

BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
BİLİMSEL ARAŞTIRMA PROJESİ SONUÇ RAPORU

**YAPAY ARI KOLONİ (ARTIFICIAL BEE COLONY, ABC)
ALGORİTMASININ BULANIK SİSTEM TANIMA VE
MODELLEMEDE BAŞARIMININ İNCELENMESİ**

PROJE YÜRÜTÜCÜSÜ : Yrd. Doç. Dr. Cihan KARAKUZU

**ARAŞTIRMACILAR : Yrd. Doç. Dr. Uğur Yüzgeç
Yrd. Doç. Dr. Metin Kesler
Yüksek Lisans Öğrencisi Özlem YILDIRIM**

BAŞLAMA TARİHİ: 01. 01. 2011

BİTİŞ TARİHİ: 30.06.2012

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
BİLECİK, 2012



BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
BİLİMSEL ARAŞTIRMA PROJESİ SONUÇ RAPORU

**YAPAY ARI KOLONİ (ARTIFICIAL BEE COLONY, ABC)
ALGORİTMASININ BULANIK SİSTEM TANIMA VE
MODELLEMEDE BAŞARIMININ İNCELENMESİ**

PROJE YÜRÜTÜCÜSÜ : Yrd. Doç. Dr. Cihan KARAKUZU

**ARAŞTIRMACILAR : Yrd. Doç. Dr. Uğur Yüzgeç
Yrd. Doç. Dr. Metin Kesler
Yüksek Lisans Öğrencisi Özlem YILDIRIM**

BAŞLAMA TARİHİ: 01.01.2011

BİTİŞ TARİHİ: 30.06.2012

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
BİLECİK, 2012

ABSTRACT

INVESTIGATION OF ARTIFICIAL BEE COLONY (ABC) ALGORITHM PERFORMANCE ON FUZZY SYSTEM IDENTIFICATION AND MODELLING

Artificial Bee Colony (ABC) algorithm, defined by Derviş Karaboğa in 2005 by inspiring behavior of honey bees, is a population-based optimization algorithm. In ABC algorithm, artificial bees fly around in a multidimensional search space and adjust their position in the space depending on experience sharing among them. Since 2005, ABC has been applied on function optimization, clustering and artificial neural network training. The algorithm has been compared with GA, PSO and DE algorithms based on mostly known benchmark functions in the literature for problem of finding the global minimum points and it has shown that ABC gave good results. ABC has not been applied on problem of system identification and modeling based on fuzzy logic so far. The main objective of this project is to eliminate this deficiency in the literature. In this project, the ABC's performance in solving the problem of fuzzy logic-based system identification and modeling is examined. ABC algorithm is operated for parameter optimization of a fuzzy-neural system to be used for system identification and modeling. For this purpose, five non-linear dynamic benchmark systems often used in the literature are used. The obtained results are compared with the that of other well-known its competitor algorithms (PSO, DE) as a versatile.

Keywords: Artificial bee colony, fuzzy logic, system identification, evolutionary learning, Heuristic Optimization

ÖZET

YAPAY ARI KOLONİ (ARTIFICIAL BEE COLONY, ABC) ALGORİTMASININ BULANIK SİSTEM TANIMA VE MODELLEMEDE BAŞARIMININ İNCELENMESİ

Yapay Arı Koloni (YAK) algoritması 2005 yılında Dervis Karaboğa tarafından bal arılarının davranışından esinlenerek geliştirilmiş popülasyon tabanlı bir optimizasyon algoritmasıdır. YAK algoritmasında yapay arılar çok boyutlu arama uzayında uçarlar ve birbirleri arasındaki deneyim paylaşımına dayalı olarak uzaydaki pozisyonlarını belirlerler. 2005'den beri algoritma fonksiyon optimizasyonu, kümeleme ve yapay sinir ağları eğitiminde kullanılmıştır. Algoritma, literatürde çok bilinen denektaşı (benchmark) fonksiyonların küresel minimum noktasının bulunması problemi baz alınarak, GA, PSO, DE algoritmalarıyla kıyaslanmış ve daha iyi sonuç verdiği gösterilmiştir. YAK algoritması şimdiye kadar bulanık mantık tabanlı sistem tanıma ve modelleme problemine uygulanmamıştır. Bu projenin temel konusu literatürdeki bu eksikliği gidermektir. Bu projede, YAK'ın bulanık mantık tabanlı sistem tanıma ve modelleme probleminin çözümünde başarımı incelenmiştir. YAK algoritması sistem tanıma ve modelleme amaçlı kullanılacak bulanık-nöral bir sistemin parametrelerinin optimize edilmesinde kullanılmıştır. Bu amaçla literatürde sıkça kullanılan doğrusal olmayan beş adet denektaşı dinamik sistem kullanılmıştır. Elde edilen sonuçlar bilinen diğer rakip algoritmaların (PSO, DE) sonuçları ile çok yönlü olarak karşılaştırılmıştır.

Anahtar Kelimeler: Yapay arı koloni, bulanık mantık, sistem tanıma, evrimsel öğrenme, sezgisel optimizasyon

İÇİNDEKİLER

ABSTRACT	i
ÖZET	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	iv
ÇİZELGELER DİZİNİ	vi
KISALTMALAR VE GÖSTERİMLER	7
1. GİRİŞ.....	8
2. I. ARA DÖNEMDE YAPILAN ÇALIŞMALAR	10
2.1 ABC ALGORİTMASI İLE FONKSİYON OPTİMİZASYONU	11
2.2 ABC ALGORİTMASI İLE PID DENETLEYİCİ PARAMETRE OPTİMİZASYONU VE DİĞER ALGORİTMALARLA KARŞILAŞTIRILMASI.....	12
2.3 SONUÇ VE DEĞERLENDİRME.....	15
3. II. ARA DÖNEMDE YAPILAN ÇALIŞMALAR.....	16
3.1 BULANIK SİSTEM YAPISI VE ÖRNEK DİNAMİK SİSTEMLER.....	16
3.2 GELİŞTİRİLEN ABC ALGORİTMALARI	18
3.3 ABC ALGORİTMALARI BAŞARIM SONUÇLARI	18
4. III. ARA DÖNEMDE YAPILAN ÇALIŞMALAR.....	36
4.1 PARÇACIK SÜRÜ ALGORİTMASI (PSO) İLE YAPILAN ÇALIŞMALAR	36
4.2 ABC VE PSO ALGORİTMALARI İLE ELDE EDİLEN SONUÇLARIN DEĞERLENDİRİLMESİ	52
4.3 FARKSAL GELİŞİM (DIFFERENTIAL EVOLUTION, DE) ALGORİTMASI İLE YAPILAN ÇALIŞMALAR	53
4.2. ABC VE DE ALGORİTMALARI İLE ELDE EDİLEN SONUÇLARIN DEĞERLENDİRİLMESİ	62
5. SONUÇ.....	64
6. BİLİMSEL ETKİNLİKLER	64
KAYNAKLAR.....	67
EK-1: ASYU 2012'DE SUNULAN BİLDİRİ	68

ŞEKİLLER DİZİNİ

Şekil 1. “Peaks” fonksiyonun tanımladığı yüzeyden iki görünüm.....	11
Şekil 2. Algoritmanın son adımında arıların yüzeydeki konumları.....	12
Şekil 3. DA motor modeli öbek şeması [2].....	12
Şekil 4. Açısal hız denetiminde için kullanılan ön ayar (set) değerleri değişimi.....	13
Şekil 5. ABC algoritmasının koşturulması sonucu bulunan K_p , K_d , K_i değerleri.....	14
Şekil 6. İki girişli-tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi.....	16
Şekil 7. ÖDS sistemleri için sistem tanımanın eğitim fazında kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için [-1 1] aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için [-5 5] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için [-2 2] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe.....	19
Şekil 8. ABC öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.....	20
Şekil 9. ABC ile eğitim fazında ÖDS 1 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları.....	21
Şekil 10. ABC öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları.....	22
Şekil 11. ABC ile eğitim fazında ÖDS 2 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları.....	23
Şekil 12. ABC öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları.....	24
Şekil 13. ABC ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları.....	25
Şekil 14. ABC öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları.....	26
Şekil 15. ABC ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları.....	27
Şekil 16. ABC öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları.....	28
Şekil 17. ABC ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları.....	29
Şekil 18. ÖDS sistemleri için sistem tanımanın test fazında kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için [-1 1] aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için [-5 5] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için [-2 2] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe.....	30
Şekil 19. ABC ile ÖDS 1 için elde edilen ANFIS modelinin test sonuçları.....	31
Şekil 20. ABC ile ÖDS 2 için elde edilen ANFIS modelinin test sonuçları.....	32
Şekil 21. ABC ile ÖDS 3 için elde edilen ANFIS modelinin test sonuçları.....	33
Şekil 22. ABC ile ÖDS 4 için elde edilen ANFIS modelinin test sonuçları.....	34
Şekil 23. ABC ile ÖDS 5 için elde edilen ANFIS modelinin test sonuçları.....	35
Şekil 24. PSO öğrenmeli ANFIS ile ÖDS 1 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	37
Şekil 25. PSO öğrenmeli ANFIS ile ÖDS 1 için <i>TEST</i> fazı sistem tanıma sonuçları.....	38

Şekil 26. PSO ile eğitim fazında ÖDS 1 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	39
Şekil 27. PSO öğrenmeli ANFIS ile ÖDS 2 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları	40
Şekil 28. PSO öğrenmeli ANFIS ile ÖDS 2 için <i>TEST</i> fazı sistem tanıma sonuçları	41
Şekil 29. PSO ile eğitim fazında ÖDS 2 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	42
Şekil 30. PSO öğrenmeli ANFIS ile ÖDS 3 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları	43
Şekil 31. PSO öğrenmeli ANFIS ile ÖDS 3 için <i>TEST</i> fazı sistem tanıma sonuçları	44
Şekil 32. PSO ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	45
Şekil 33. PSO öğrenmeli ANFIS ile ÖDS 4 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları	46
Şekil 34. PSO öğrenmeli ANFIS ile ÖDS 4 için <i>TEST</i> fazı sistem tanıma sonuçları	47
Şekil 35. PSO ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	48
Şekil 36. PSO öğrenmeli ANFIS ile ÖDS 5 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları	49
Şekil 37. PSO öğrenmeli ANFIS ile ÖDS 5 için <i>TEST</i> fazı sistem tanıma sonuçları	50
Şekil 38. PSO ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	51
Şekil 39. Klasik Farksal Gelişim Algoritmasının sözde kodu.....	54
Şekil 40. DE öğrenmeli ANFIS ile ÖDS 1 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	55
Şekil 41. DE öğrenmeli ANFIS ile ÖDS 1 için <i>TEST</i> fazı sistem tanıma sonuçları	55
Şekil 42. DE ile eğitim fazında ÖDS 1 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	56
Şekil 43. DE öğrenmeli ANFIS ile ÖDS 2 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	56
Şekil 44. DE öğrenmeli ANFIS ile ÖDS 2 için <i>TEST</i> fazı sistem tanıma sonuçları	57
Şekil 45. DE ile eğitim fazında ÖDS 2 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	57
Şekil 46. DE öğrenmeli ANFIS ile ÖDS 3 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	58
Şekil 47. DE öğrenmeli ANFIS ile ÖDS 3 için <i>TEST</i> fazı sistem tanıma sonuçları	58
Şekil 48. DE ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	59
Şekil 49. DE öğrenmeli ANFIS ile ÖDS 4 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	59
Şekil 50. DE öğrenmeli ANFIS ile ÖDS 4 için <i>TEST</i> fazı sistem tanıma sonuçları	60
Şekil 51. DE ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	60
Şekil 52. DE öğrenmeli ANFIS ile ÖDS 5 için <i>EĞİTİM</i> fazı sistem tanıma sonuçları.....	61
Şekil 53. DE öğrenmeli ANFIS ile ÖDS 5 için <i>TEST</i> fazı sistem tanıma sonuçları	61
Şekil 54. DE ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları	62

ÇİZELGELER DİZİNİ

Tablo 1. DA motoru parametreleri	13
Tablo 2. Algoritmaların 100 kez koşturulma sonucu başarımların kıyaslama ölçütleri	15
Tablo 3. ABC öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS)	17
Tablo 4. Her bir ÖDS için kullanılan ANFIS yapısı	17
Tablo 5. ABC parametreleri	19
Tablo 6. PSO parametreleri	36
Tablo 7. Eğitim fazı için ABC ve PSO algoritmaları kıyaslama tablosu	52
Tablo 8: Test fazı için ABC ve PSO algoritmaları kıyaslama tablosu	53
Tablo 9. DE algoritmasında kullanılan parametre değerleri.....	55
Tablo 10. Eğitim fazı için ABC ve DE algoritmaları kıyaslama tablosu	63
Tablo 11. Test fazı için ABC ve DE algoritmaları kıyaslama tablosu	63
Tablo 12. ABC, PSO ve DE algoritmalarının ÖDS modellemede <i>EĞİTİM</i> fazı kıyaslaması	65
Tablo 13. ABC, PSO ve DE algoritmalarının ÖDS modellemede <i>TEST</i> fazı kıyaslaması	66

KISALTMALAR ve GÖSTERİMLER

ABC	: Artificial Bee Colony
ANFIS	: Adaptive Neuro-Fuzzy Inference system
DA	: Doğru Akım
DE	: Differential Evolution
GA	: Genetik Algoritma
G-ABC	: Geliştirilmiş ABC
G-PSO	: Geliştirilmiş PSO
Kd	: Türevsel kazanç
Ki	: İntegral kazancı
Kp	: Oransal kazanç
OBS	: Ortalama Başarım Sayısı
ÖDS	: Örnek Dinamik Sistem
PID	: Proportional Integral Derivative
PSO	: Parçacık Sürü Optimizasyon (Particle Swarm Optimization)
TACO	: Touring Ant Colony Optimization
ÜF	: Üyelik Fonksiyonu
YAK	: Yapay Arı Koloni

1. GİRİŞ

Bilindiği üzere, gerçekte çoğu sistem doğrusal olmayan karmaşık bir davranış sergiler. Özellikle değişken sayılarının ve sistemin elde edilebilen modelinin derecesinin yüksek olması alışlagelmiş yöntemleri başarısız kılmaktadır. Günümüzde gelişen teknolojiye bağlı olarak çoğu zor/karmaşık mühendislik problemlerin çözümünde yeni yaklaşımlar/algortmalar türetilmiştir. Doğrusal olmayan sistem modelleme ve tanıma yapay sinir ağları ve bulanık mantığın etkin kullanımıyla birlikte bu iki popüler modelleme araçlarının parametrelerinin optimizasyonu için etkin ve karmaşık işlemler gerektirmeyen doğadaki sistemlerden esinlenerek ortaya çıkarılmış birçok algoritma geliştirilmiştir. Özellikle popülasyon tabanlı sezgisel arama algoritmaları son zamanlarda yaygın kullanım alanı bulmuşlardır.

Sezgisel arama algoritmalarının ortak noktası işleyişlerinde sürü zekâsı adı verilen kullandıkları bireylerinin çevresiyle ve diğer bireylerle olan etkileşimini kullanmalarıdır. Yapay Arı Kolonisi Algoritması (Artificial Bee Colony, ABC) bu tip algoritmaların en güncel olanlarından biridir. Bu algoritma, bal arılarının zengin çiçek nektarları bulmak için öncelikle kâşif arıyı çevreye gönderip çiçek nektarı olan alanları tespit ederek kovana geri dönmeleri ve kovandaki işçi arılara farklı dans çeşitleriyle nektar kaynağının uzaklık ve verimlilik bilgisini vermelerinden esinlenerek 2005 yılında Kayseri Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Prof. Dr. Derviş Karaboğa tarafından [1] ile bilimsel camiaya tanıtılmıştır.

ABC algoritması dışında doğadaki sürülerin yiyecek bulma davranışlarından yola çıkarak modellenen birçok başarılı algoritmalar bulunmaktadır. Bu algoritmaların başlıcaları şunlardır: 70'li yıllarda Holland tarafından tanımlanmış ve 80'li yıllarda Golberg tarafından geliştirilmiş Genetik Algoritma (GA), Marco Dorigo tarafından 1992'de özellikle mühendislik alanında sürekli optimizasyon problemleri için önerilmiş karınca kolonisi algoritması, 1995 yılında J.Kennedy ve R.C.Eberhart tarafından geliştirilmiş parçacık sürü optimizasyonu (PSO), Price ve Storn tarafından 1995 yılında geliştirilmiş farksal gelişim (differential evolutionary, DE)'dir.

Bu projede bilimsel camiada henüz yeni olan Yapay Arı Kolonisi (YAK) (Artificial Bee Colony, ABC) algoritması kullanılarak bulanık sistem tanıma ve modelleme üzerine odaklanılmıştır. Yapılan literatür taramasında, algoritmanın 2005'den beri fonksiyon optimizasyonu, kümeleme ve yapay sinir ağları eğitiminde kullanılmış olmasına rağmen henüz bulanık mantık tabanlı sistem tanıma ve modelleme probleminde parametre optimizasyonuna uygulanmadığı görülmüştür. Bu projede bulanık işleyişi gerçekleyen ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık mantık modeli kullanılarak; ABC'nin doğrusal olmayan dinamik sistemlerin bulanık modellenmesinde optimizasyon başarımı, popüler ve sıklıkla kullanılan PSO ve DE'nin başarımlarıyla kıyaslamalı olarak incelenmiştir. Raporun takip eden bölümleri bu çalışmaları açıklamaktadır.

2. I. ARA DÖNEMDE YAPILAN ÇALIŞMALAR

Proje ile ilgili olarak literatür taraması yapılmış ve ABC algoritması ile ilgili yapılan çalışmalar incelenmiştir. Özellikle ilk iki ay için öngörülen bu zaman diliminde, ABC'nin işleyişi incelenmiş, koda dökülmesi ve projede öngörülen amaç için kullanımına yönelik ön çalışmalar yapılmıştır. ABC algoritmasının temel adımları aşağıdaki gibi özetlenebilir:

- 1:Başlangıç yiyecek kaynağı bölgelerinin üretilmesi
- 2: *Çevrim sayısı=Maksimum* çevrim sayısı oluncaya değin **Tekrarla** {
- 3:İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi
- 4: Olasılıksal seleksiyonda kullanılacak olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması
- 5: Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri
- 6:Kaynağı bırakma kriteri: Limit ve kâşif arı üretimi }

Bu genel adımlar aşağıda verilen sözde kod çerçevesinde MATLAB'da kodlanmıştır.

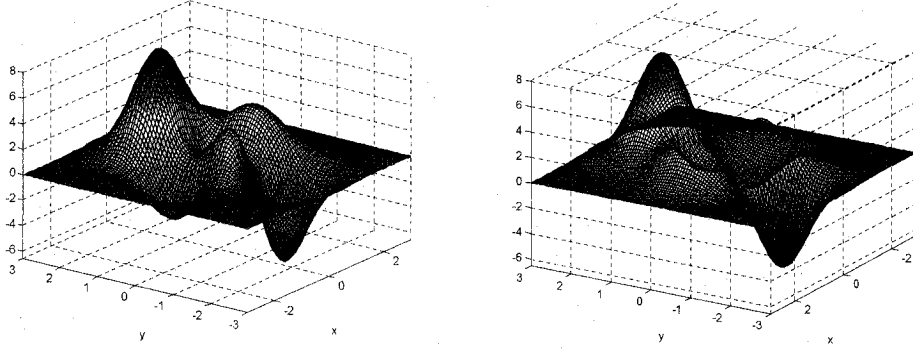
- Problemin parametrelerini ilk koşulla
- YAK algoritmasının parametrelerini ata
- Başlangıç işçi arı çözümlerini yapılandır
- Her bir yapay arı için uygunluk değeri (fit) hesapla
- $K=0$
- **(K=Max iterasyon sayısı)** oluncaya değin**Tekrarla1**
 - $N=0$
 - **(N=işçi arı sayısı)** oluncaya değin **Tekrarla2**
 - $k= i$ 'nin komşuluğundaki bir çözüm
 - $r=[-1 1]$ aralığında rasgele bir sayı
 - $x_{i,j}$ 'nin komşuluğunda aşağıdaki formülasyonla işçi arılar için yeni çözümler (yiyecek kaynakları), $u_{i,j}$, üret
 - $u_{i,j}= x_{i,j}+ r_{ij}(x_{i,j}-x_{k,j})$
 - Gözcü arılar için u_i ve x_i arasında aç gözlü seçme işlemini uygula
 - x_i çözümleri için $P_i = \frac{fit(i)}{\sum fit(i)}$ olasılık değerini hesapla
 - Olasılıklara göre gözcü arıları işçi arılara ata ve değerlendir
 - **Her gözcü arı için tekrarla**
 - u_i ve x_i arasında seçme işlemini uygula
 - **Eğer $fit(en_iyi_gözcü) < fit(işçi)$ ise**
 - İşçi arı çözümünü ilgili gözcü çözümü olarak al
 - **Eğer_son**
 - **Her gözcü arı için tekrarla_son**
 - **Eğer $fit(olasi_tüm_en_iyi_gözcü) < fit(en_iyi)$ ise**
 - En iyi gözcü çözümünü en_iyi olarak al
 - **Eğer_son**
 - $N=N+1$
- **Tekrarla2_son**
- Faydasız çözümleri belirle, yerine (kaşif arı için) yeni bir rasgele çözüm üret
 - $x_{i,j} = \min_j + rand(0,1)(\max_j - \min_j)$
- Popülasyondaki kaşif arı sayısı kadar en kötü işçi arı çözümlerini kaşif arı çözümleri ile karşılaştır. Eğer kaşif arı çözümü işçi arı çözümünden daha iyi ise, kaşif arı çözümünü ilgili işçi arı çözümü ile yer değiştir.
- $K=K+1$
- **Tekrarla1_son**

Projenin ilk altı aylık döneminde aşağıdaki alt bölümlerde açıklanan uygulamalar üzerinde çalışılarak algoritma anlaşılmasına çalışılmıştır. Bu çalışmalar projenin başlangıç uygulamalarıdır.

2.1 ABC algoritması ile fonksiyon optimizasyonu

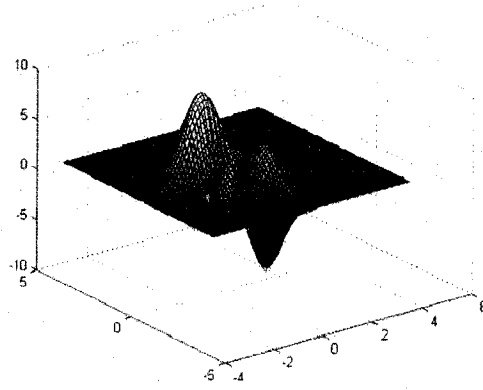
ABC algoritmasının çalışmasını anlamak ve başarısını test edebilmek için matematiksel ifadesi (1)'de verilen Şekil 1'den de görüleceği üzere birçok yerel maksimumu ve minimumu bulunan "peaks" fonksiyonu üzerinde çalışılmış algoritmaya küresel minimum buldurulmuştur.

$$f = 3(1-x)^2 e^{-(x^2-(y+1)^2)} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-(x^2+y^2)} - \frac{1}{3} e^{-((x+1)^2+y^2)} \quad (1)$$



Şekil 1. "Peaks" fonksiyonun tanımladığı yüzeyden iki görünüm

ABC algoritması kullanılarak yapılan fonksiyon optimizasyonunda; koloni büyüklüğü 50, maksimum çevrim sayısı 100, limit 30 alınmıştır. Algoritma 100 kez ayrı ayrı koşurulduğunda bulunduğu çözümlerin ortalaması için $x_1=0.2284$, $x_2=-1.6253$ olduğu gözlenmiş ve bu giriş değerlerine karşılık fonksiyonun $y=-6,5511$ sonucunu ürettiği belirlenmiştir. Bu değer Şekil 1'den de anlaşılacağı üzere fonksiyon yüzeyindeki küresel minimumdur. Şekil 2'de algoritmanın çalışmasının sonunda arıların ilgili uzaydaki pozisyonları görülmektedir. Şekilden arıların başlangıçta uzayda rasgele dağılmış olmalarına rağmen en son algoritma adımında küresel minimuma çok yakın civarda toplandıkları görülmektedir.

