67, 266-279, (2017).
- [8] Durmuş, B., Gün, A., Parameter identification using particle swarm optimization, **Proceedings , 6th International Advanced Technologies Symposium**, 188-192, Elazığ, (2011).
- [9] Mathews, V.J., Sicuranza, G.L., **Polynomial signal processing**, Wiley, New York, (2000).
- [10] Greblicki, W., Nonlinearity estimation in hammerstein systems based on ordered observations, **IEEE Transactions on Signal Processing**, 44(5), 1224-1233, (1996).
- [11] Kristinsson, K., Dumont, G.A., System identification and control using genetic algorithm, **IEEE Transactions on Systems, Man, and Cybernetics**, 22(5), 1033-1046, (1992).
- [12] Kuo, S.M., Wu, H.T., Nonlinear adaptive bilinear filters for active noise control systems, **IEEE Transactions on Circuits and Systems**, 52(3), 617-624, (2005).
- [13] Kalouptsidis, N., Koukoulas, P., Mathews, V.J., Blind identification of bilinear systems, **IEEE Transactions on Signal Processing**, 51, 484-499, (2003).
- [14] Wang, Z., Gu, H., Parameter identification of bilinear system based on genetic algorithm, **Lecture Notes in Computer Science**, 4688, 83-91, (2007).
- [15] Modares, H., Alfi, A., Sistani, M.N., Parameter estimation of bilinear systems based on an adaptive particle swarm optimization, **Engineering Applications of Artificial Intelligence**, 23, 1105-1111, (2010).
- [16] Kawaria, N., Patidar, R., George, N.V., Parameter estimation of MIMO bilinear systems using a Levy shuffled frog leaping algorithm, **Soft Computing**, 21, 3849-3858, (2017).
- [17] Durmuş, B., Optimal components selection for active filter design with average differential evolution algorithm, **International Journal of Electronics and Communications**, 94, 293-302, (2018).
- [18] Geem, Z.W., Kim, J.H., Loganathan, G.V., A new heuristic optimization algorithm: harmony search, **Simulation**, 76(2), 60-68, (2001).
- [19] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., GSA: A gravitational search algorithm, **Information Sciences**, 179(13), 2232-2248, (2009).
- [20] Kaveh, A., Talatahari, S., A novel heuristic optimization method: charged system search, **Acta Mechanica**, 213(3-4), 267-289, (2010).
- [21] Lee, S.H., Kong, J.S., Seo, J.H., 1997. Observers for bilinear systems with unknown inputs and application to superheater temperature control, **Control Engineering Practice**, 5(4), 493-506, (1997).
- [22] Özer, Ş., Zorlu, H., Identification of bilinear systems using differential evolution algorithm, **Sadhana**, 36(3), 281-292, (2011).
- [23] Mete, S., Özer, Ş., Zorlu, H., System identification using Hammerstein model optimized with differential evolution algorithm, **International Journal of Electronics and Communications**, 70, 1667-1675, (2016).