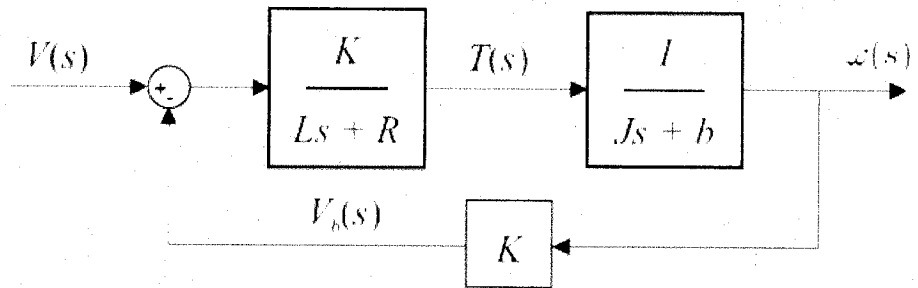


Şekil 2. Algoritmanın son adımında arıların yüzeydeki konumları

2.2 ABC algoritması ile PID denetleyici parametre optimizasyonu ve diğer algoritmalarla karşılaştırılması

Bu kısımda literatürde sıklıkla kullanılan sezgisel algoritmalar olan GA, DE, PSO ve TACO algoritmalarıyla ABC algoritmasının başarımlarını kıyaslaması DC motoru hız denetimi için PID parametrelerinin (K_p , K_d ve K_i) belirlenmesi problemi üzerinde yapılmıştır. Uygulamada kullanılan DA motorun modeli Şekil 3'de, motora ait parametreler ise Tablo 1'de verilmiştir. Bu çalışmada, gerçek bir DA motoruna ait ve Tablo 1'de verilen parametrelerle aynı zamanda sayısal çözümlenme ile sezgisel algoritmaların başarımlarını sınamıştır. Algoritmalar çalıştırılırken denetleyici için ön ayar değerleri Şekil 4'de verildiği biçimde kullanılmıştır.

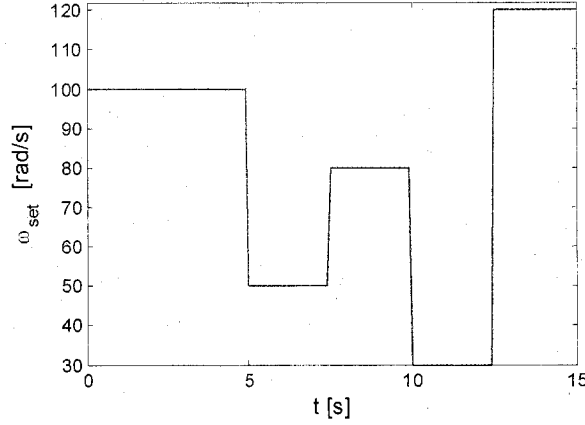
Her bir algoritma ile elde edilen sonuçlar merkezi eğilim ölçülerine dayalı olarak irdelenmiş Tablo 2'de toplu olarak verilen istatistikî sonuçlar elde edilmiştir. Tabloda merkezi eğilim ölçüleri baz alınarak, en iyi başarımları gösteren algoritmaya ait veri koyu mavi yazı tipi ile, en kötü başarımları gösteren algoritmaya ait veri kırmızı yazı tipi ile verilmiştir.



Şekil 3. DA motor modeli öbek şeması [2]

Tablo 1. DA motoru parametreleri

Motor gücü	$p=93.3 \text{ kW}$
Motor gerilimi	$v=210 \text{ V}$
Hız	1150 devir/dk
Rotor eylemsizlik momenti endüktansı	$J=2.83 \text{ kg.m}^2$
Mekanik sistem sürtünmesi	$b=1.89 \text{ Nms}$
Motor sargı endüktansı	$L=1.2 \text{ mH}$
Motor sargı direnci	$R=0.0125 \Omega$
Zıt emk sabiti	$K=1.78 \text{ N/A}$



Şekil 4. Açısal hız denetiminde için kullanılan ön ayar (set) değerleri değişimi

PID parametrelerinin optimizasyonu için sezgisel algoritmalarının bireyleri uygunluk değeri (2)'de verilen ortalama karesel hata (mse) ölçütü kullanılmıştır. Eşitlikte N bireylerin belirlediği değerlerle işleyen PID denetleç denetimi altında sayısal sistem çözümünde kullanılan toplam adım sayısını, ω_{set} motor açısal hız önayar (set) değeri, ω_a ise sistem çözümünde elde edilen açısal hız değerini ve k örnekleme indeksini göstermektedir.

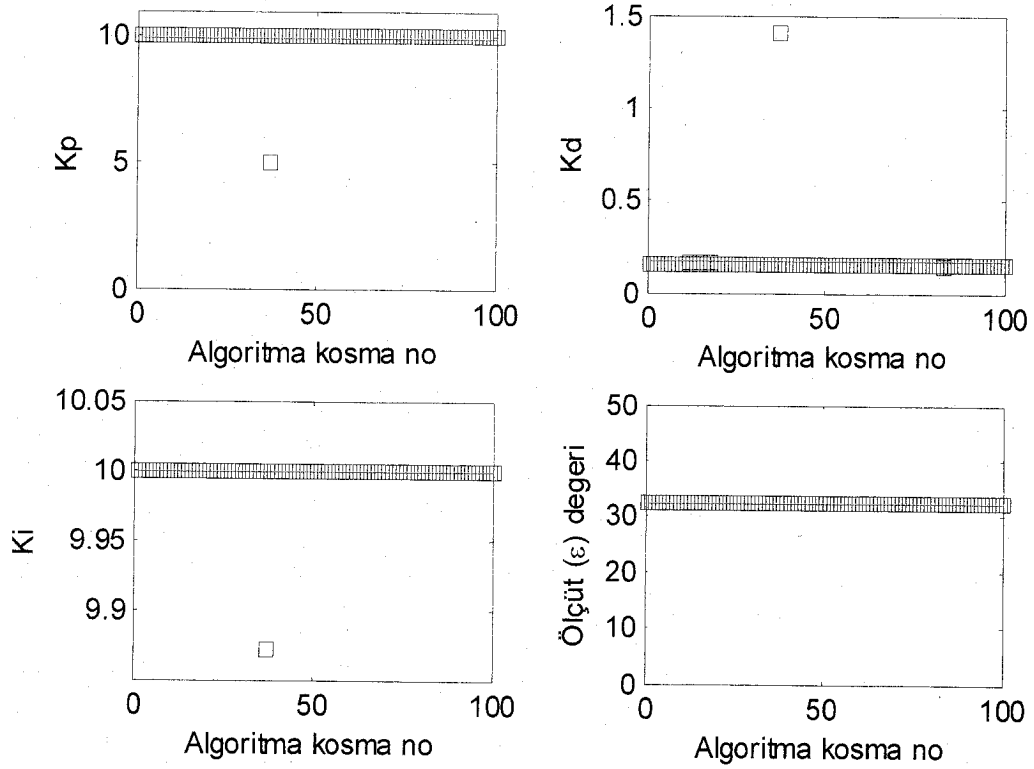
$$\varepsilon = \frac{1}{N} \sum_{k=1}^N (\omega_{set}(k) - \omega_a(k))^2 \quad (2)$$

Kullanılan sezgisel algoritmalarda sürü büyüklüğü (birey sayısı) 30, en büyük nesil sayısı olarak 100 alınmıştır. Algoritmalarda kullanılan parametreler sırasıyla; ABC için kaynağı bırakma limit değeri LC=90, GA için 18 bit gen gösterimi, mutasyon olasılık 0.5, ve çaprazlama olasılık oranı ise 0.01; DE algoritması için ölçekleme faktörü 0.5 ve çaprazlama olasılık oranı ise 0.7; PSO için $\xi=0.7298$, $c_1=c_2=2.05$, parçacık hızları parametre alt /üst sınır değerlerinin 1/5'i ile sınırlı; TACO için bireyler için 18 bit gösterim, buharlaşma oranı 0.1'dir. PID parametrelerinin sınır değerleri (3)'de tanımlandığı gibi belirlenip kullanılmıştır.

$$0 \leq K_p \leq 10, \quad 0 \leq K_d \leq 3, \quad 0 \leq K_i \leq 10 \quad (3)$$

Her bir sezgisel algoritma MATLAB programlama dilinde kodlanarak Intel(R) Core(TM) i7, Q 740 @ 1.73 GHz CPU, 6 GB RAM donanımlı, 64 bit işletim sistemine sahip bir diz üstü bilgisayarda çalıştırılmıştır. Her bir algoritma, bir önceki bölümde tanımlanan problemin

çözümünde başarımını irdelemek için 100'er kez koşturulmuştur. Her bir ayrı çalışma sonucu bulunan PID parametreleri ve en son nesilde elde edilen en iyi ölçüt değeri, ABC algoritması için Şekil 5'de verilmiştir. Diğer algoritmaların sonuçları [3]'den incelenebilir. Görüleceği üzere ABC algoritması %99 oranında hep aynı parametre değerlerini çözüm olarak işaret etmiştir.



Şekil 5. ABC algoritmasının koşturulması sonucu bulunan Kp, Kd, Ki değerleri

Her bir algoritma ile elde edilen sonuçlar merkezi eğilim ölçülerine dayalı olarak irdelenmiş Tablo 2'de toplu olarak verilen istatistikî sonuçlar elde edilmiştir. Tabloda merkezi eğilim ölçüleri baz alınarak, en iyi başarımı gösteren algoritmaya ait veri koyu mavi yazı tipi ile, en kötü başarımı gösteren algoritmaya ait veri kırmızı yazı tipi ile verilmiştir.

Tablo 2'deki verilere dayalı olarak, DE algoritmasının nesil başına gerekli koşma zamanı hariç tüm kategorilerde en iyi başarımı gösterdiği görülmektedir. ABC algoritmasının ise nesil başına gerekli koşma zamanında son derece çarpıcı üstünlüğünün yanı sıra diğer kategorilerde de DE algoritmasının en yakın rakibi olduğu değerlendirilebilir. DE ve ABC'nin ardından başarımları sıralaması olarak PSO, TACO ve GA verilebilir. Ancak, en önemli gösterge olarak görülebilecek nesil başına ortalama koşma zamanı kategorisinde sıralama ABC, DE, PSO, TACO ve GA şeklindedir.

Tablo 2. Algoritmaların 100 kez koşturulma sonucu başarımların kıyaslama ölçütleri

Kategori	Ölçüt	ABC	GA	DE	PSO	TACO
En iyi uygunluk değeri (Son nesilde bulunan)	Aritmetik ortalama	32.451355	164.068177	32.450369	60.3918143	<u>191.952581</u>
	Ort. sapma	0.001619	<u>15.770520</u>	5.46e-14	12.636871	6.510355
	Varyans	0.005299	<u>19.158736</u>	6.59e-14	16.825271	7.883210
Kp Değeri (Son nesilde bulunan)	Aritmetik ortalama	9.949624	7.626984	10	9.999825	7.039683
	Ort. sapma	0.099743	<u>0.588624</u>	6.93e-16	3.47e-04	0.256561
	Varyans	0.501230	<u>0.758087</u>	2.58e-15	0.001742	0.322914
Kd Değeri (Son nesilde bulunan)	Aritmetik ortalama	0.169251	2.521111	0.156927	0.968981	2.863651
	Ort. sapma	0.024792	0.250060	6.24e-09	<u>0.318168</u>	0.083505
	Varyans	0.124600	0.289377	7.49e-09	<u>0.423048</u>	0.108125
Ki Değeri (Son nesilde bulunan)	Aritmetik ortalama	9.998712	7.470370	10	9.9937182	7.941269
	Ort. sapma	0.002549	0.601682	3.91e-16	0.012203	<u>0.696921</u>
	Varyans	0.012811	0.735951	2.15e-15	0.046348	<u>0.849996</u>
1 nesil için gerekli süre [s]	Aritmetik ortalama	0.095165	<u>0.321317</u>	0.198012	0.215582	0.272611

2.3 Sonuç ve Değerlendirme

Bölüm 2.2'de verilen verilere dayalı olarak, ABC ve DE algoritmasının tüm kategorilerde en iyi başarımlarını gösterdiği, PSO algoritmasının ise çoğu kategoride en kötü başarımlarını sergilediği görülmektedir. Ortalama koşma zamanı kategorisinde karşılaştırıldığında en iyi koşma zamanı ABC algoritmasına aittir. Bu sonuca dayalı olarak, gerçek zamanlı uygulamalarda ABC, DE ve PSO'nun kullanımının TACO ve GA'ya göre daha elverişli olduğu anlaşılmaktadır. ABC algoritmasının diğer algoritmalara göre daha basit bir yapısı olduğu görülmüştür. ABC algoritmasının kontrol parametre sayısının az olması da algoritmaya ayrı bir üstünlük kazandırmaktadır.

Tablo 2'de verilen sonuçlar çerçevesinde yukarıda yapılan analiz sonucu, projenin bundan sonraki kısımlarında ABC algoritmasını TACO ve GA ile kıyaslamının gereksiz olduğu, kıyaslamının PSO ve DE ile yapılmasının daha anlamlı olacağı değerlendirilmiştir.

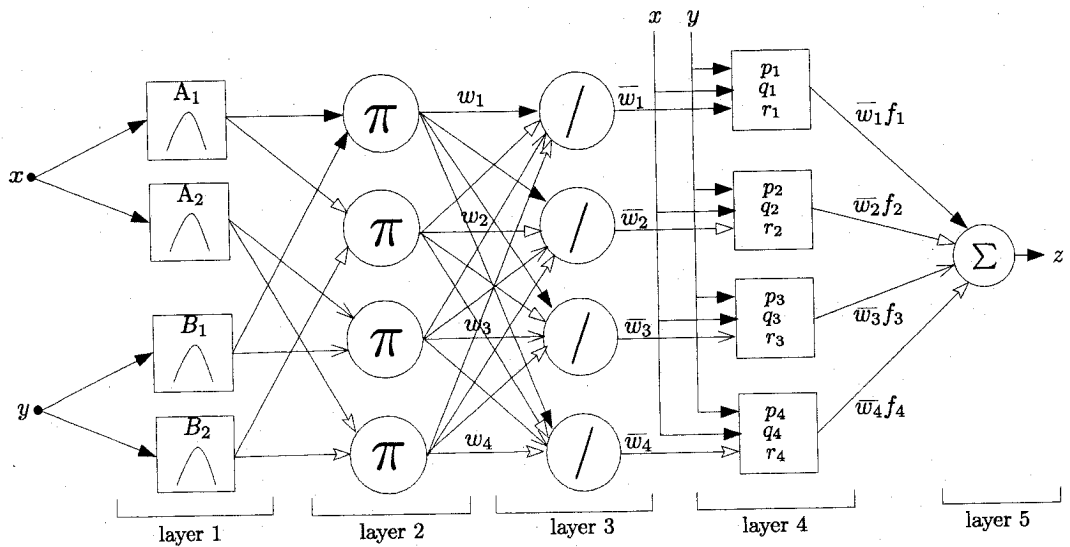
3. II. ARA DÖNEMDE YAPILAN ÇALIŞMALAR

Bu dönem içinde ABC algoritması ile eğitilen bulanık mantık tabanlı dinamik sistem tanıma ve modelleme üzerinde çalışılmıştır. Çalışmada sistem modelleme yöntemi olarak ANFIS bulanık çıkarım sistemi kullanılmıştır. Proje çalışmalarının bu aşama sonrasında kullanılacak sezgisel algoritmalar (ABC, PSO, DE) ANFIS'in parametrelerinin belirlenmesinde kullanılacaktır. Bu algoritmaların başarımları literatürden seçilen örnek dinamik sistemlerin ANFIS ile modellenmesinde ANFIS'in parametre optimizasyonu problemi üzerinde kıyaslamak projenin ana hedefidir.

3.1 Bulanık Sistem Yapısı ve Örnek Dinamik Sistemler

Modelleme için Şekil 6'da verilen ANFIS (Adaptive Neuro Fuzzy Inference System) bulanık sistemi baz alınmıştır. Bu sistemin parametreleri 1. katmandaki üyelik fonksiyonlarının ve 4. katmandaki kuralların parametreleridir. Bu parametreler uygun değerlere getirildiğinde sistem istenen $(x,y) \rightarrow z$ eşleşmesini gerçekleştirir. ANFIS ile ilgili detaylı bilgi [4]'den edinilebilir. ABC algoritmasının sistem tanıma ve modellemede sorumluluğu ANFIS'in parametrelerini ilgili girişlere istenen çıkışı verecek şekilde ayarlamaktır. ABC'nin bulanık sistem parametrelerini optimize etme başarımı literatürden seçilen ve

Tablo 3'te verilen dinamik sistemler üzerinde irdelenmiştir.



Şekil 6. İki girişli-tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi

Tablo 3. ABC öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS)

ÖDS1	$y(k) = \frac{y(k-1) \cdot y(k-2) \cdot (y(k-1) + 2.5)}{1 + y^2(k-1) + y^2(k-2)} + u(k), \quad u(k) = \cos \frac{2\pi k}{100} \quad [5]$
ÖDS 2	$y(k+1) = \frac{y(k)}{1 + y(k)^2} + u^3(k), \quad u(k) = \cos \frac{2\pi k}{100} \quad [5]$
ÖDS 3	$y(k+1) = y(k) + u(k)e^{-3 y(k) } \quad [6]$
ÖDS 4	$y(k+1) = \frac{24 + y(k)}{30} y(k) - 0.8 \frac{u(k)^2}{1 + u(k)^2} y(k-1) + 0.5u(k) \quad [7]$
ÖDS 5	$y(k+1) = \frac{1}{2} \left[\frac{y(k)}{1 + y^2(k)} + [1 + u(k)]u(k)[1 - u(k)] \right] \quad [8]$

Tablo 3'de verilen her bir dinamik sistemin modellemesi/tanınması, Şekil 6'da verilen ve Tablo 4'de tanımlanan ANFIS yapısı kullanılarak yapılmıştır. Her bir sistem için ANFIS modeli elde edebilmek ve ABC ile bu modelin parametrelerini ayarlamak amacıyla, eğitimde kullanılmak üzere Şekil 7'de verilen sistem girişleri kullanılarak birer eğitim veri seti hazırlanmıştır. ABC algoritması ile sayısı Tablo 4'ün en sağındaki sütununda verilen sayıdaki ANFIS parametreleri modelleme/tanımaya için optimize edilecektir.

Tablo 4. Her bir ÖDS için kullanılan ANFIS yapısı

ÖDS No	Girişler	Giriş ÜF sayıları	Kural sayısı	Ayarlanacak parametre sayısı
1	$u(k), y(k-2), y(k-1)$	2, 2, 2	8	36
2	$u(k), y(k), y(k-1)$	2, 2, 2	8	36
3	$u(k), y(k)$	2, 2	4	20
4	$u(k), y(k), y(k-1)$	2, 2, 2	8	36
5	$u(k), y(k)$	2, 2	4	20

3.2 Geliştirilen ABC Algoritmaları

Proje çalışmaları sırasında standart ABC algoritması modifiye edilerek iki adet geliştirilmiş ABC algoritması (G-ABC-I, G-ABC-II) tanımlanmıştır. Dönem içi çalışmalarda belirlenen dinamik sistem tanıma problemleri üzerinde üç farklı ABC algoritması ile çalışılmıştır. Bu algoritmaların kısaca tanımını aşağıda vermiştir:

- **Standart ABC algoritması:** ABC algoritması üzerinde herhangi bir değişiklik yapılmamış hali.
- **Geliştirilmiş ABC-I (G-ABC-I) :** Standart ABC algoritmasında bulunan Φ (teta) değişkeninin lineer bir şekilde azaltılması ile algorithmada bir değişiklik yapılmıştır. Geliştirilmiş ABC-I için yapılan değişikliğin kod kısmı aşağıdaki gibidir.

```
teta=(rand-0.5)*2;  
wi=(-1.4/iteS)*ite+1.5;  
v(:,j)=x(i,j)+teta*wi*(x(i,j)-x(k,j));
```

- **Geliştirilmiş ABC-II (G-ABC-II) :** Standart ABC algoritmasında bulunan Φ (teta) değişkeni lineer bir şekilde azaltılırken, küçük bir normal dağılımlı rasgele sabitin eklenmesi ile algorithmada bir değişiklik yapılmıştır. Geliştirilmiş ABC-II 'de yapılan değişikliğin kod kısmı aşağıdaki gibidir.

```
wi=(-1.4/iteS)*ite+1.5;  
v(:,j)=x(i,j)+teta*wi*(x(i,j)-x(k,j));  
katsayi=(lamda*rand(size(v)));  
v(:,j)=v(:,j)+katsayi(:,j);
```

Yukarıdaki kod bloğunda “lamda” olarak görülen ek parametre, giriş eğitim verilerinin öz değerlerine göre eşitlik (4) ile belirlenmiştir. Eşitlikte X giriş veri kümesini içeren matristir.

$$\lambda = \frac{1}{\max(\text{eig}(XX^T))} \quad (4)$$

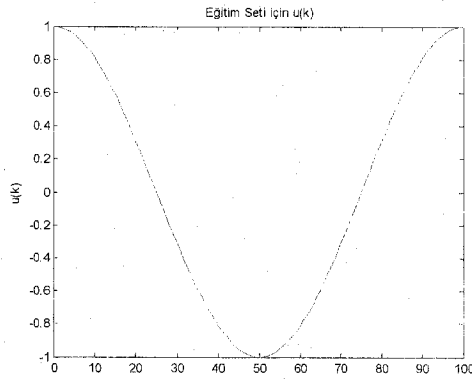
3.3 ABC Algoritmaları Başarım Sonuçları

Yukarıda kısaca tanımlanan ABC algoritmaları her bir sistem tanıma/modelleme örneği için Tablo 5'de verilen parametrelerle 50'şer kez koşturulmuştur. Eğitim fazında Şekil 7'de

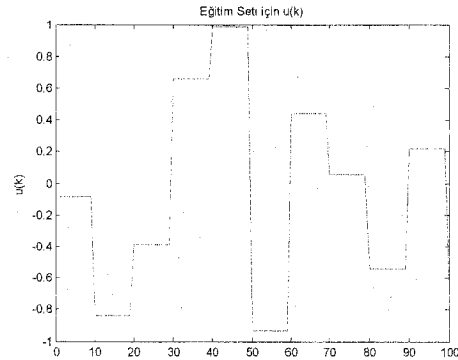
verilen giriş darbe dizileri için elde edilen sonuçlar örnek bazında grafiksel olarak Şekil 8-Şekil 17’de, test fazında Şekil 18’de verilen giriş darbe dizileri elde edilen sonuçlar örnek bazında grafiksel olarak Şekil 19-Şekil 23’de verilmiştir.

Tablo 5. ABC parametreleri

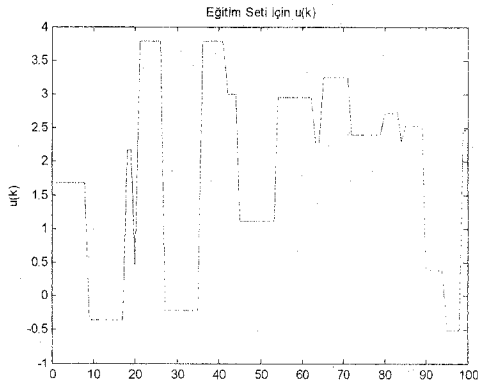
<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
<i>Koloni büyüklüğü (N)</i>	40
<i>Maksimum nesil sayısı</i>	1000
<i>Limit</i>	$(N/2).D.0.5$



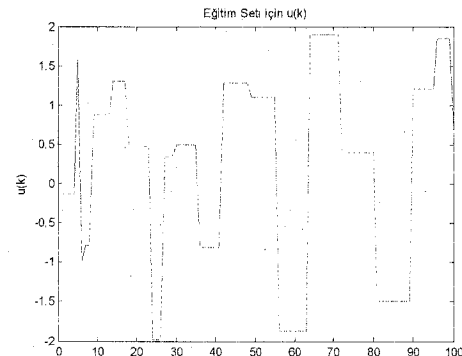
(a)



(b)

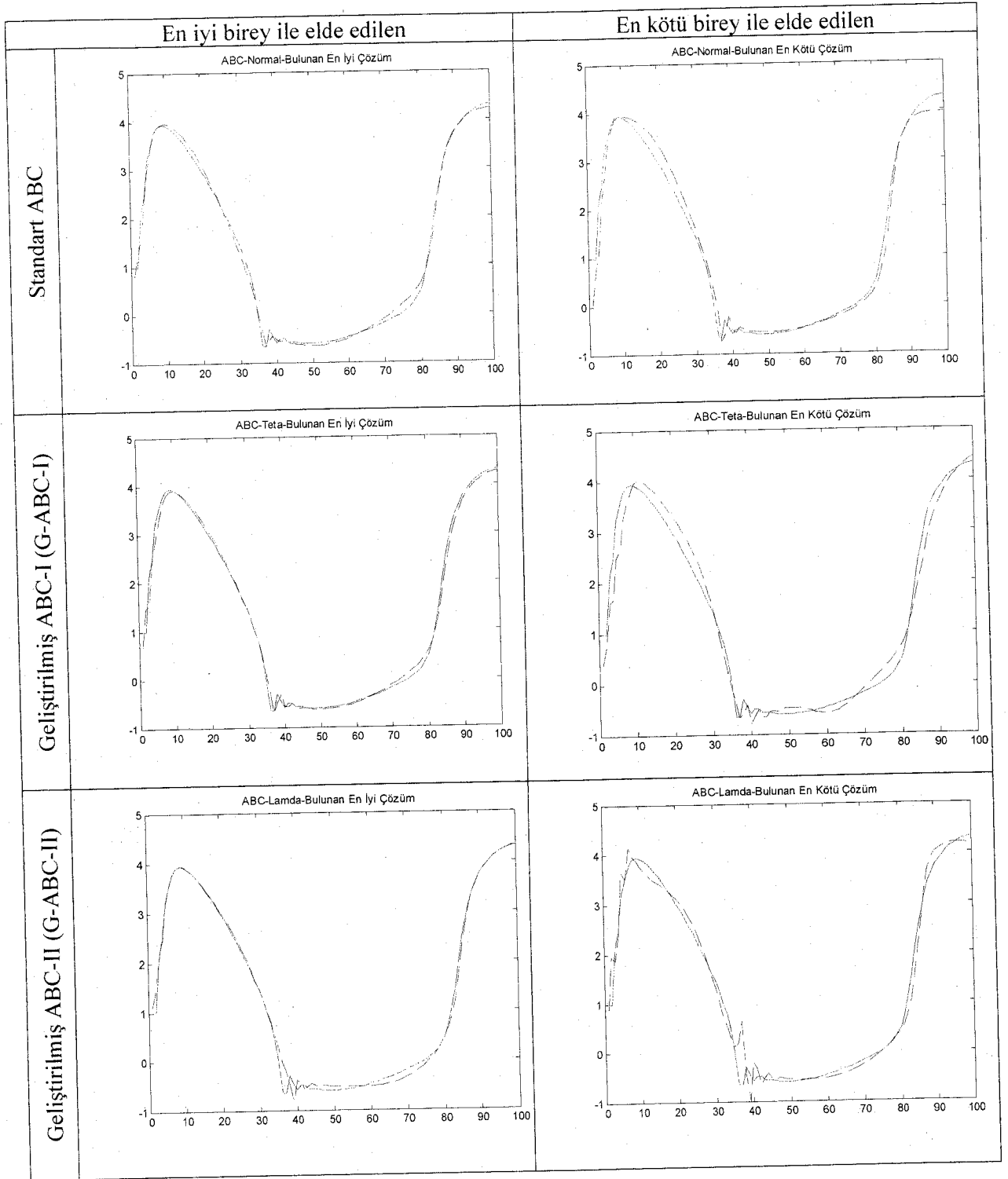


(c)

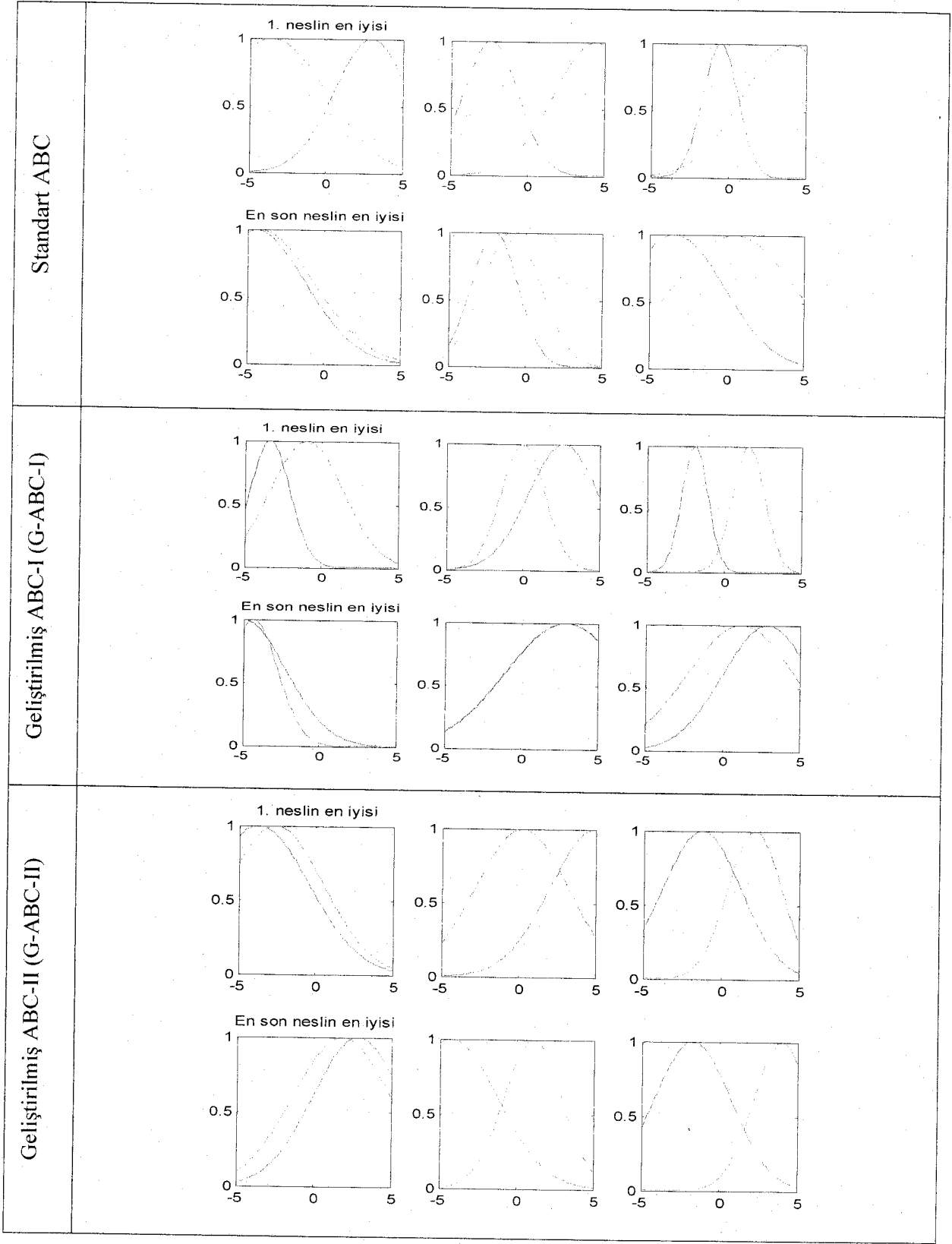


(d)

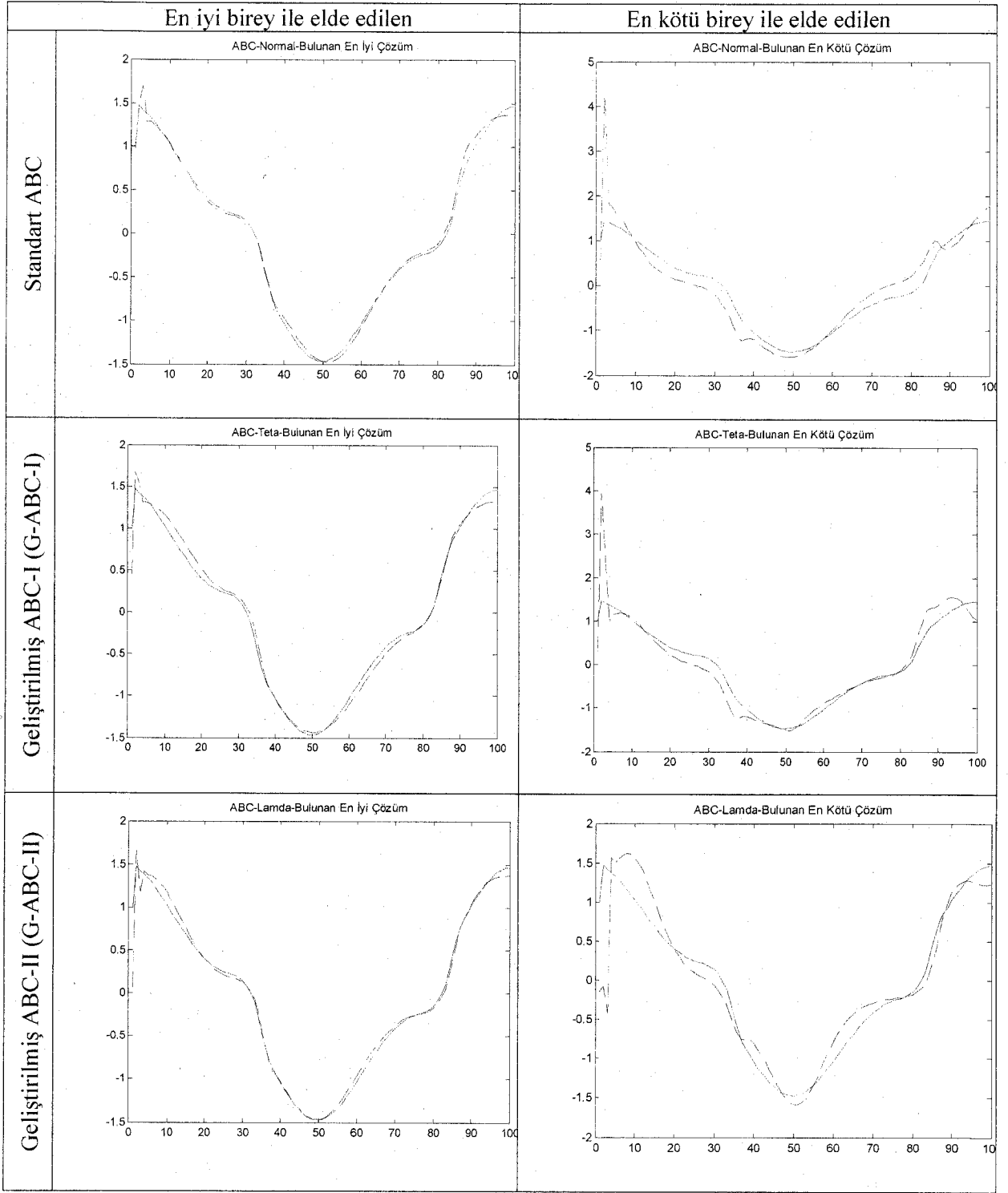
Şekil 7. ÖDS sistemleri için sistem tanımanın eğitim fazında kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için $[-1 \ 1]$ aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için $[-5 \ 5]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için $[-2 \ 2]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe



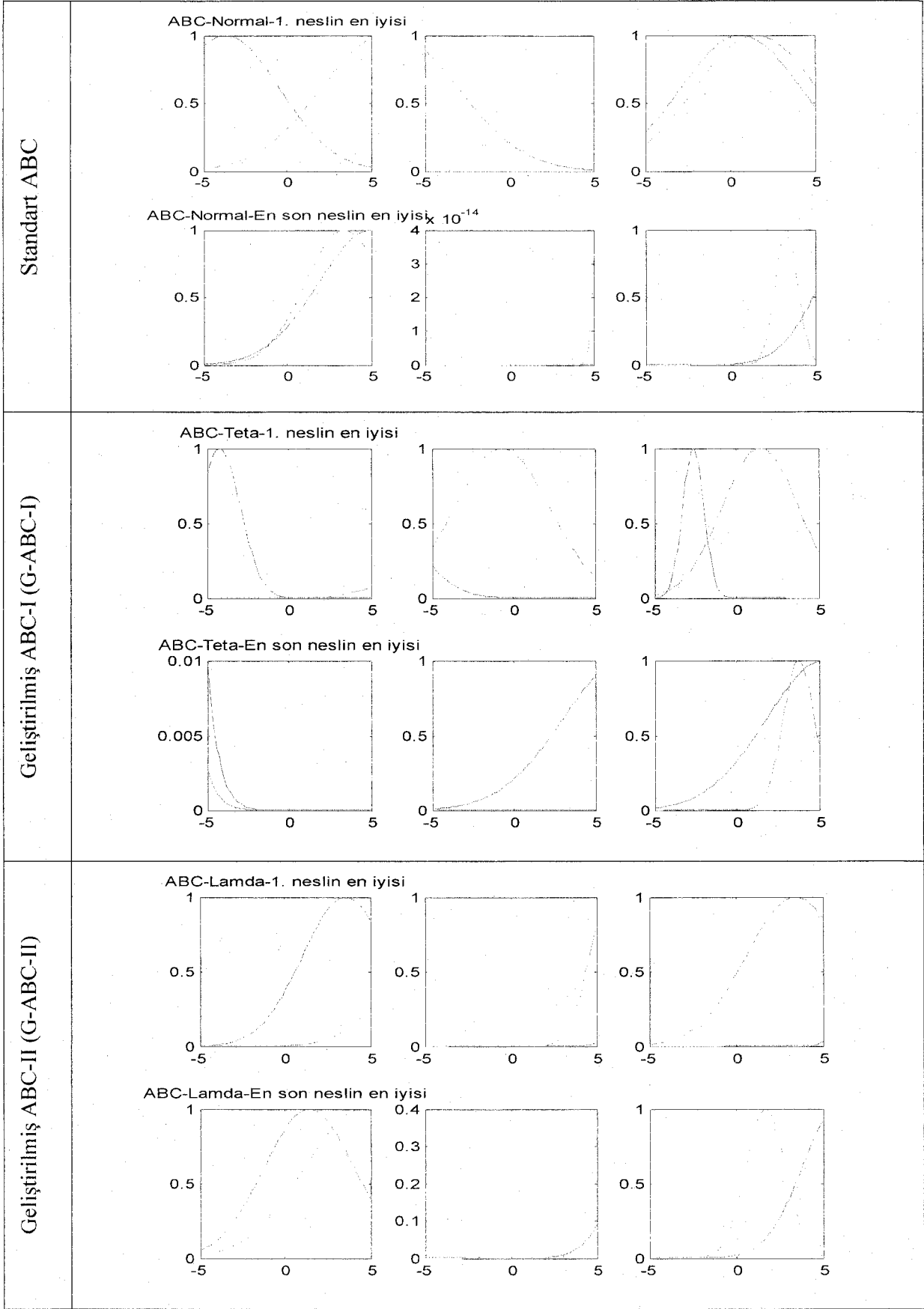
Şekil 8. ABC öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları



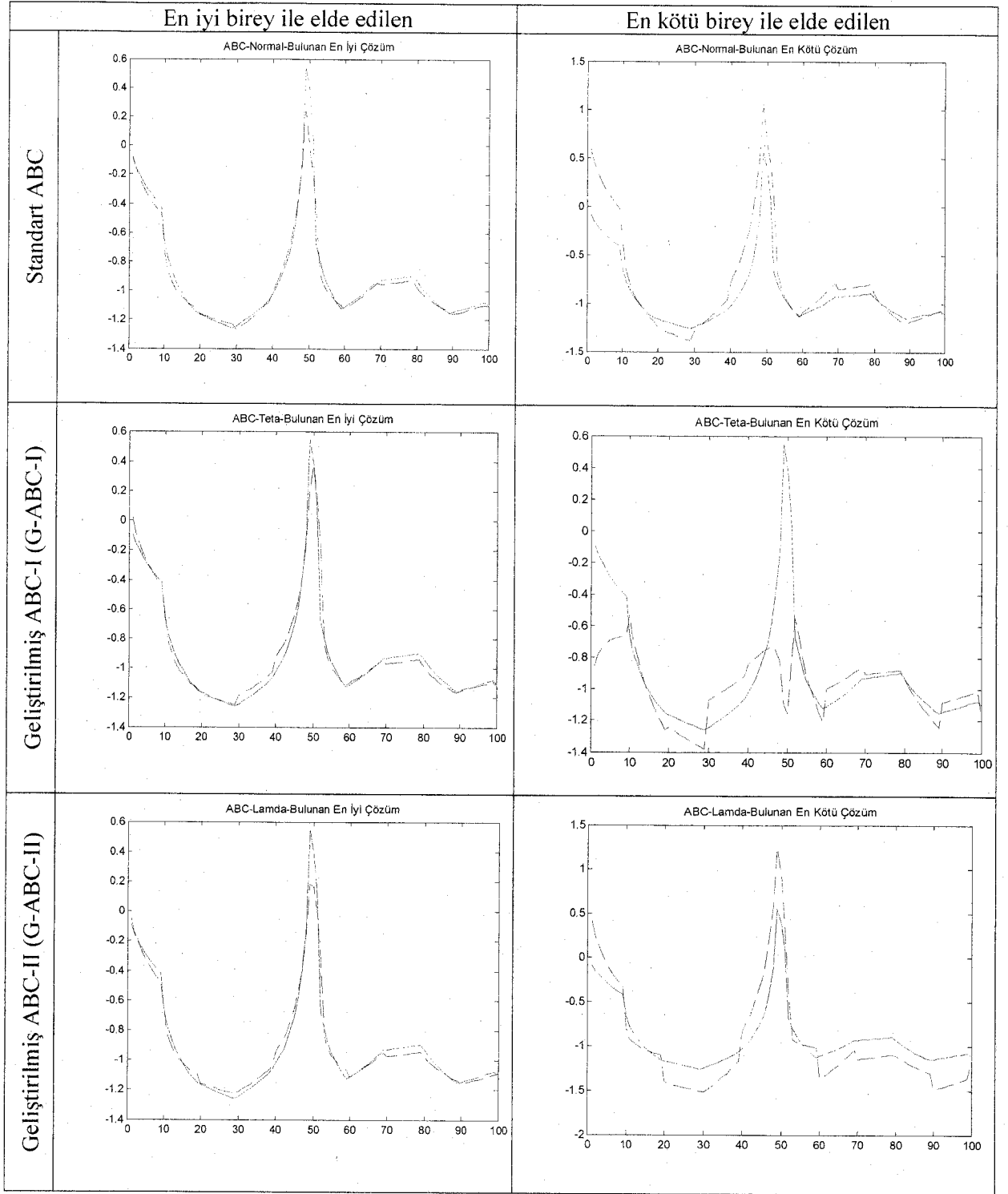
Şekil 9. ABC ile eğitim fazında ÖDS 1 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



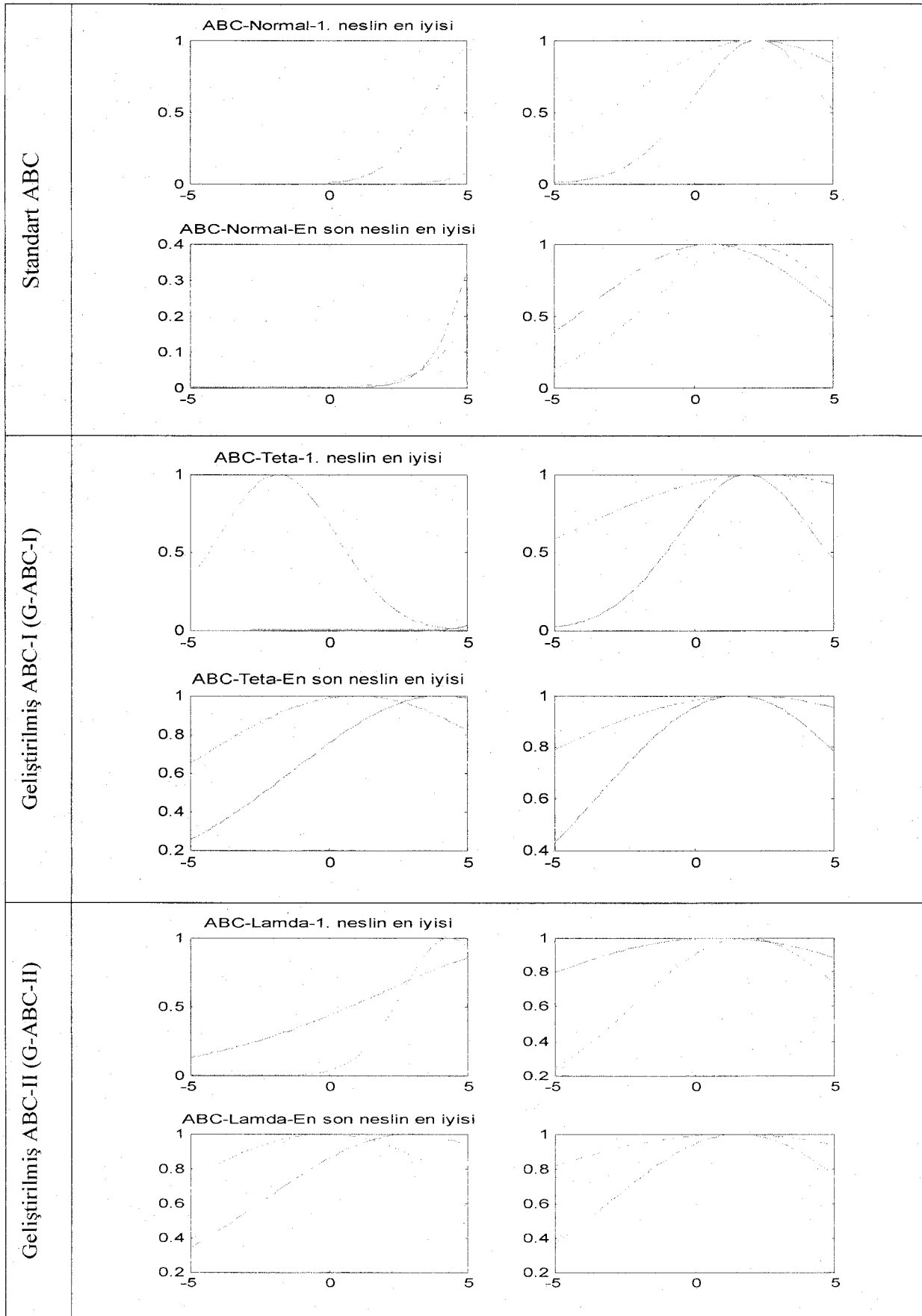
Şekil 10. ABC öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları



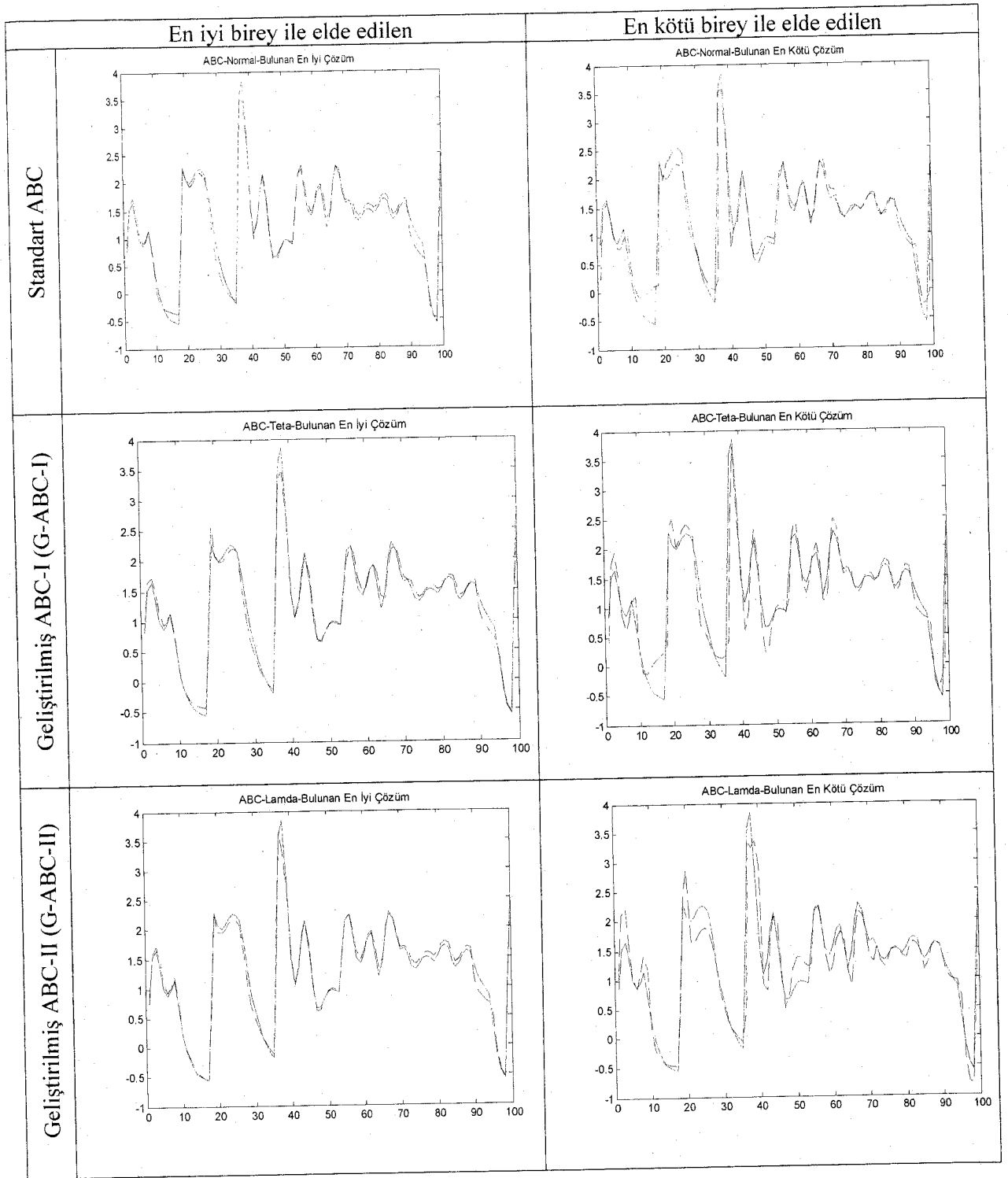
Şekil 11. ABC ile eğitim fazında ÖDS 2 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



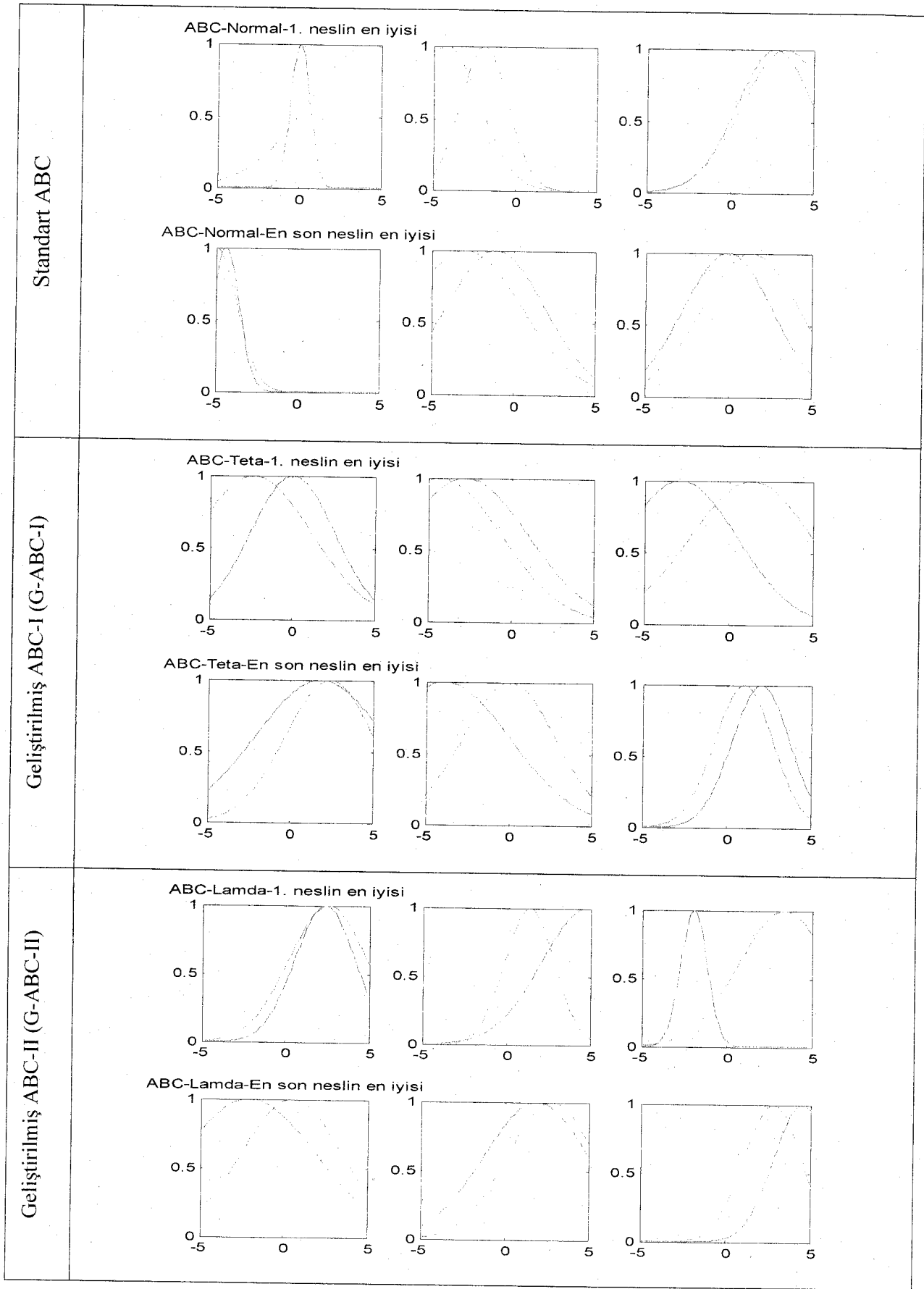
Şekil 12. ABC öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları



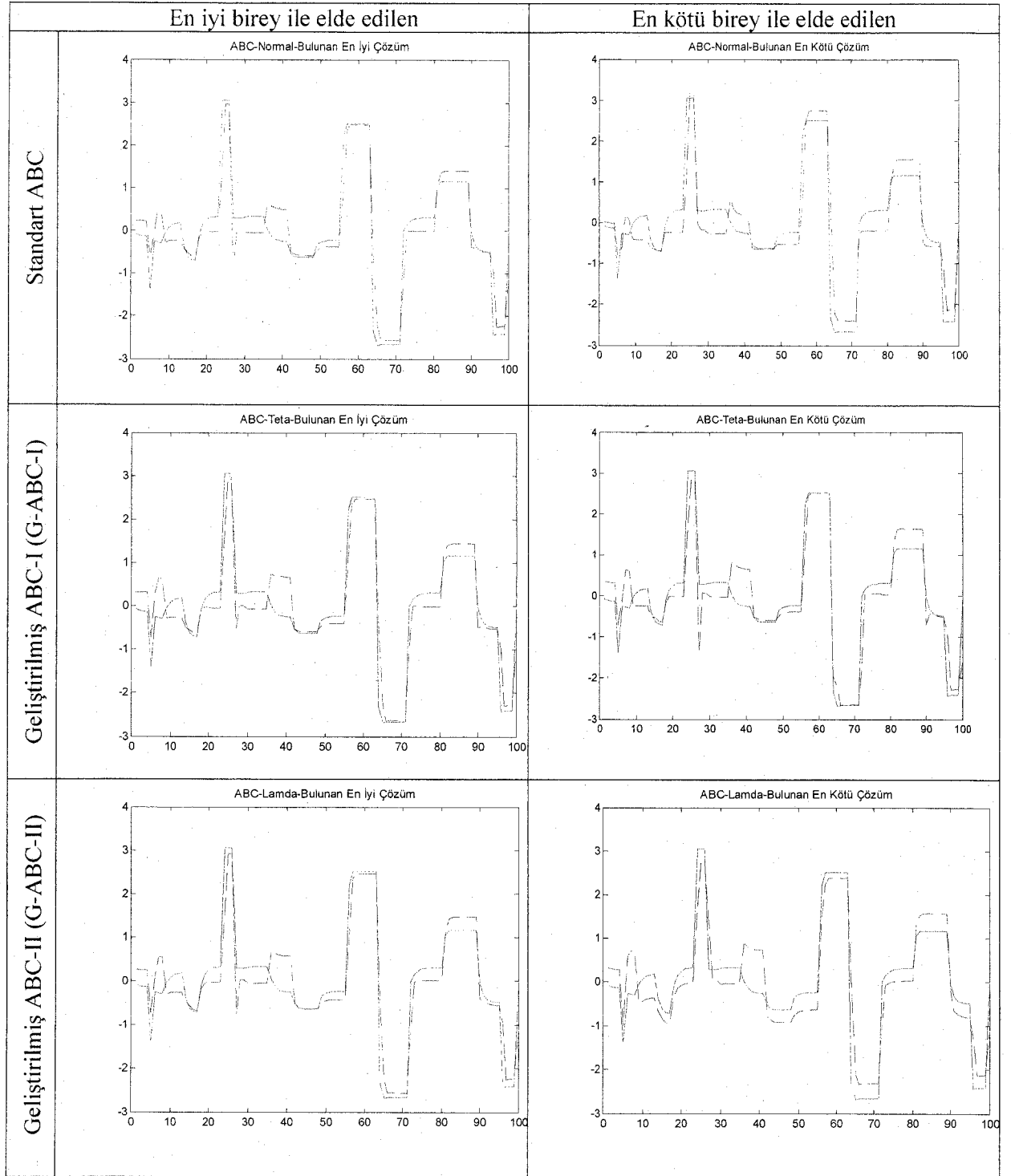
Şekil 13. ABC ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



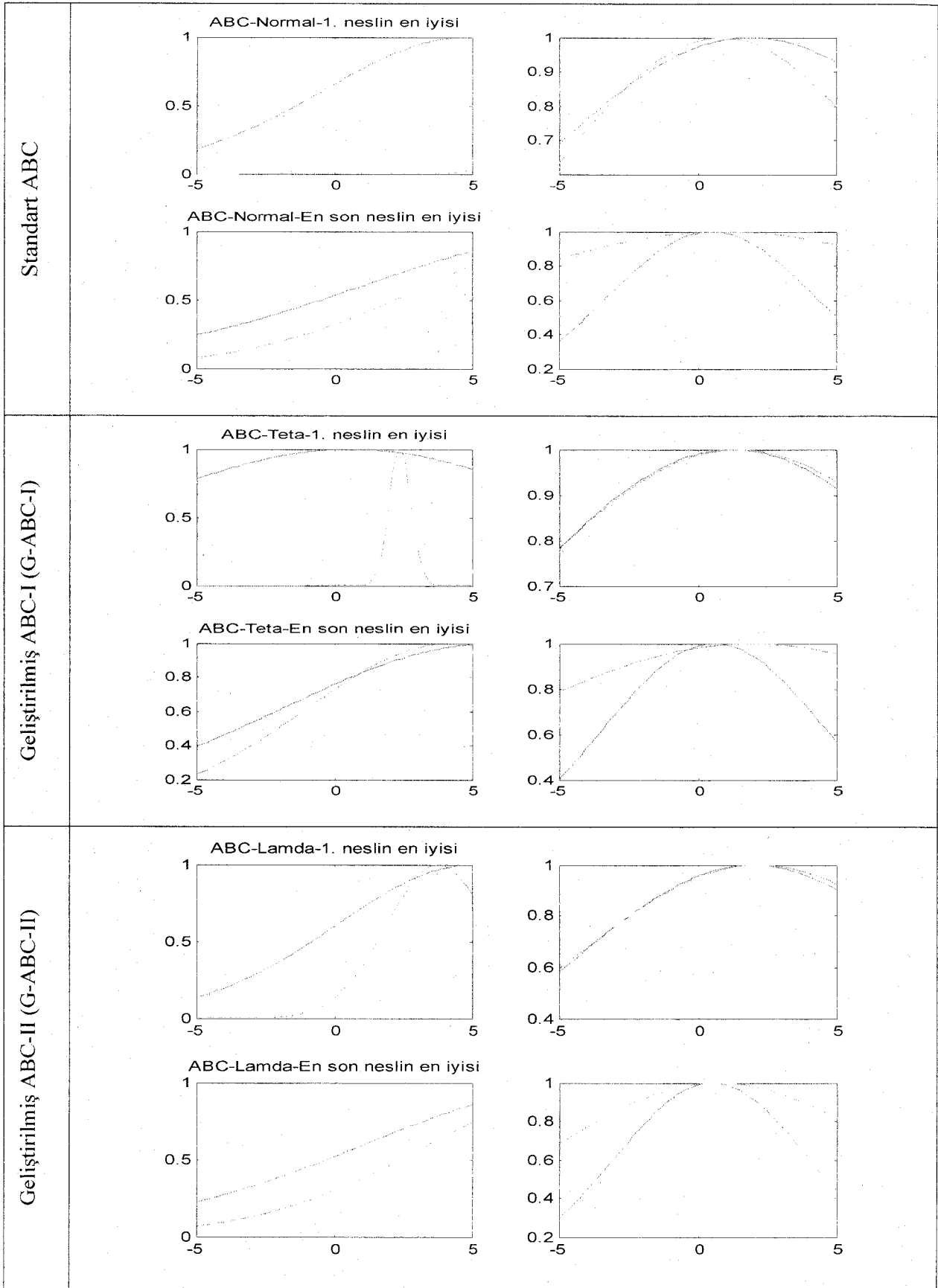
Şekil 14. ABC öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları



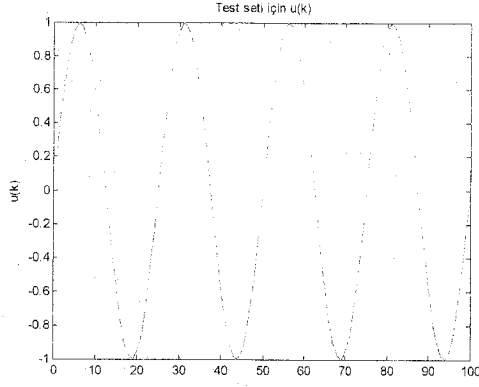
Şekil 15. ABC ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



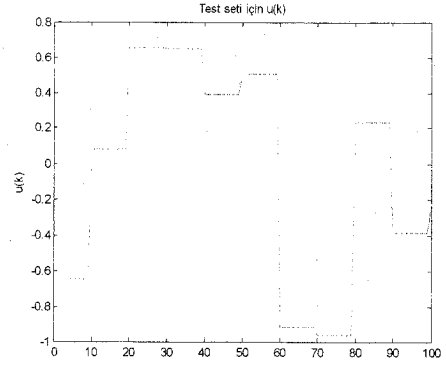
Şekil 16. ABC öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları



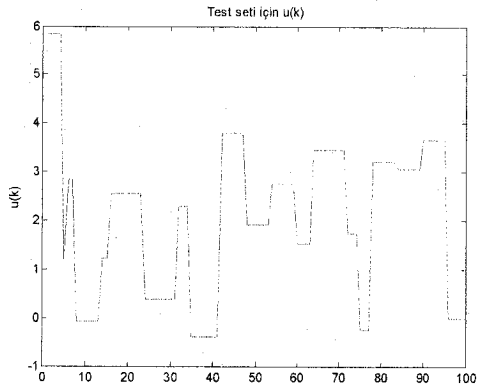
Şekil 17. ABC ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



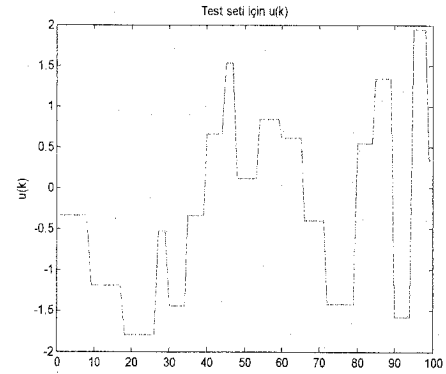
(a)



(b)

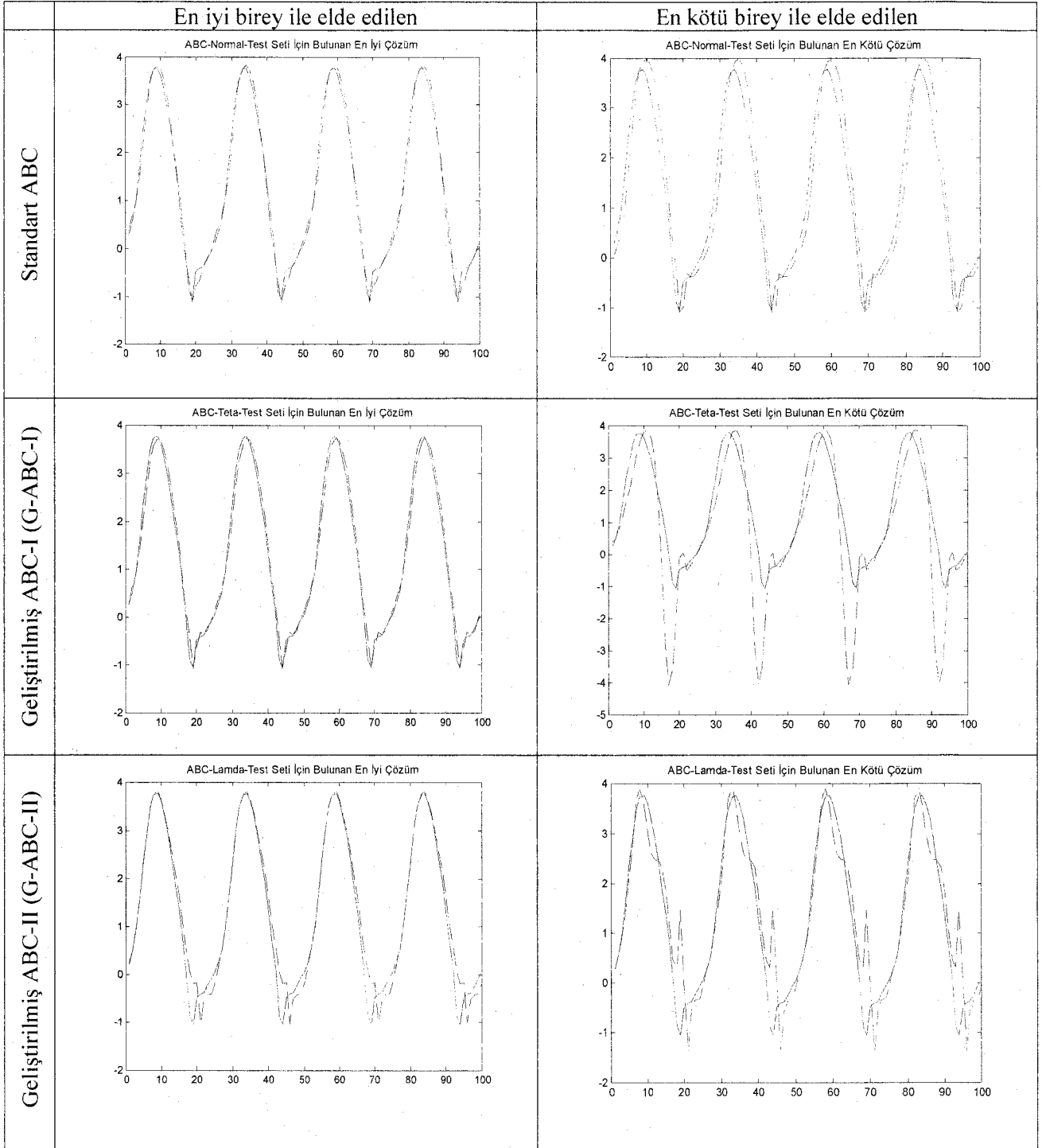


(c)

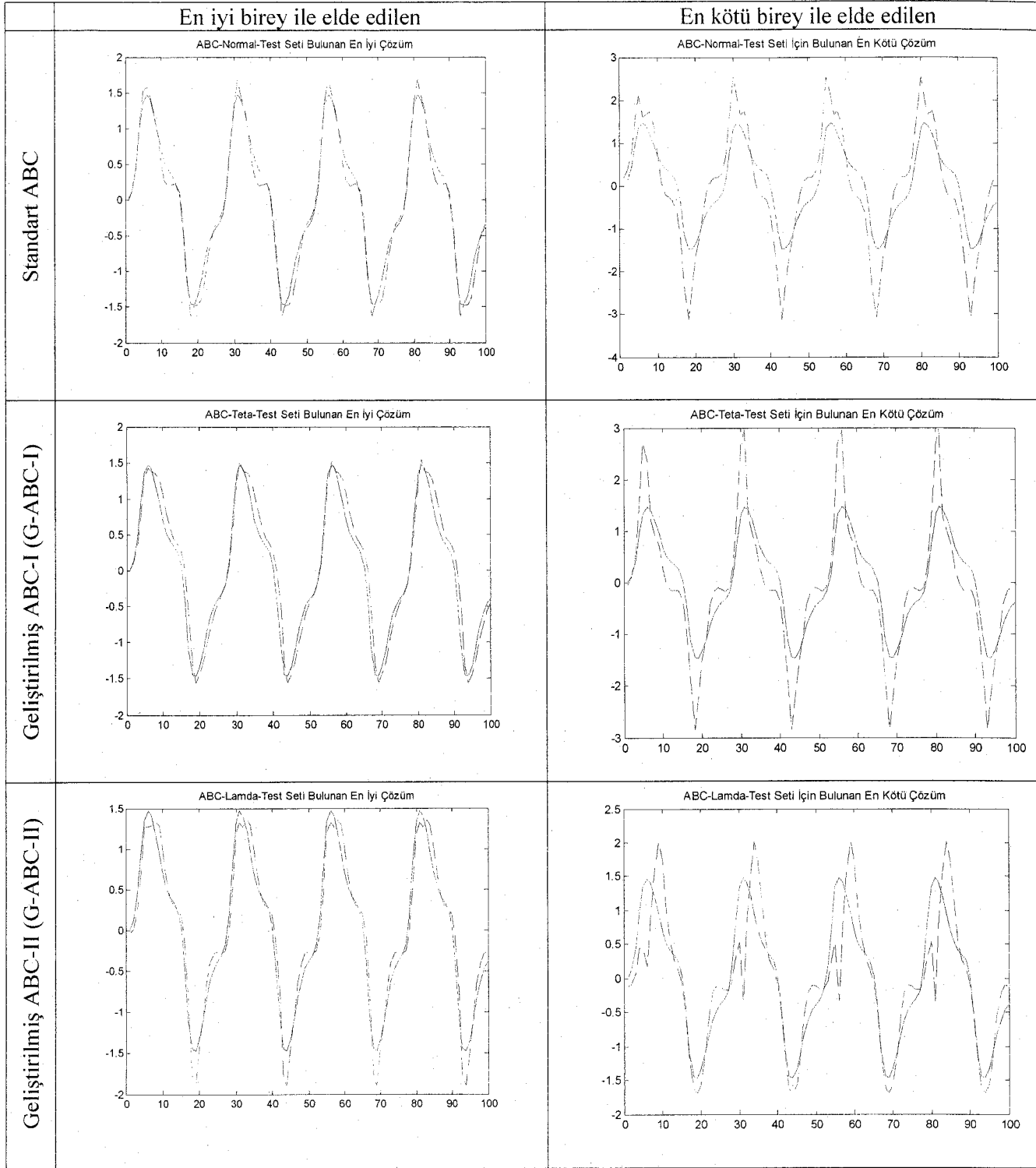


(d)

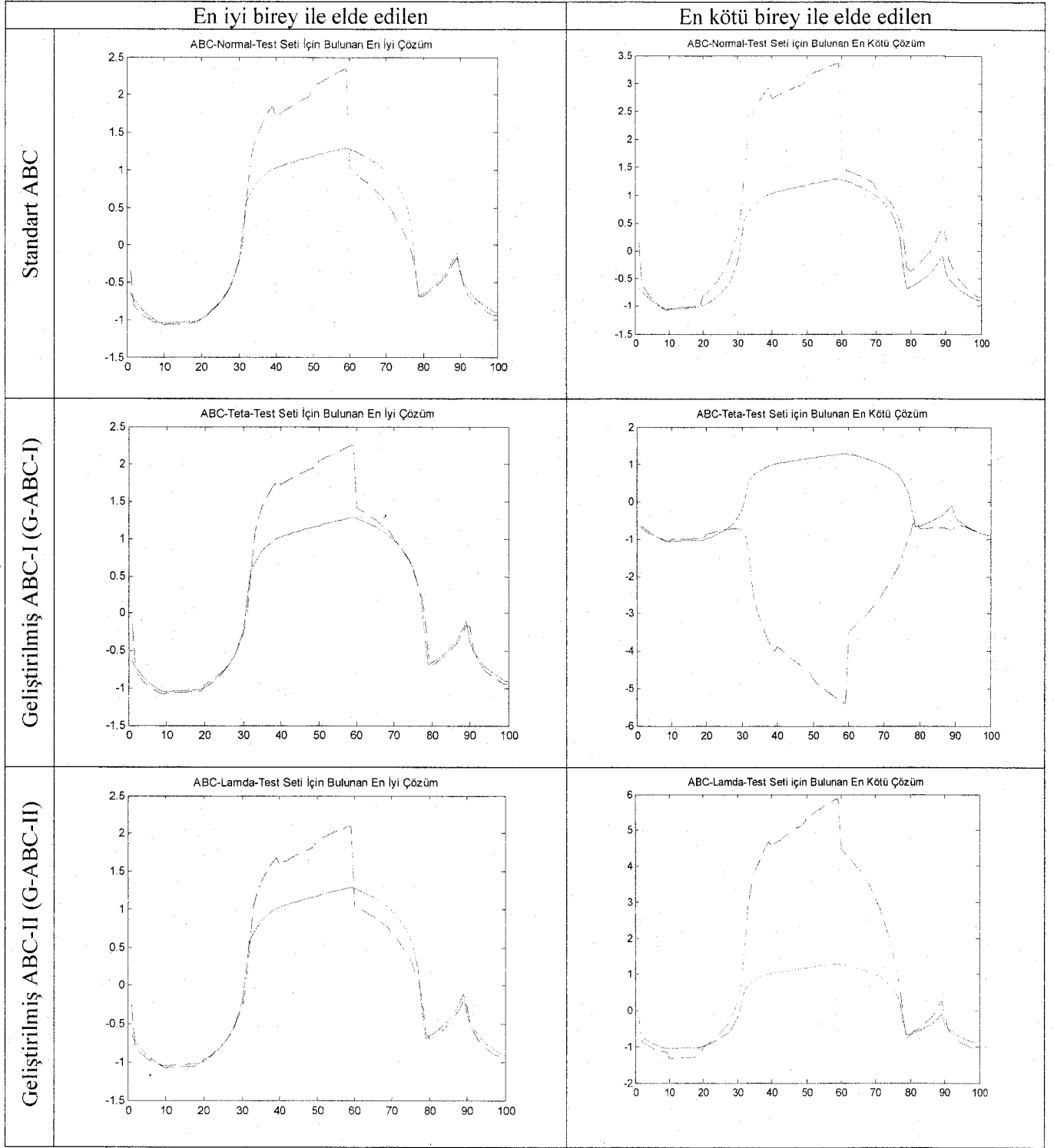
Şekil 18. ÖDS sistemleri için sistem tanımının test fazında kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için $[-1 \ 1]$ aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için $[-5 \ 5]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için $[-2 \ 2]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe



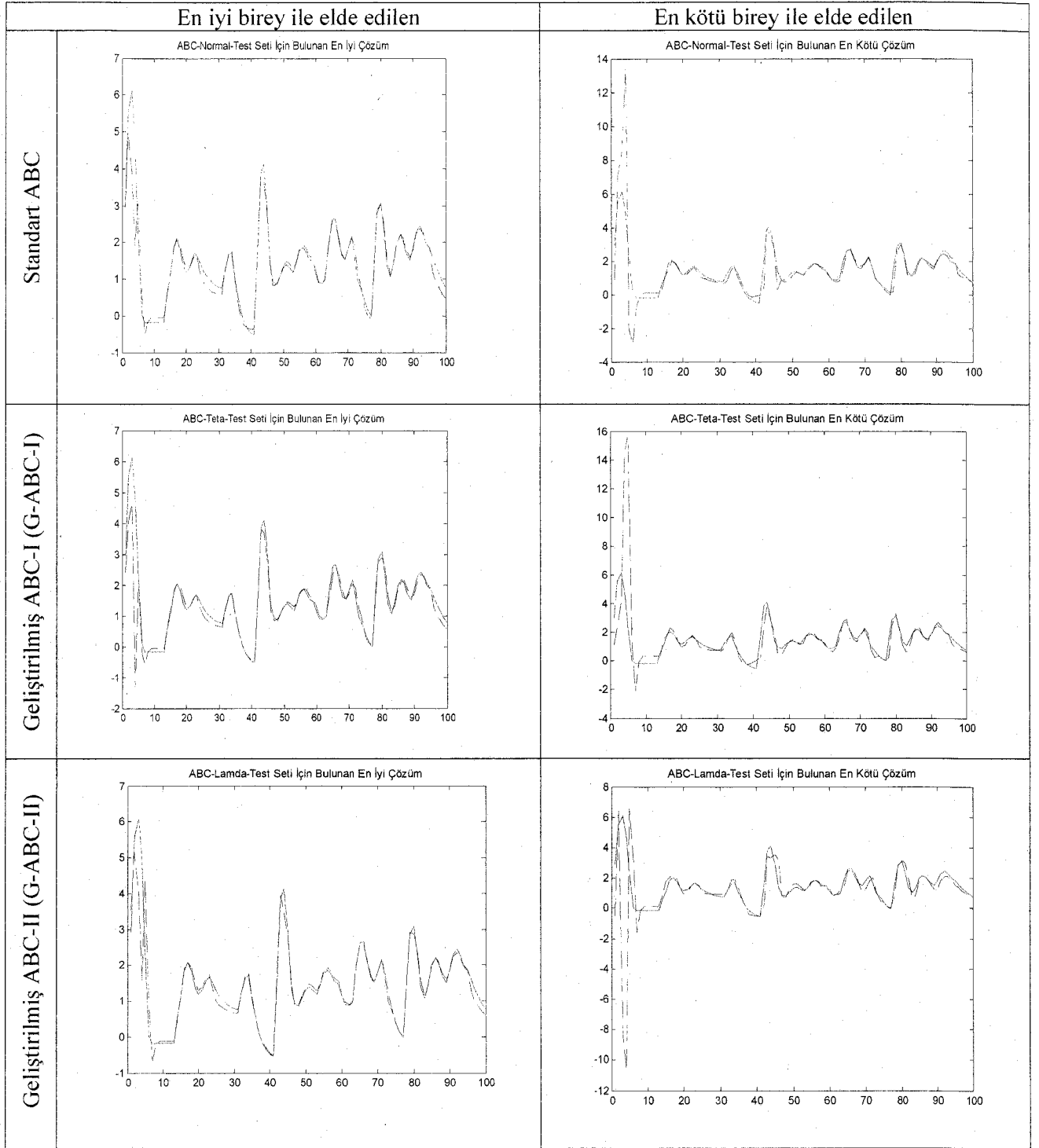
Şekil 19. ABC ile ÖDS 1 için elde edilen ANFIS modelinin test sonuçları



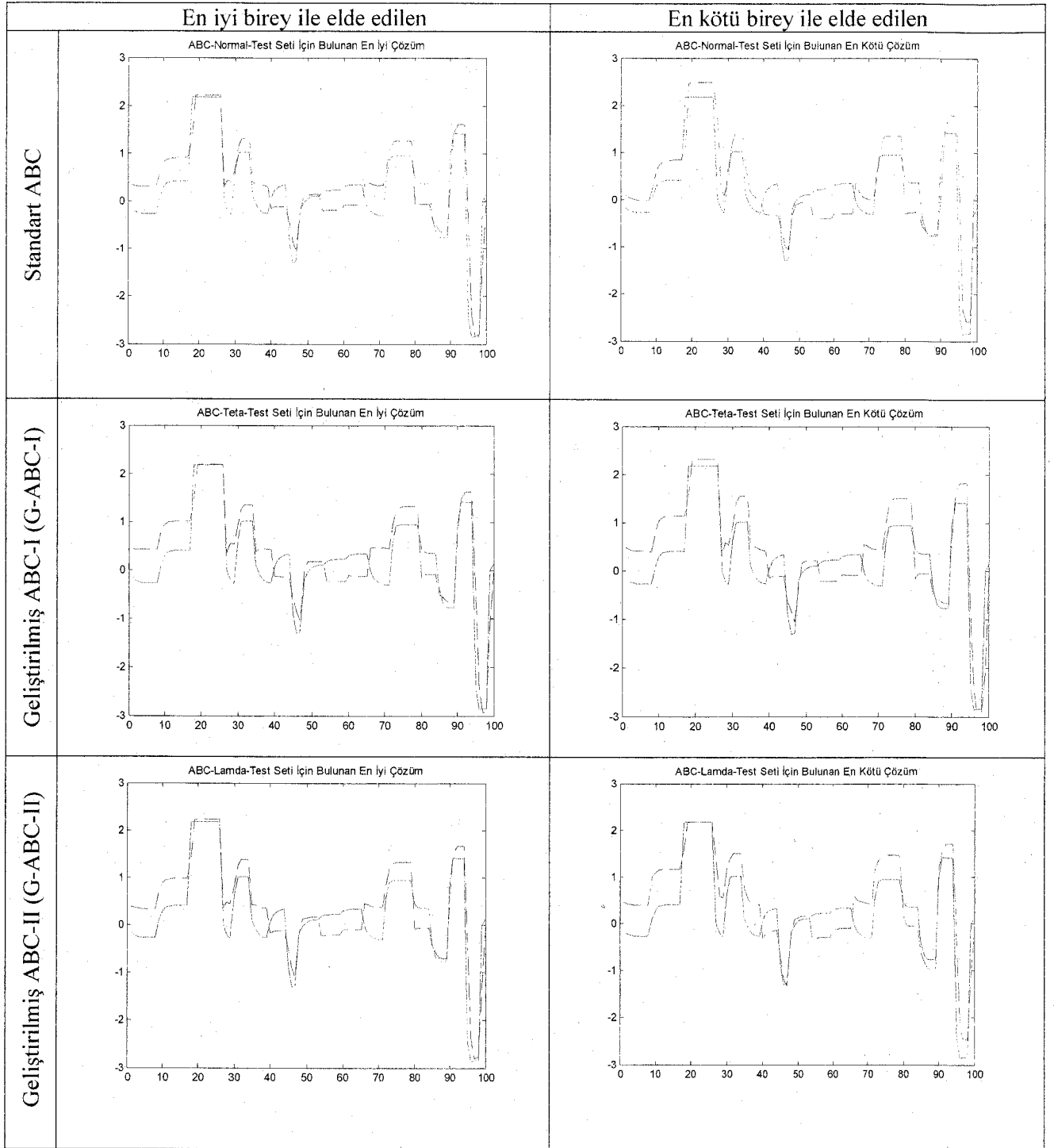
Şekil 20. ABC ile ÖDS 2 için elde edilen ANFIS modelinin test sonuçları



Şekil 21. ABC ile ÖDS 3 için elde edilen ANFIS modelinin test sonuçları



Şekil 22. ABC ile ÖDS 4 için elde edilen ANFIS modelinin test sonuçları



Şekil 23. ABC ile ÖDS 5 için elde edilen ANFIS modelinin test sonuçları

4. III. ARA DÖNEMDE YAPILAN ÇALIŞMALAR

Bir önceki bölümde açıklanan çalışmalar aynı tarzda PSO algoritması ile tekrarlanmıştır. Bu çalışmaların sonuçları izleyen alt bölümde özetlenmiştir. ABC algoritmasının başarımının kıyaslanacağı diğer rakip algoritma olan farksal gelişim (DE) algoritması için de gerekli yazılımlar tamamlanmış ve önceki bölümde verilen aynı tarzda denektaşı problem üzerinde başarımı irdelenmiştir. DE ile elde edilen sonuçlar bu bölümün son kısmında verilecektir.

4.1 Parçacık Sürü Algoritması (PSO) ile yapılan çalışmalar

Parçacık sürü optimizasyonu algoritması 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiş, bireysel ve sosyal etkileşimi kullanarak bireylerini arama uzayında en anlamlı bölgeye yönlendiren sürü tabanlı sezgisel bir arama algoritmasıdır. Algoritma rasgele atanmış bireylerle (parçacık) başlayarak her bir nesilde parçacıkların hızları ve pozisyonları güncellenir.

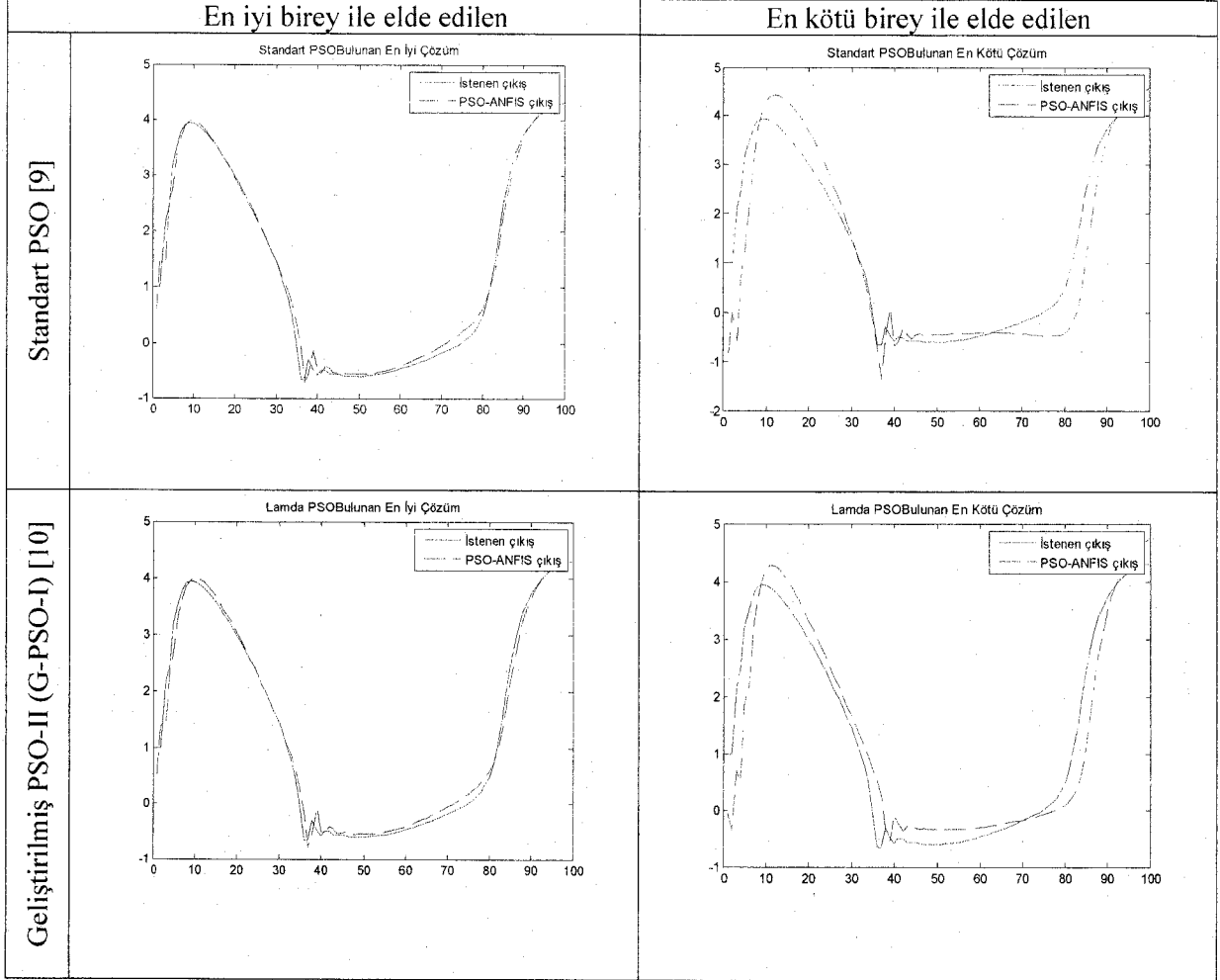
Bu çalışmada parçacık hızlarının güncellenmesi için algoritmanın yalın hallerinden biri olan (5)'de verilen yöntem [9] ve geliştirilmiş şekli olan (6)'de [10] verilen yöntem Tablo 6'da verilen parametrelerle ayrı ayrı kullanılmıştır. Bu iki algoritma 3. Bölümde verilen denektaşı dinamik sistemlerin ANFIS bulanık-nöral ağ mimarisi ile modellenmesi amacıyla aynı tarzda ayrı ayrı 50'şer kez koşturulmuştur. Elde edilen sonuçlar Şekil 24-38'de grafiksel olarak özetlenmiştir.

$$v_i(n) = \xi(v_i(n-1) + c_1 r_1 (p_{i,lbest} - p_i(n)) + c_2 r_2 (p_{gbest} - p_i(n))) \quad (5)$$

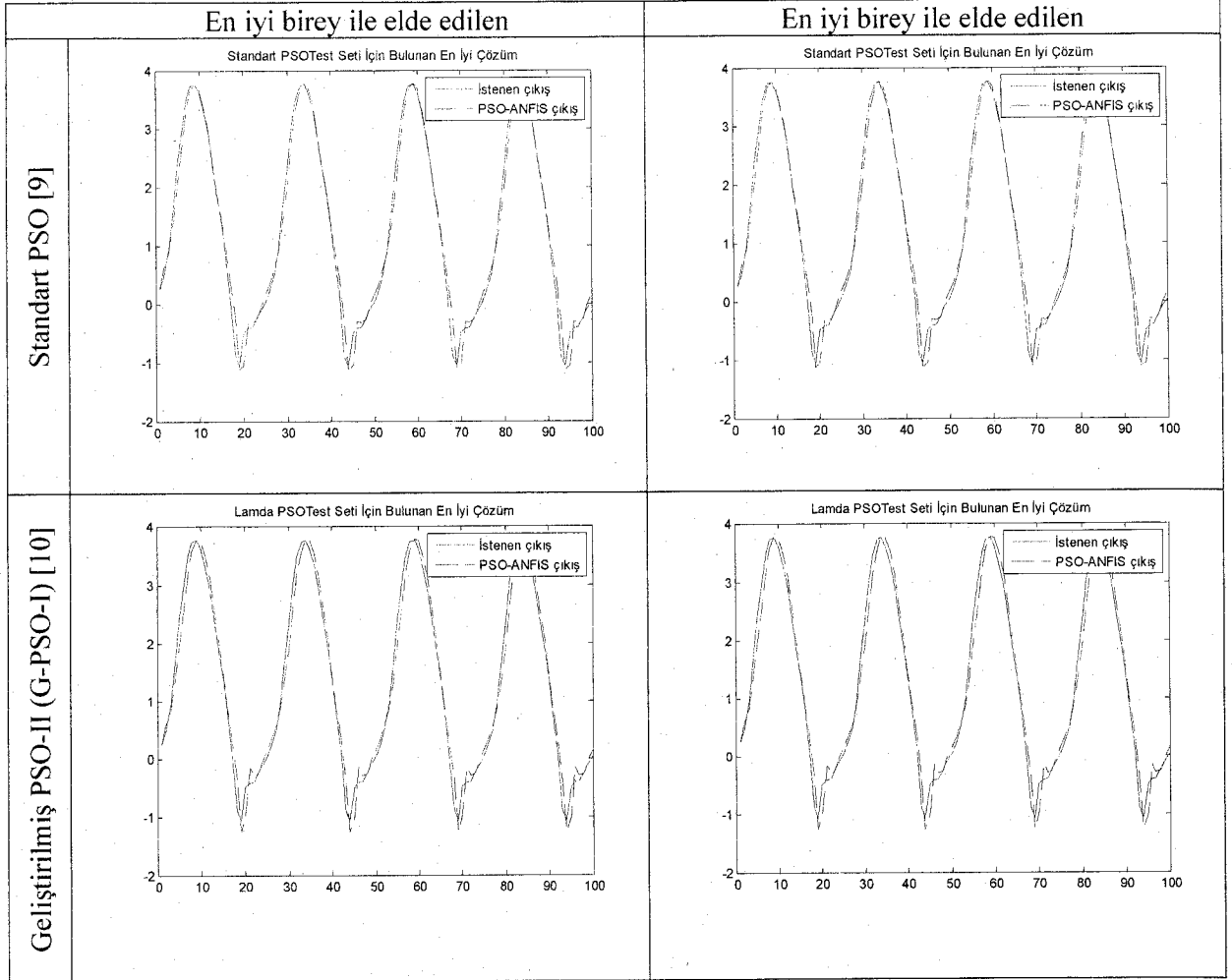
$$v_i(n) = \xi(v_i(n-1) + c_1 r_1 (p_{i,lbest} - p_i(n)) + c_2 r_2 (p_{gbest} - p_i(n))) + \alpha \lambda(n) \quad (6)$$

Tablo 6. PSO parametreleri

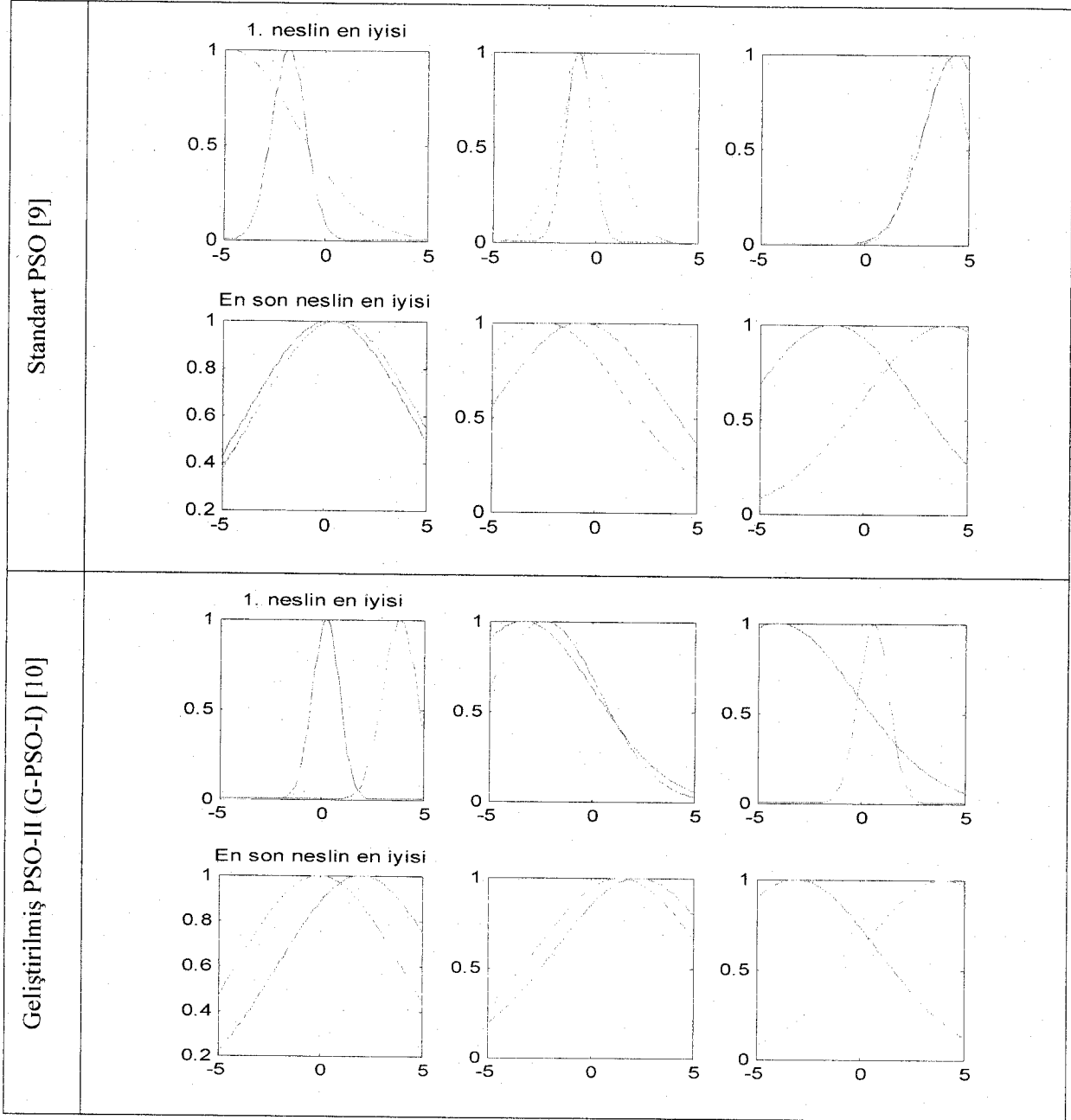
<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
<i>Koloni büyüklüğü (N)</i>	40
<i>Maksimum nesil sayısı</i>	1000
<i>Öğrenme faktörleri</i>	$c_1=c_2=2.05, \alpha = \frac{1}{100 \max(\text{eig}(XX^T))}$
<i>Kısıtlama faktörü</i>	$\xi = 0.723$



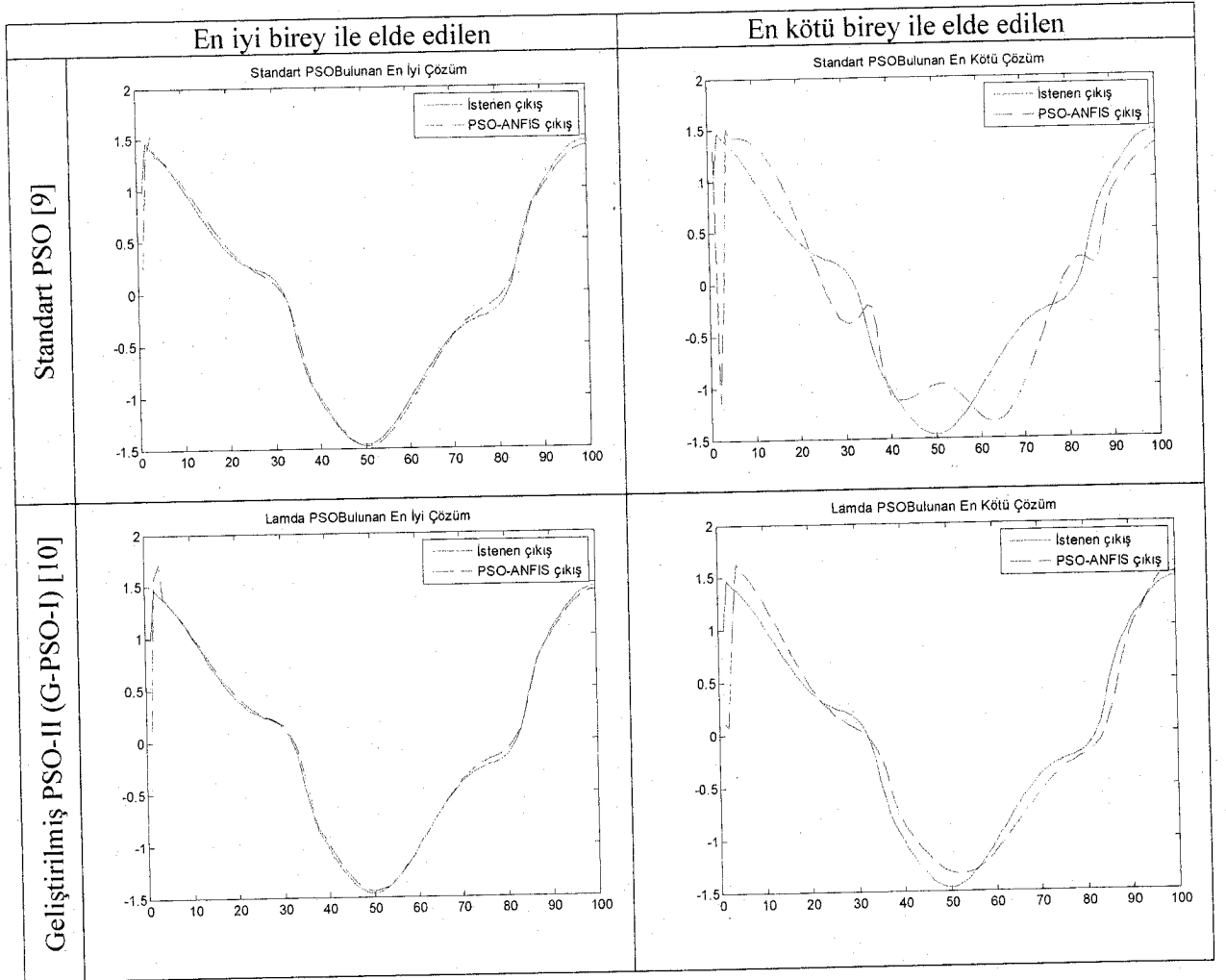
Şekil 24. PSO öğrenmeli ANFIS ile ÖDS 1 için *EĞİTİM* fazı sistem tanıma sonuçları



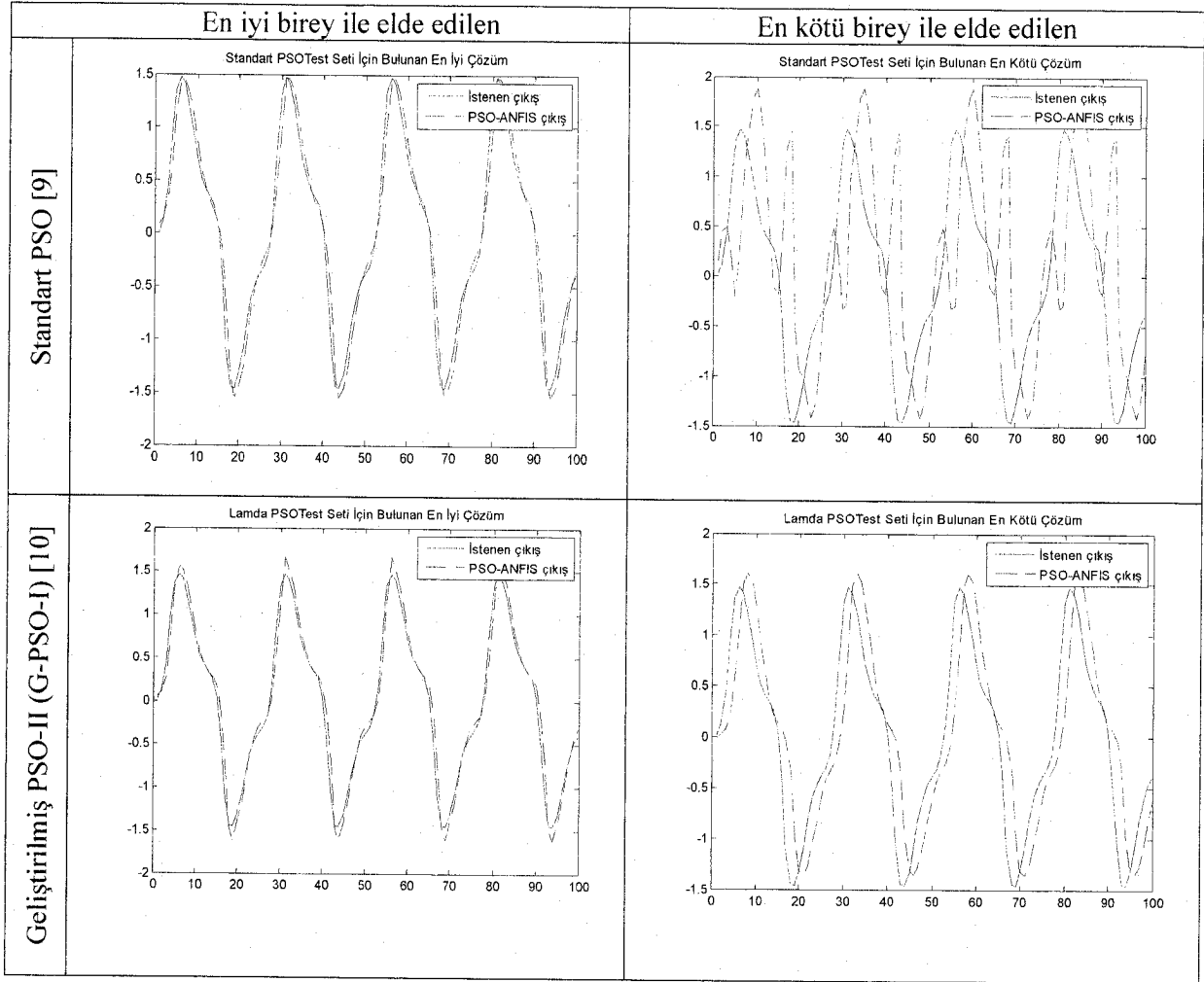
Şekil 25. PSO öğrenmeli ANFIS ile ÖDS 1 için *TEST* fazı sistem tanıma sonuçları



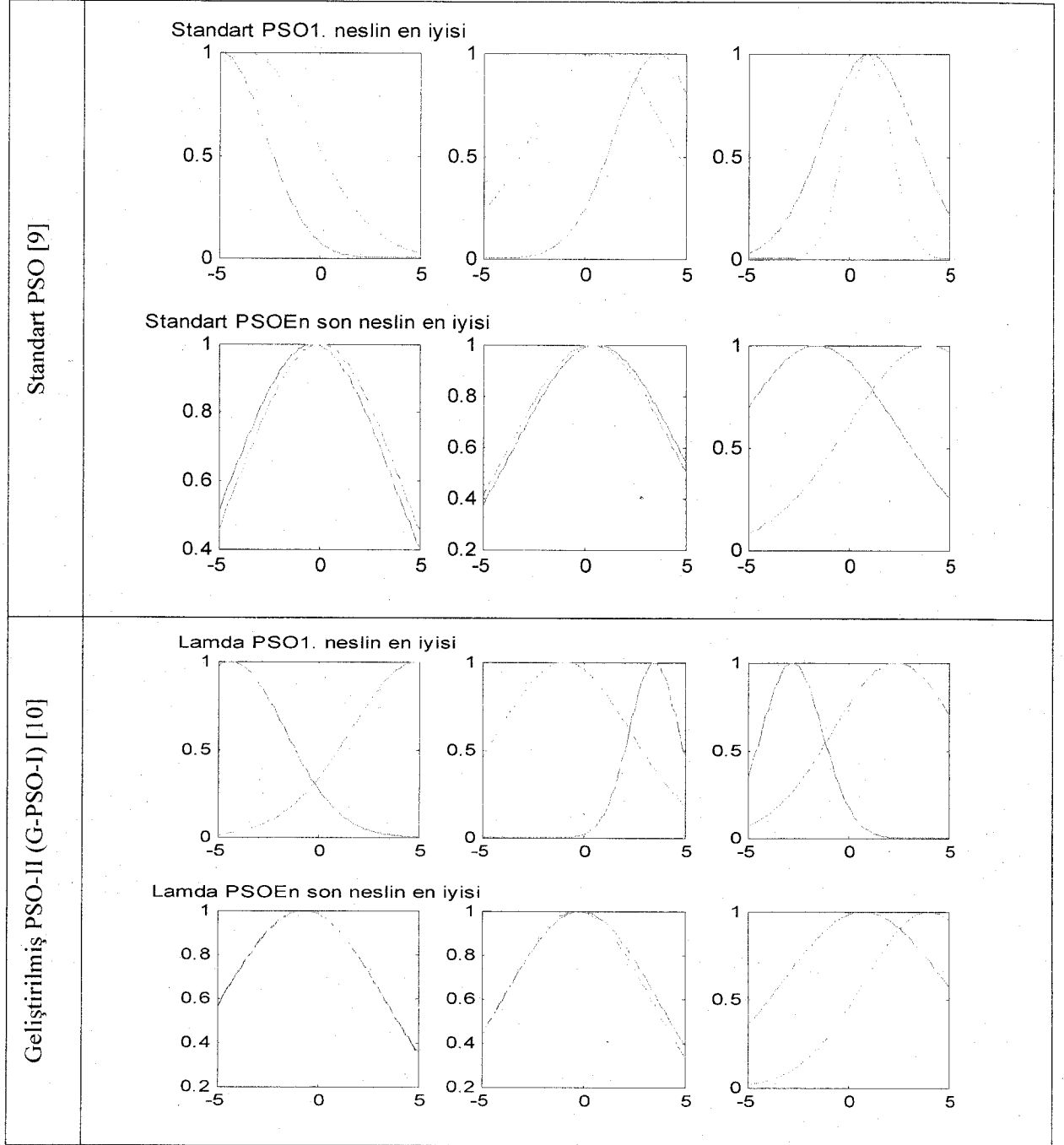
Şekil 26. PSO ile eğitim fazında ÖDS 1 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



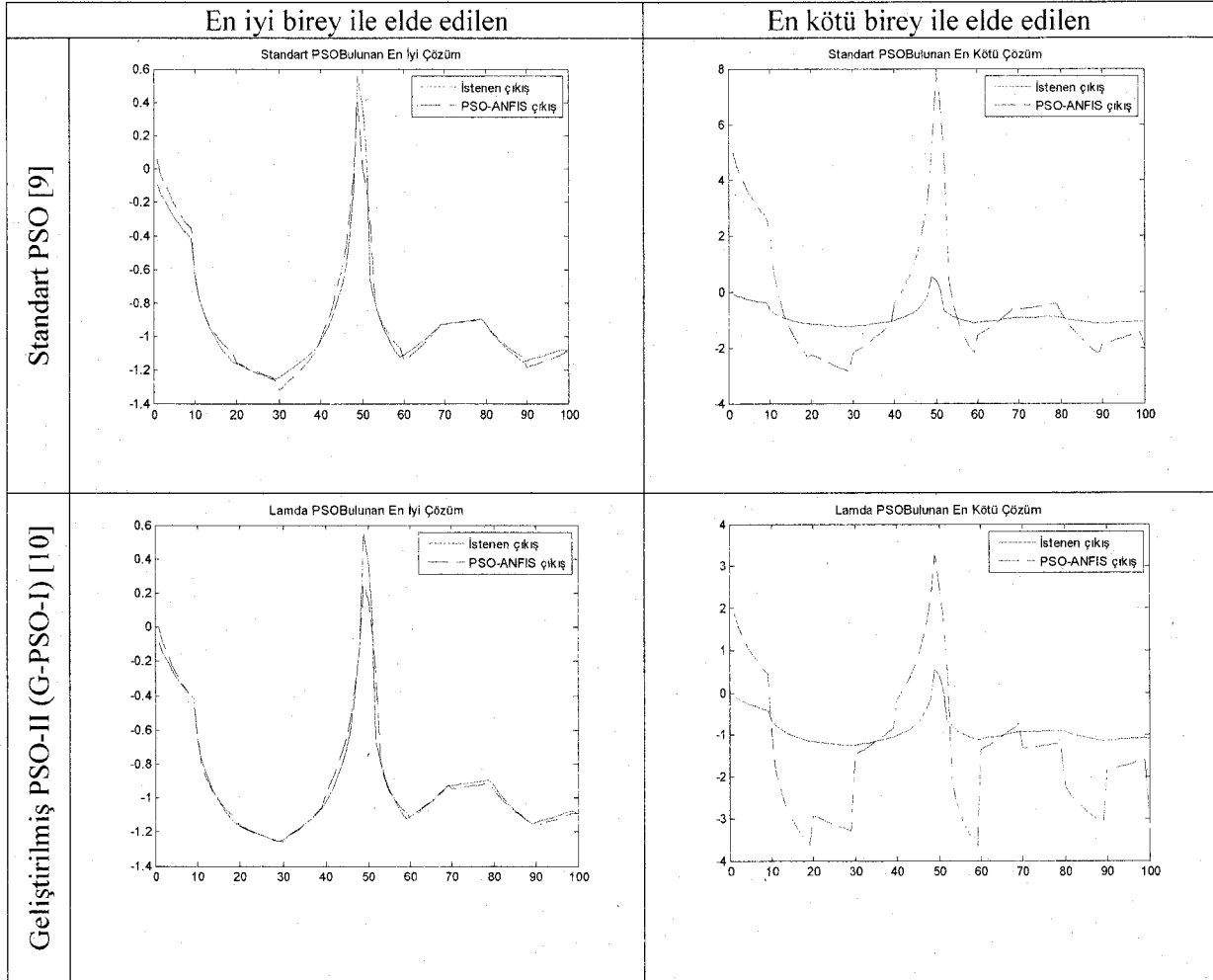
Şekil 27. PSO öğrenmeli ANFIS ile ÖDS 2 için *EĞİTİM* fazı sistem tanıma sonuçları



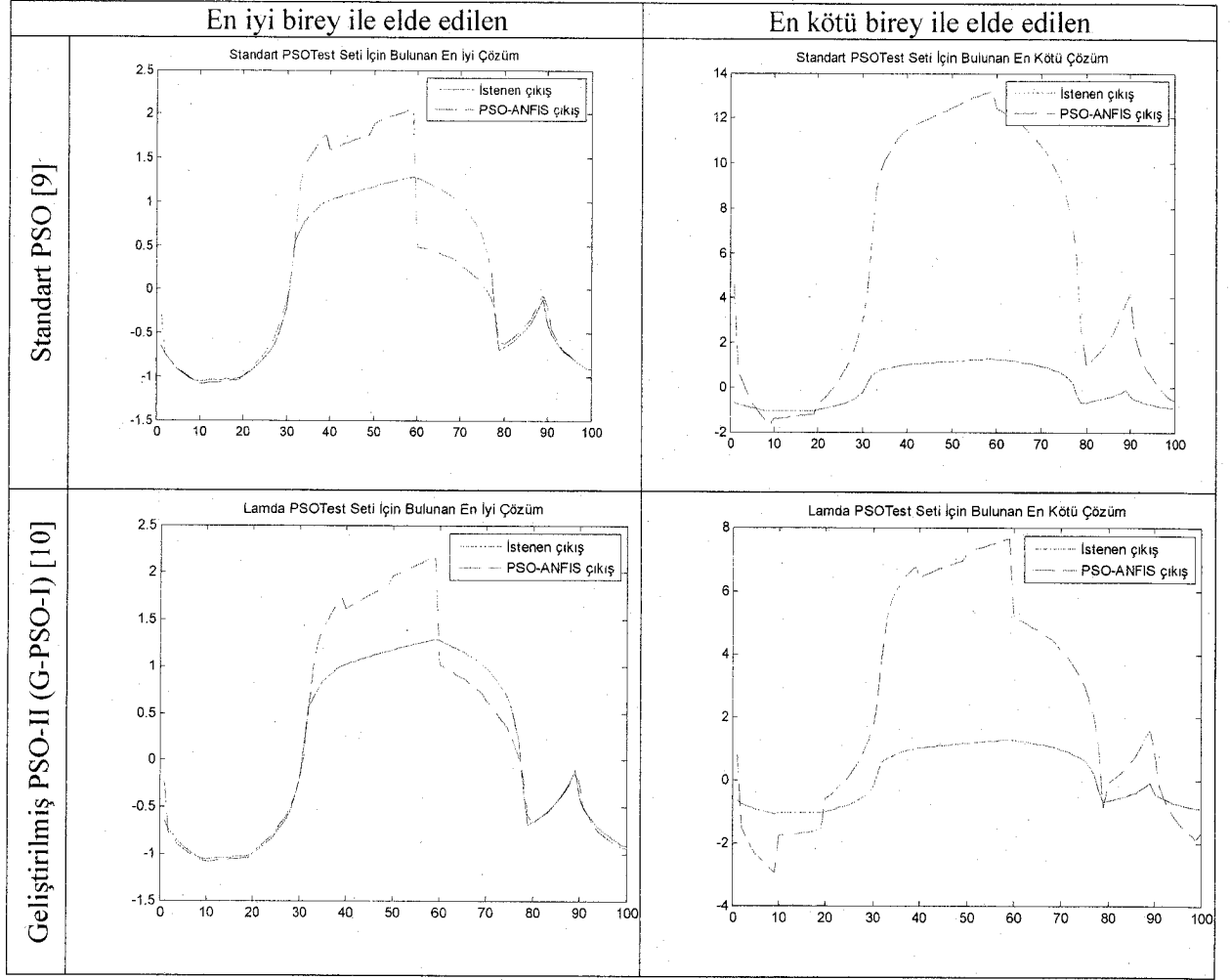
Şekil 28. PSO öğrenmeli ANFIS ile ÖDS 2 için *TEST* fazı sistem tanıma sonuçları



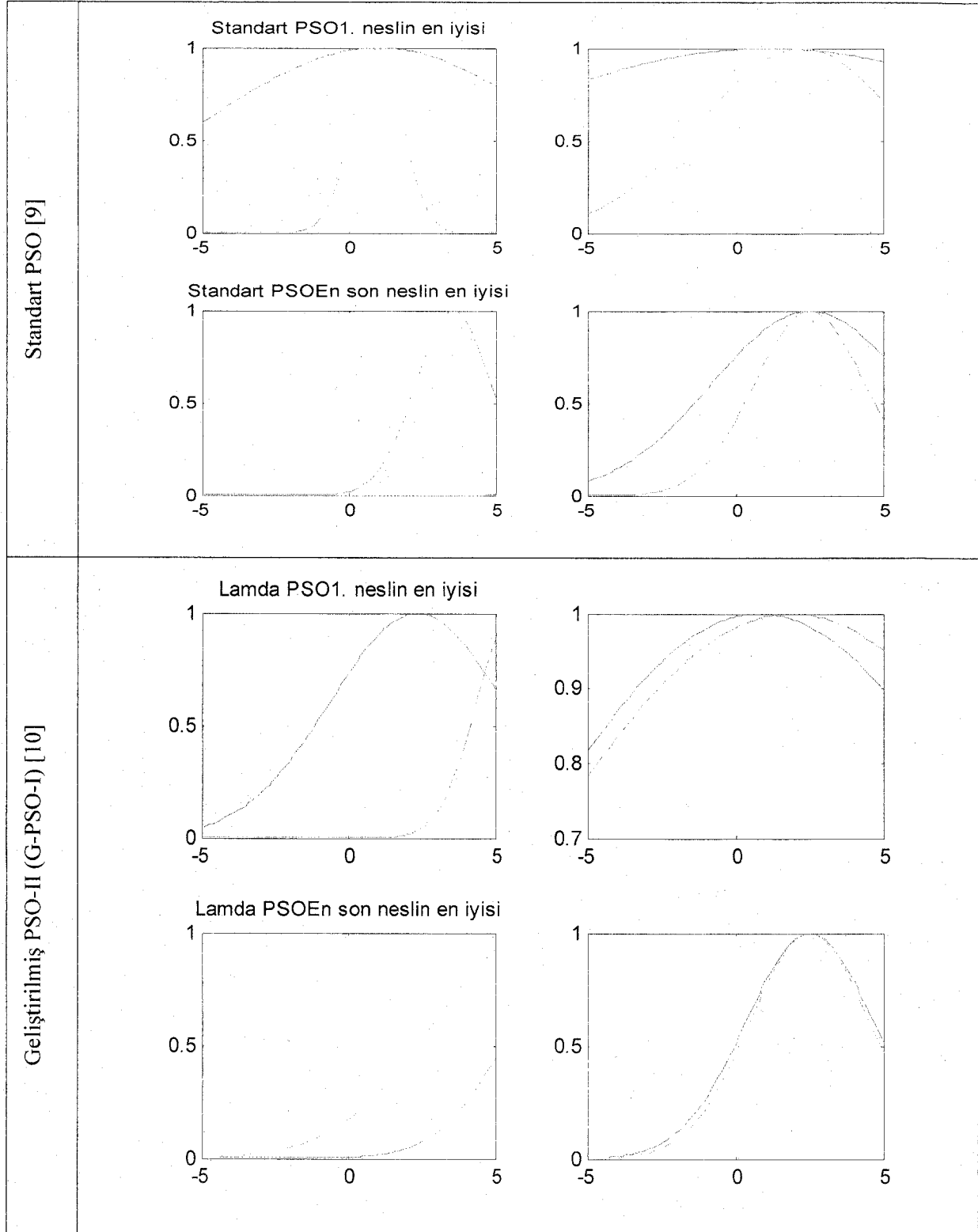
Şekil 29. PSO ile eğitim fazında ÖDS 2 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



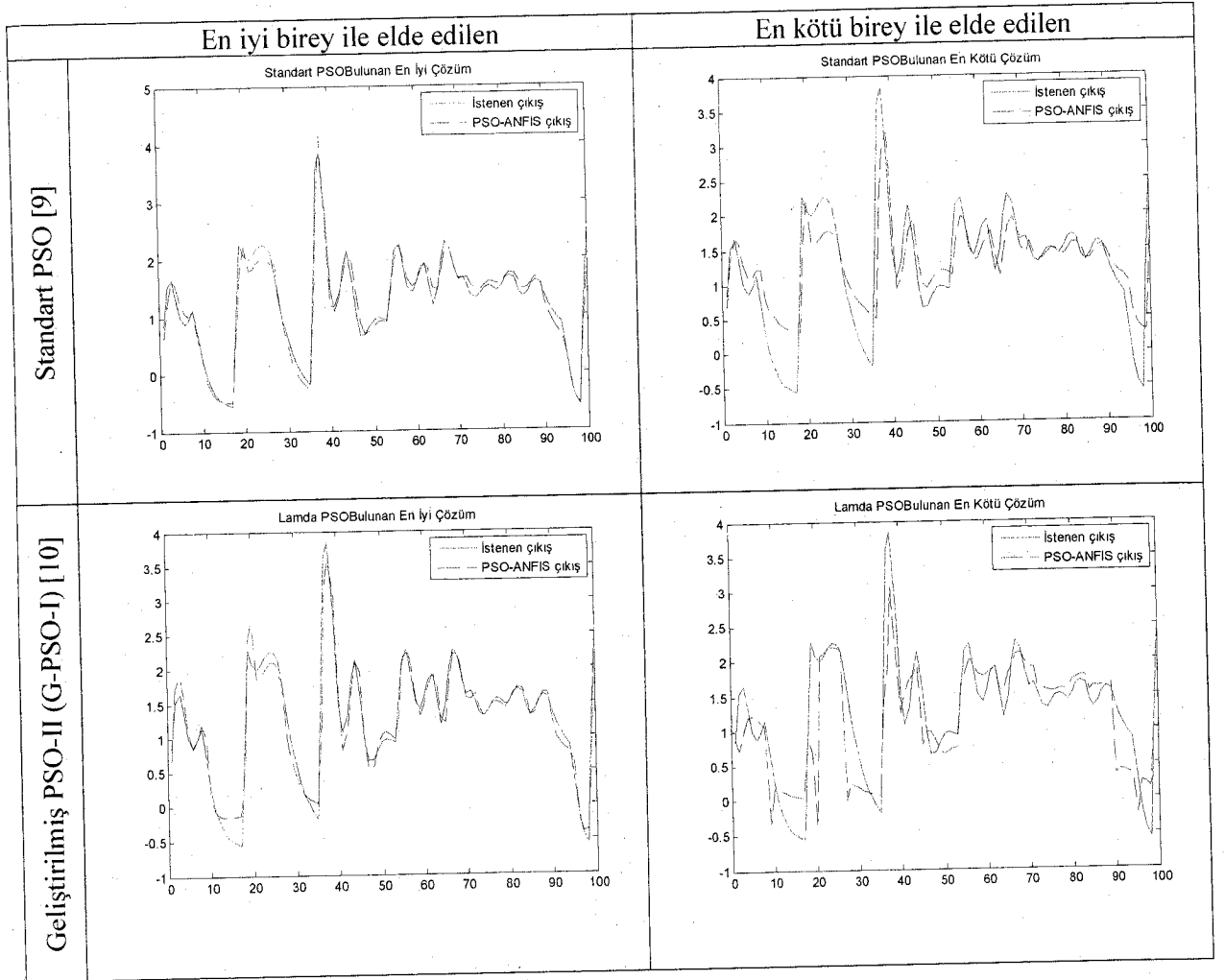
Şekil 30. PSO öğrenmeli ANFIS ile ÖDS 3 için *EĞİTİM* fazı sistem tanıma sonuçları



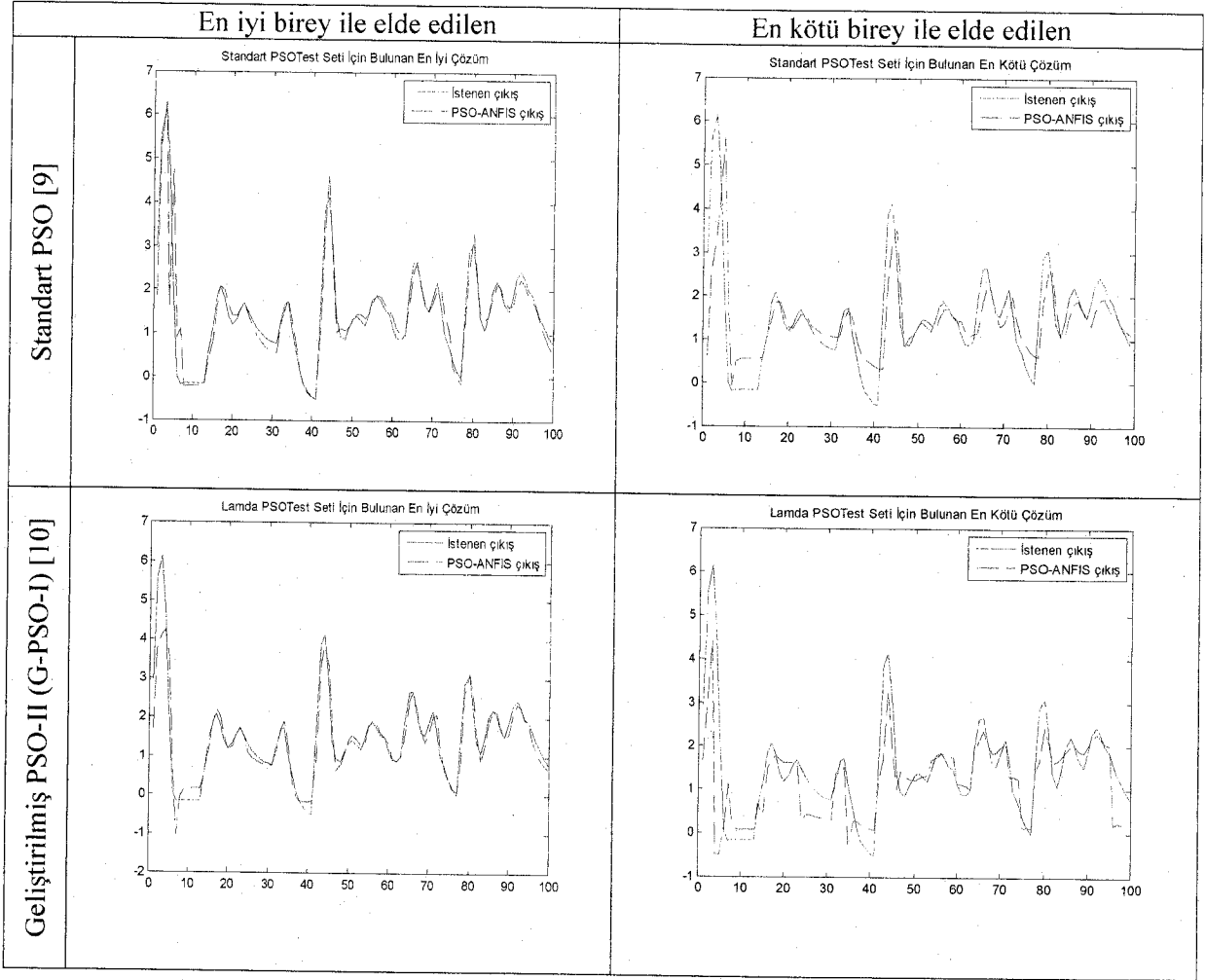
Şekil 31. PSO öğrenmeli ANFIS ile ÖDS 3 için *TEST* fazı sistem tanıma sonuçları



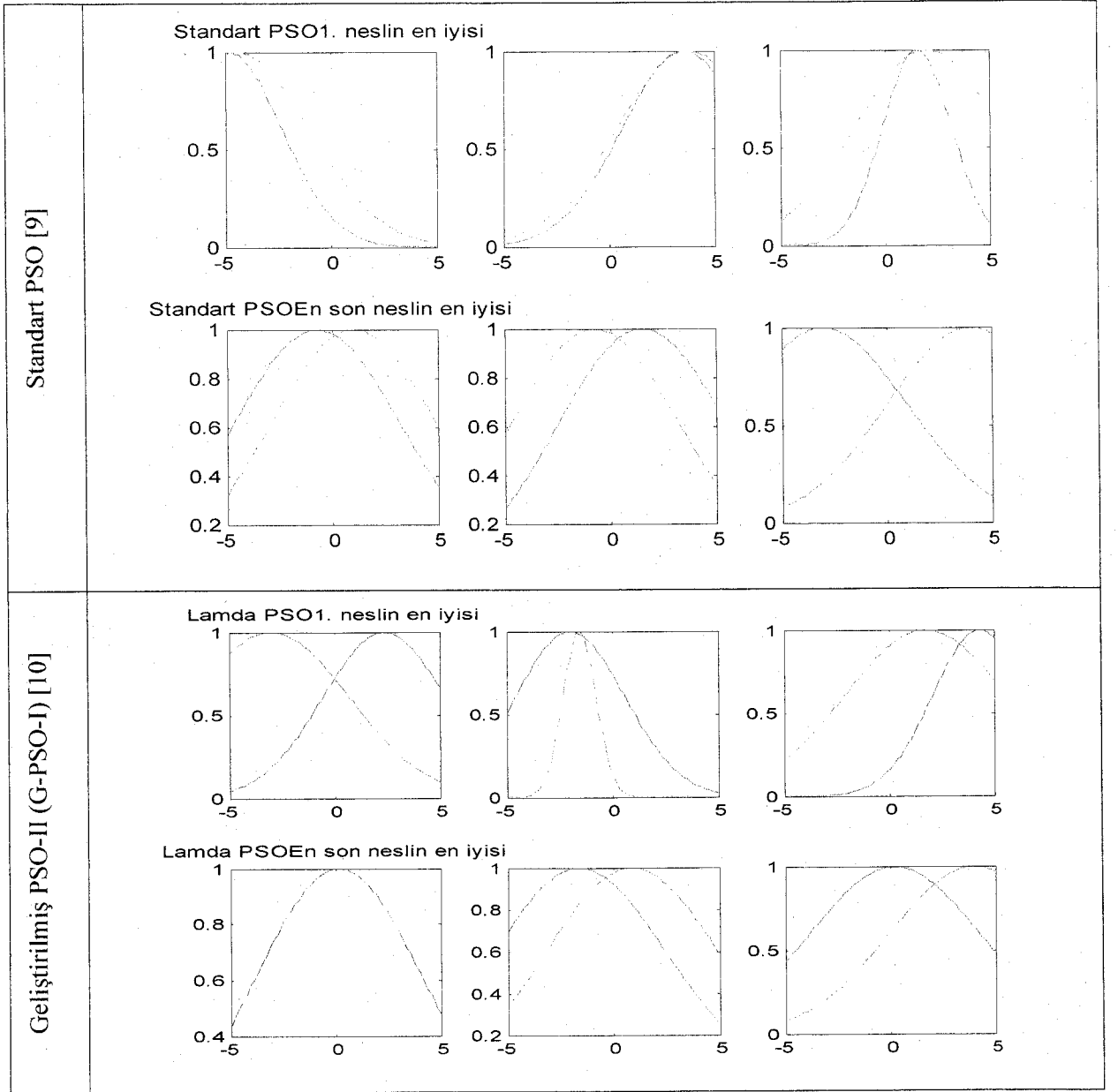
Şekil 32. PSO ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



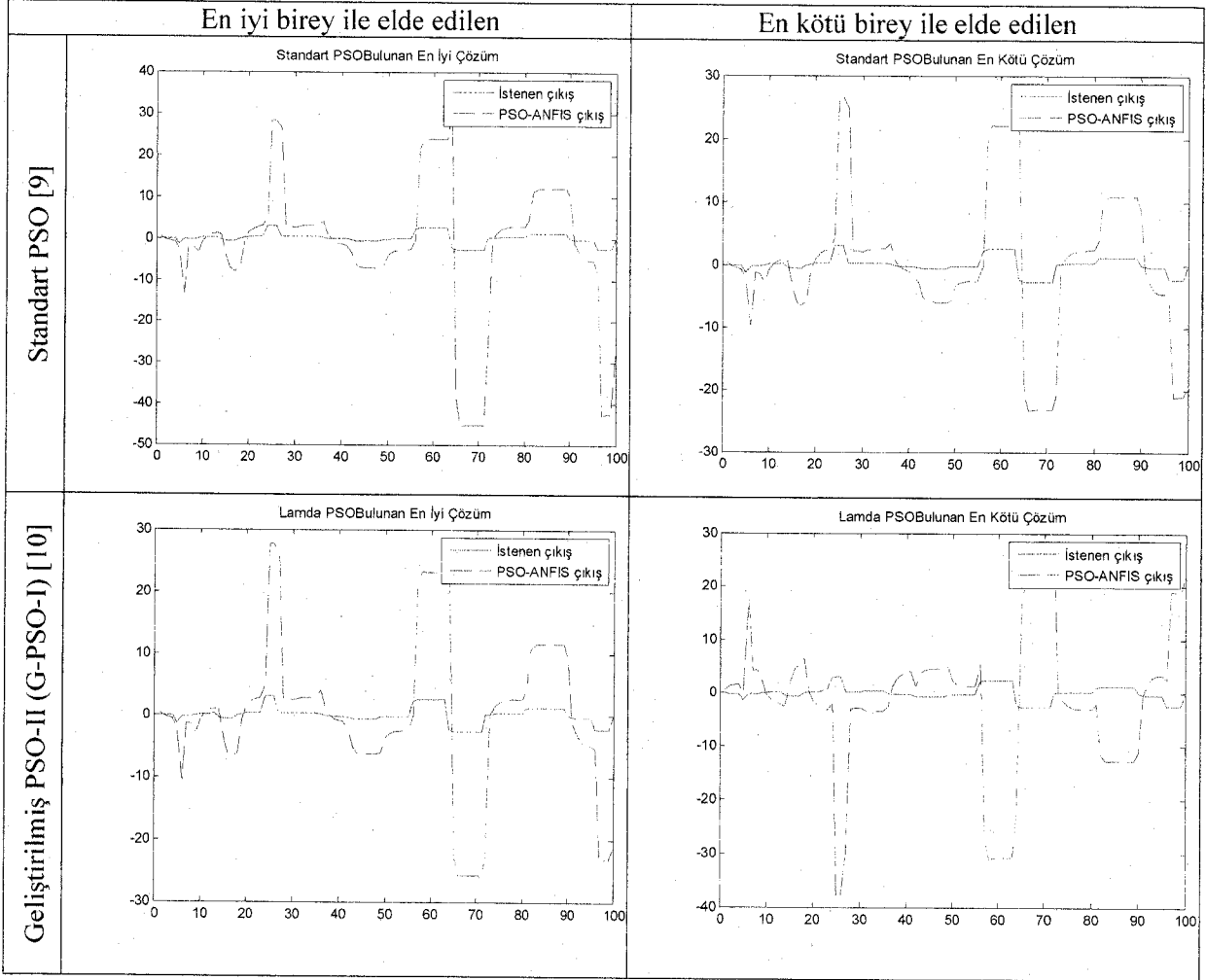
Şekil 33. PSO öğrenmeli ANFIS ile ÖDS 4 için EĞİTİM fazı sistem tanıma sonuçları



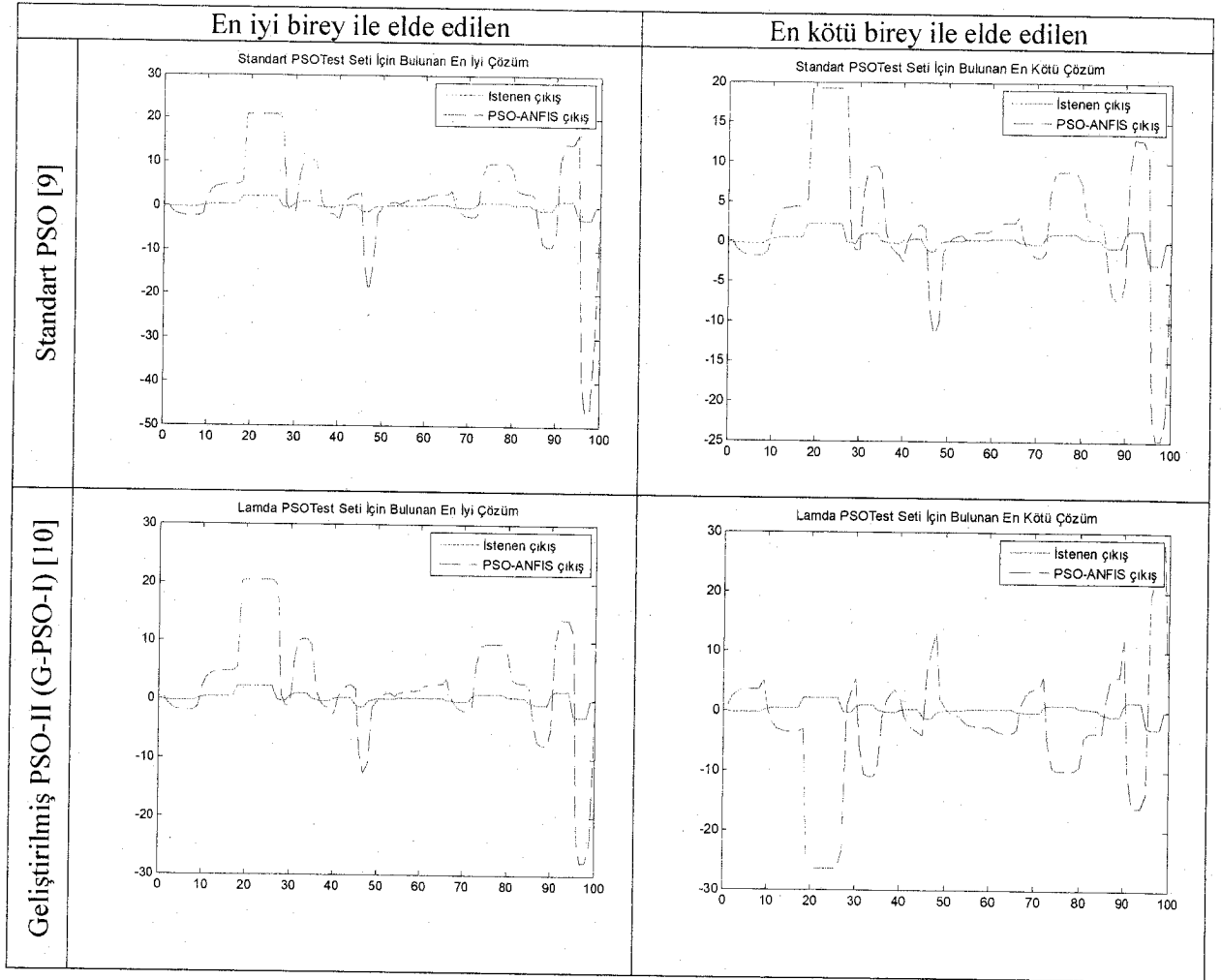
Şekil 34. PSO öğrenmeli ANFIS ile ÖDS 4 için *TEST* fazı sistem tanıma sonuçları



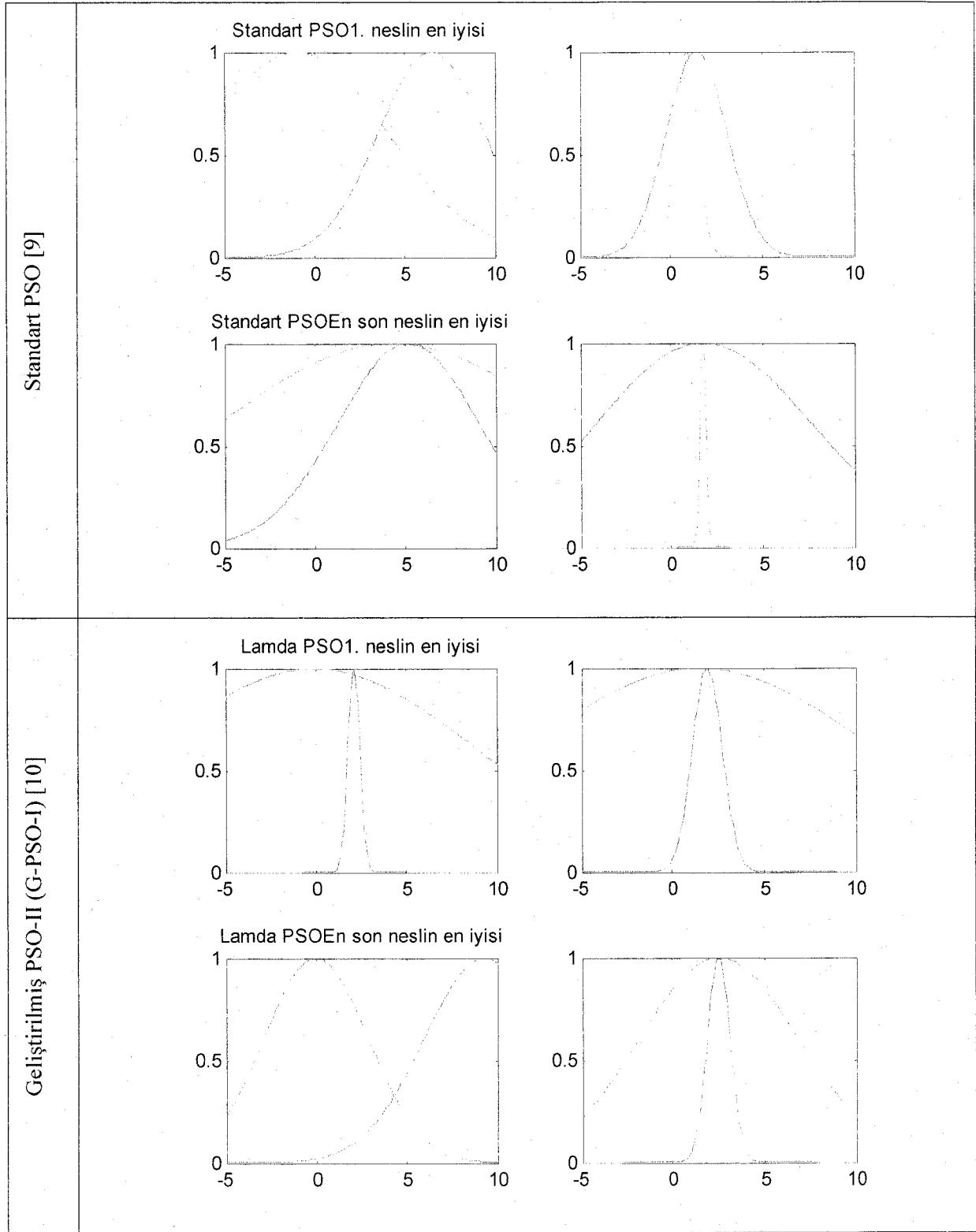
Şekil 35. PSO ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



Şekil 36. PSO öğrenmeli ANFIS ile ÖDS 5 için EĞİTİM fazı sistem tanıma sonuçları



Şekil 37. PSO öğrenmeli ANFIS ile ÖDS 5 için *TEST* fazı sistem tanıma sonuçları



Şekil 38. PSO ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları

4.2 ABC ve PSO algoritmaları ile elde edilen sonuçların değerlendirilmesi

Üçüncü bölüm ve bir önceki alt bölümde sonuçları grafiksel verilen sonuçlar, toplu olarak merkezi eğilim ölçüleri baz alınarak Tablo 7 ve Tablo 8’de verilmiştir. Tabloda koyu punto ile verilmiş değerler kategorisinde elde edilmiş en iyi sonucu göstermektedir.

Tablo 7. Eğitim fazı için ABC ve PSO algoritmaları kıyaslama tablosu

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	Ortalama	0.036994	0.045392	0.012165	0.050564	0.255944
	Std sapma	0.012073	0.032781	0.013409	0.019603	0.014218
	En iyi	0.013664	0.003924	0.004422	0.01568	0.236442
	En kötü	0.058886	0.145957	0.056209	0.092783	0.289824
	1 nesil süresi (s)	0.026451	0.02701	0.015753	0.024448	0.016379
	Başarım sayısı	2	1	2	3	0
	Ortalama Başarım Sayısı	8/25=0.32				
G-ABC-I	Ortalama	0.036842	0.040892	0.016855	0.050983	0.246771
	Std sapma	0.011271	0.022851	0.013653	0.018154	0.00849
	En iyi	0.016055	0.009197	0.005407	0.015181	0.235473
	En kötü	0.071235	0.117495	0.092333	0.106934	0.271231
	1 nesil süresi (s)	0.024406	0.025045	0.015585	0.024542	0.015048
	Başarım sayısı	3	0	0	0	5
	Ortalama Başarım Sayısı	8/25=0.32				
G-ABC-II	Ortalama	0.038039	0.046537	0.018718	0.052317	0.253359
	Std sapma	0.012515	0.025984	0.012315	0.016816	0.014552
	En iyi	0.015961	0.012833	0.004421	0.0135	0.235528
	En kötü	0.060354	0.111644	0.066238	0.105594	0.30777
	1 nesil süresi (s)	0.025276	0.0247	0.015538	0.025028	0.015218
	Başarım sayısı	0	1	2	2	0
	Ortalama Başarım Sayısı	5/25=0.2				
PSO	Ortalama	0.079258	0.030886	0.212148	0.120247	0.413184
	Std sapma	0.063467	0.027827	0.568219	0.043797	0.044641
	En iyi	0.027028	0.008052	0.005427	0.03366	0.319359
	En kötü	0.437995	0.197535	3.759978	0.220542	0.512254
	1 nesil süresi (s)	0.047024	0.045804	0.02863	0.047661	0.031011
	Başarım sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0/25=0				
G-PSO-I	Ortalama	0.084378	0.023013	0.172979	0.114145	0.449172
	Std sapma	0.056409	0.011735	0.460139	0.048272	0.090489
	En iyi	0.028781	0.009869	0.003724	0.043578	0.350809
	En kötü	0.296594	0.050653	1.96447	0.311882	0.820956
	1 nesil süresi (s)	0.048889	0.045896	0.030915	0.048223	0.033067
	Başarım sayısı	0	3	1	0	0
	Ortalama Başarım Sayısı	4/25=0.16				

Tablo 8. Test fazı için ABC ve PSO algoritmaları kıyaslama tablosu

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	En iyi	0.027788	0.020664	0.025316	0.094835	0.203715
	En kötü	1.97251	0.801725	547.1167	5.72806	1322.705
	Başarım Sayısı	0	1	0	0	0
	Ortalama Başarım Sayısı	1/10=0.1				
G-ABC-I	En iyi	0.032866	0.035922	0.017745	0.120933	0.201923
	En kötü	1.869034	1.317354	10.55646	21.5831	344.9939
	Başarım Sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	2/10=0.2				
G-ABC-II	En iyi	0.023195	0.028244	0.024321	0.08736	0.201685
	En kötü	2.494889	0.924081	5.162191	6.997366	154.9019
	Başarım Sayısı	1	0	1	0	1
	Ortalama Başarım Sayısı	3/10=0.3				
PSO	En iyi	0.054068	0.025113	0.023815	0.083096	0.324227
	En kötü	1.278393	1.094454	51.89824	0.732723	0.931145
	Başarım Sayısı	0	0	0	2	1
	Ortalama Başarım Sayısı	3/10=0.3				
G-PSO-I	En iyi	0.037084	0.022213	0.014848	0.119945	0.393983
	En kötü	1.066521	0.272192	18.25806	0.742309	5.139894
	Başarım Sayısı	1	1	1	0	0
	Ortalama Başarım Sayısı	3/10=0.3				

Tablo 7'de verilen sonuçlar beş, Tablo 8'de verilen sonuçlar ise iki ana kategoride ele alınmıştır. Tablo 7'yi baz alarak kıyaslama yapmak daha doğru olur. Tablo 7 baz alınarak ABC ve G-ABC-I algoritmalarının başarımları bu proje çalışmaları çerçevesinde en iyi başarımlar olarak değerlendirilebilir. Bu sonuçla ABC'nin PSO'dan daha iyi optimizasyon başarımı gösterdiği tartışmasızdır. Özellikle 1 nesil için gerekli hesaplama süresi kategorisinde, ABC ve türevleri PSO ve türevinden tüm denektaş sistemler için üstün başarımlar göstermişlerdir. Bu bağlamda nesil başına geçen süre bakımından standart ABC'nin ve geliştirilen türevlerinin tartışmasız üstün olduğu değerlendirilmiştir.

4.3 Farksal Gelişim (Differential Evolution, DE) Algoritması ile yapılan çalışmalar

Farksal Gelişim Algoritması Price ve Storn tarafından 1995 yılında geliştirilmiş, özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, isleyiş ve operatörleri itibariyle genetik algoritmaya dayanan popülasyon temelli sezgisel optimizasyon tekniğidir [11-13]. Temel olarak, DE algoritması popülasyon içerisinde rastgele seçilen iki

bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır. Şekil 39'da DE algoritmasının sözde kodu verilmektedir.

```

Input:  $Population_{size}, Problem_{size}, Weightingfactor, Crossover_{rate}$ 
Output:  $S_{best}$ 
Population  $\leftarrow$  InitializePopulation( $Population_{size}, Problem_{size}$ )
EvaluatePopulation(Population)
 $S_{best} \leftarrow$  GetBestSolution(Population)
While ( $\neg$  StopCondition())
    NewPopulation  $\leftarrow$   $\emptyset$ 
    For ( $P_i \in$  Population)
         $S_i \leftarrow$  NewSample( $P_i, Population, Problem_{size}, Weightingfactor, Crossover_{rate}$ )
        If ( $Cost(S_i) \leq Cost(P_i)$ )
            NewPopulation  $\leftarrow$   $S_i$ 
        Else
            NewPopulation  $\leftarrow$   $P_i$ 
        End
    End
    Population  $\leftarrow$  NewPopulation
    EvaluatePopulation(Population)
     $S_{best} \leftarrow$  GetBestSolution(Population)
End
Return ( $S_{best}$ )

```

Şekil 39. Klasik Farksal Gelişim Algoritmasının sözde kodu

DE algoritması 3 önemli parametreye sahiptir: skala faktörü (SF), çaprazlama olasılık sabiti (CR) ve popülasyon boyu (PS). Bir popülasyon PS kadar bireyi içerir. Skala faktörü (SF) çaprazlama olasılık sabitine (CR) göre daha hassastır [11]. DE algoritmasının başında PS optimizasyon parametrelerine göre belirlenir ve optimizasyon sürecinde değiştirilmez. Normal şartlarda popülasyon boyu (PS) optimizasyon parametre sayısının 5-10 katı olarak seçilir. Başlangıç popülasyonu rastgele seçilir. DE algoritması 3 temel operatöre sahiptir: mutasyon, çaprazlama ve seçim [13]. Mutasyon ve çaprazlama operatörleri yeni bireyler üretir ve seçim operatörü ile uygun olanlar belirlenir, bu şekilde popülasyonda sürekli en iyi bireylerin bulunması sağlanır.

Bu çalışma kapsamında pratikte en sık kullanılan DE stratejisi olan *rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlaması* seçilmiştir.

$$v_{i,g+1} = x_{i,g} + SF(x_{b,g} - x_{i,g}) + SF(x_{r_1,g} - x_{r_2,g}) \quad (7)$$

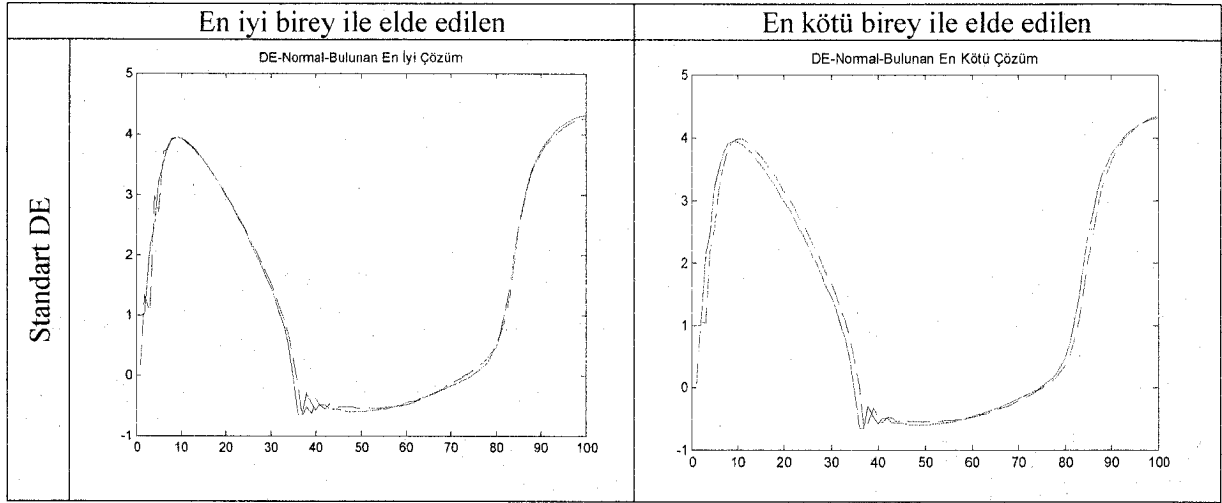
$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leq CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (8)$$

Bu denklemlerde $r_1, r_2, r_3 \in [1, PS]$ rastgele katsayılarıdır. $x_{b,g}$ popülasyon içerisindeki en iyi maliyet fonksiyonuna sahip bireyi, $v_{i,g}$ mutasyona uğramış bireyi, $u_{i,g}$ deneme bireylerini göstermektedir. Rastgele katsayılar birbirinden farklı olmak zorundadır. Skala faktörü (SF)

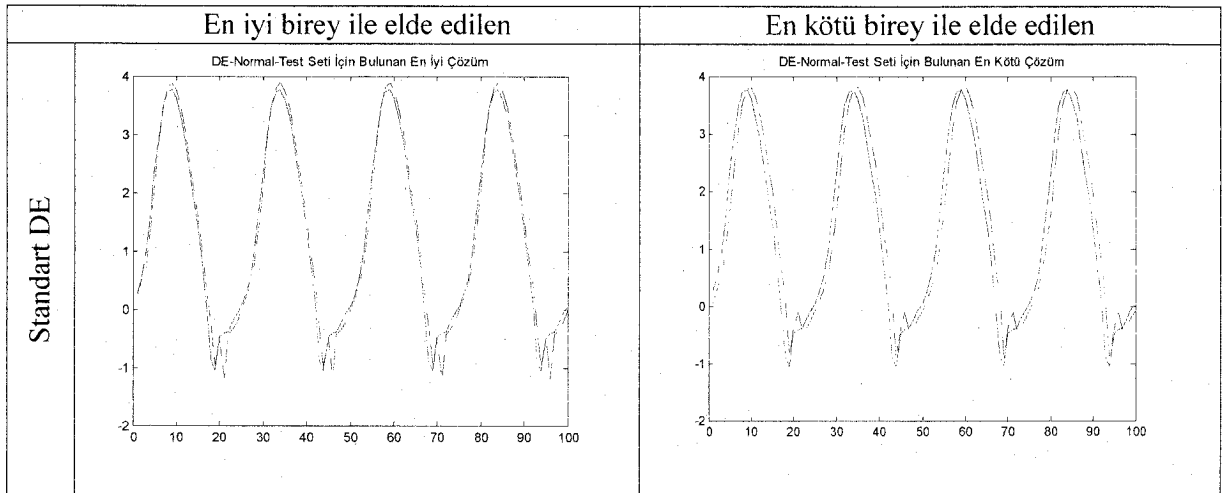
sabit bir deęerdir ve $0 \leq SF \leq 2$ aralıęında seęilir. Skala faktörü rastgele seęilen bireylerin fark vektörlerinin yükseltilmesinin kontrolünde kullanılır. Kullanılan DE algoritmasına ait parametreler Tablo 9'da verilmiştir. Bu algoritma 3. Bölümde verilen denektaşı dinamik sistemlerin ANFIS bulanık-nöral aę mimarisi ile modelleme amacıyla 50'şer kez koşturulmuştur. Elde edilen sonuçlar Şekil 40-54'de grafiksel olarak özetlenmiştir.

Tablo 9. DE algoritmasında kullanılan parametre deęerleri

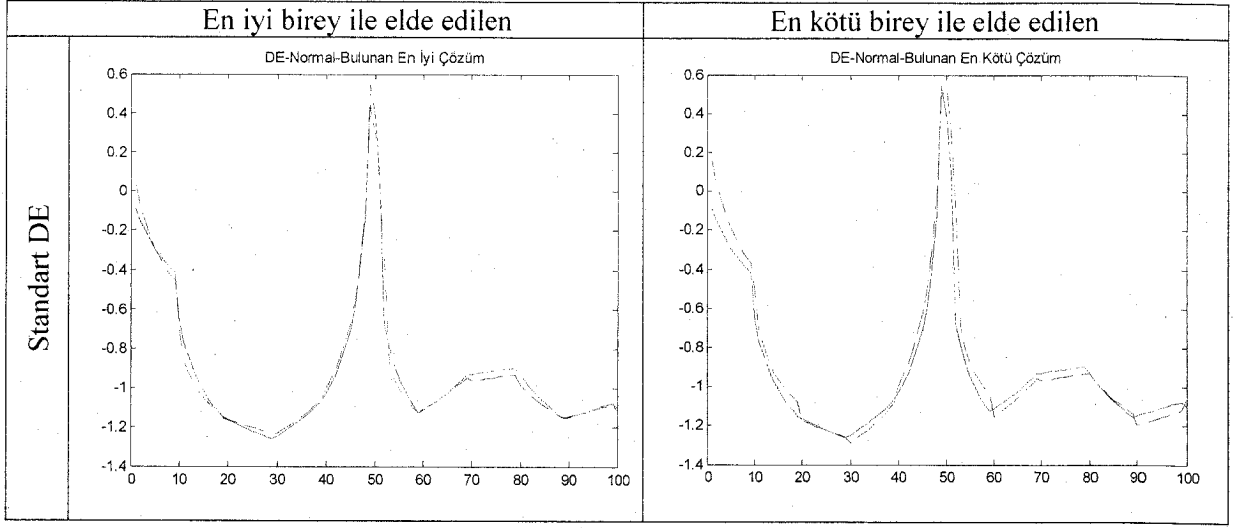
<i>Parametre Adı</i>	<i>Parametre Deęerleri</i>
<i>Populasyon büyüklüęü (NP)</i>	40
<i>Çaprazlama sabiti (CR)</i>	0.4
<i>Skala faktörü (SF)</i>	0.3



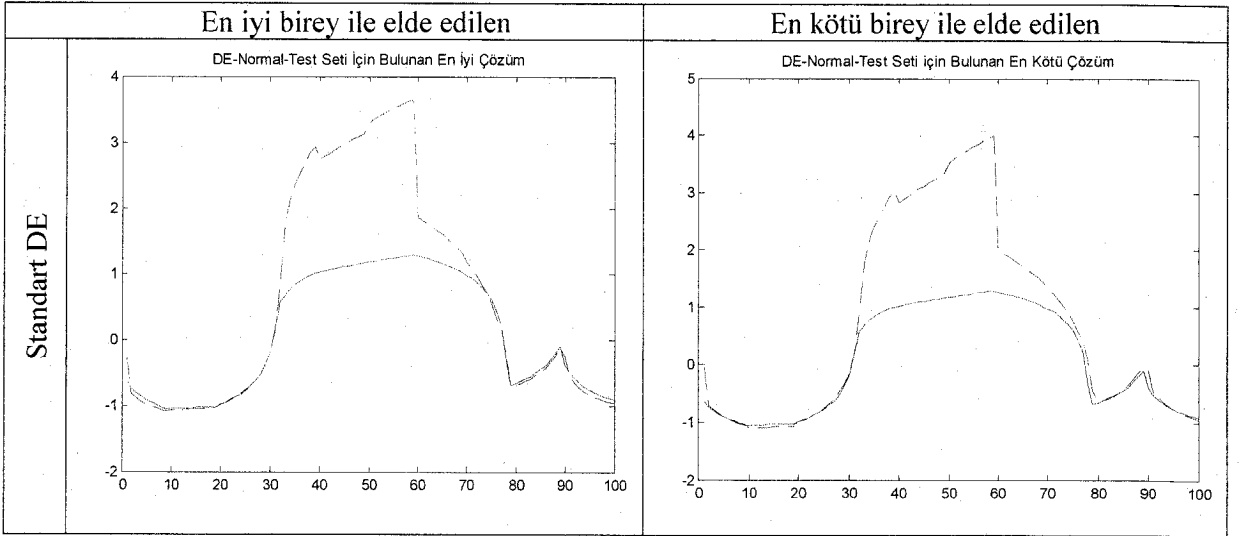
Şekil 40. DE öğrenmeli ANFIS ile ÖDS 1 için *EĞİTİM* fazı sistem tanıma sonuçları



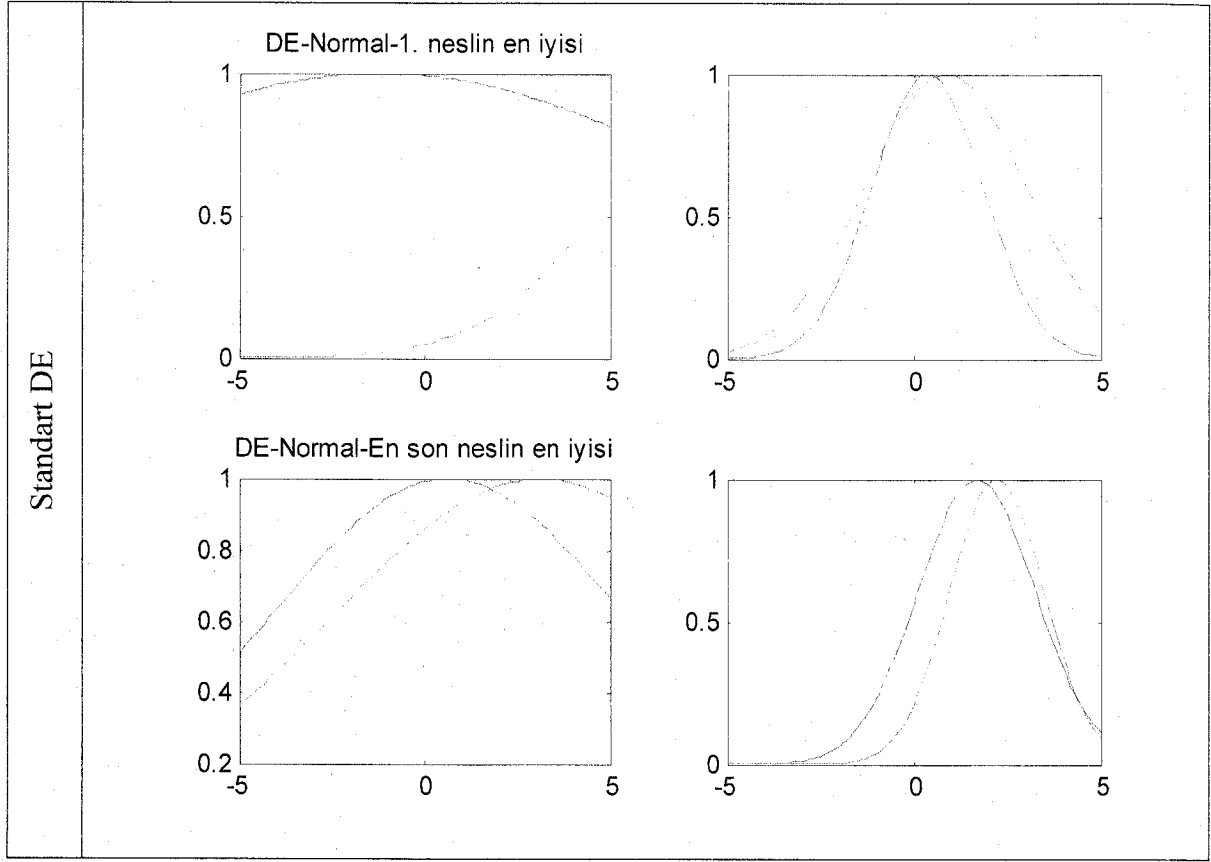
Şekil 41. DE öğrenmeli ANFIS ile ÖDS 1 için *TEST* fazı sistem tanıma sonuçları



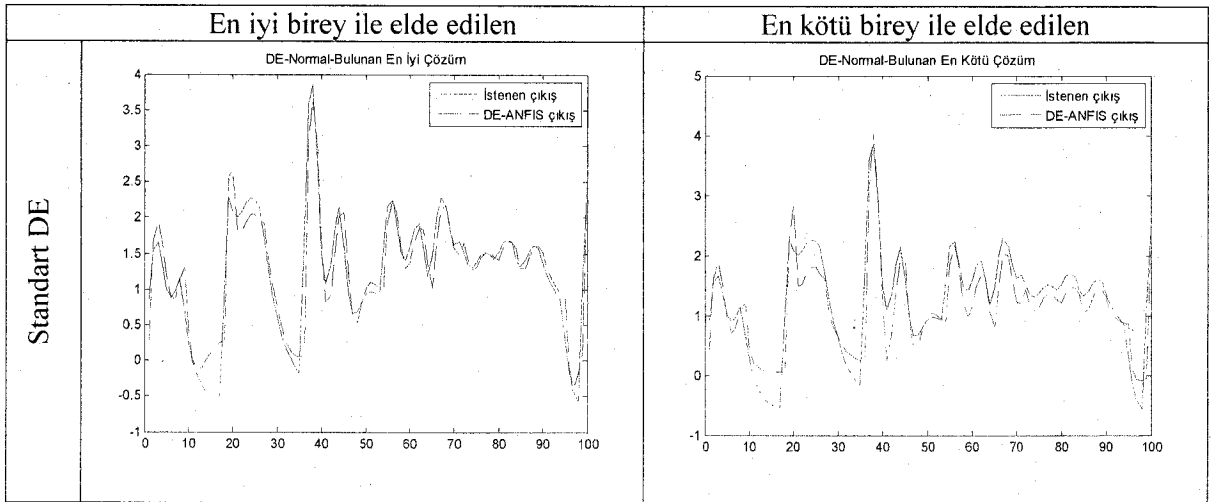
Şekil 46. DE öğrenmeli ANFIS ile ÖDS 3 için *EĞİTİM* fazı sistem tanıma sonuçları



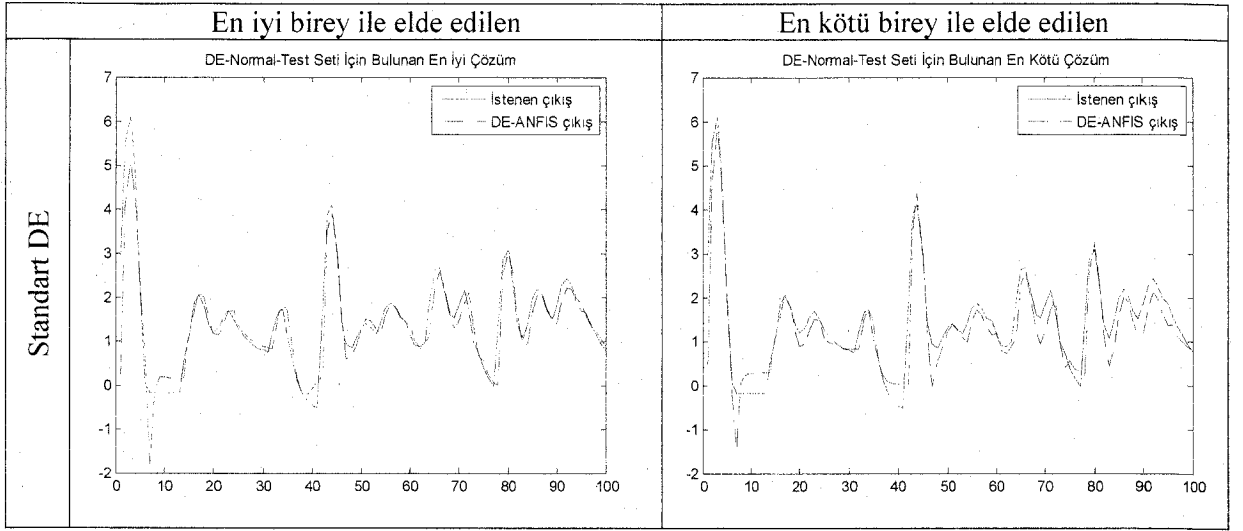
Şekil 47. DE öğrenmeli ANFIS ile ÖDS 3 için *TEST* fazı sistem tanıma sonuçları



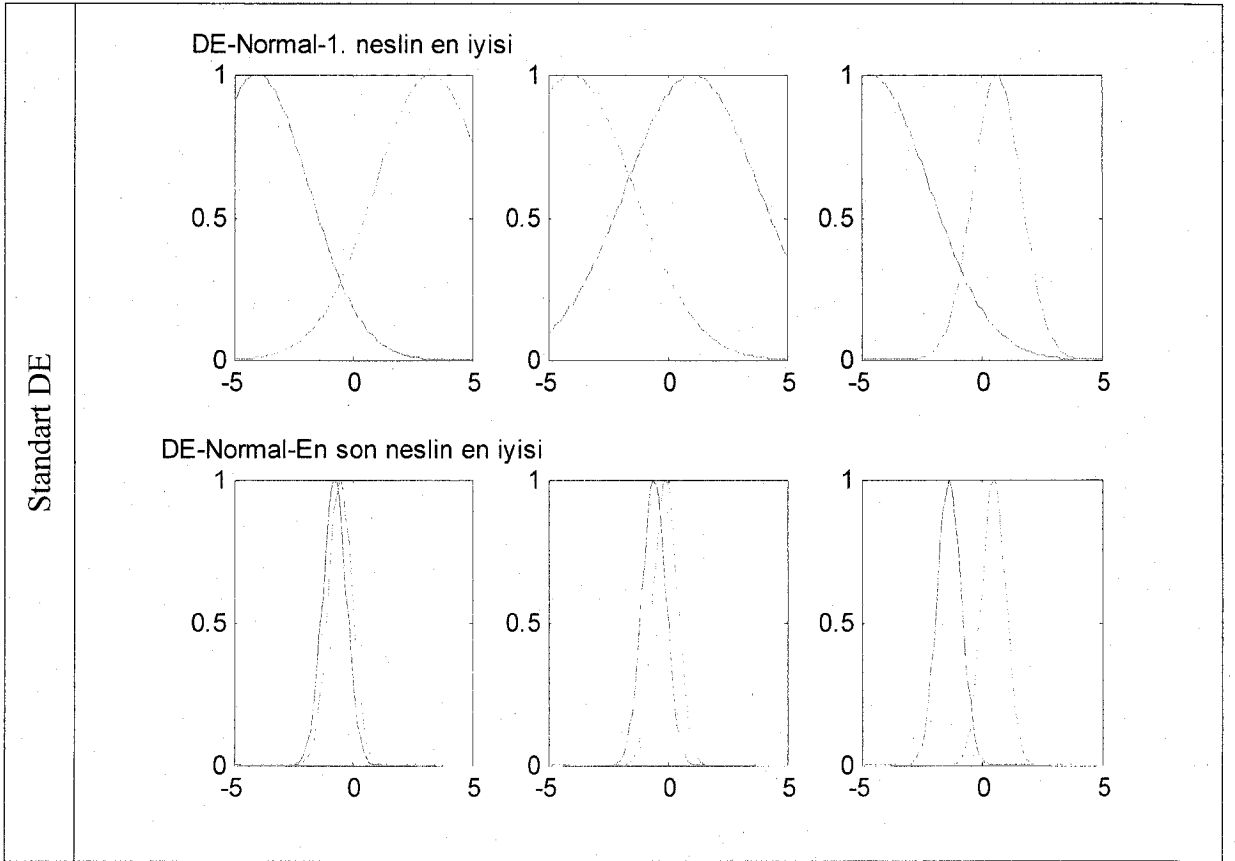
Şekil 48. DE ile eğitim fazında ÖDS 3 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



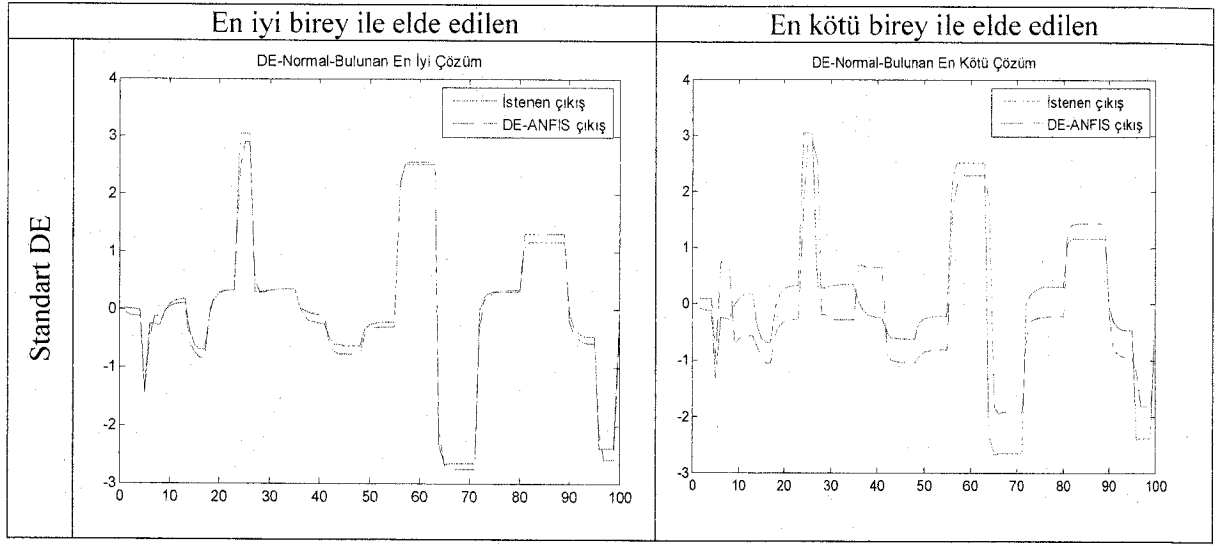
Şekil 49. DE öğrenmeli ANFIS ile ÖDS 4 için EĞİTİM fazı sistem tanıma sonuçları



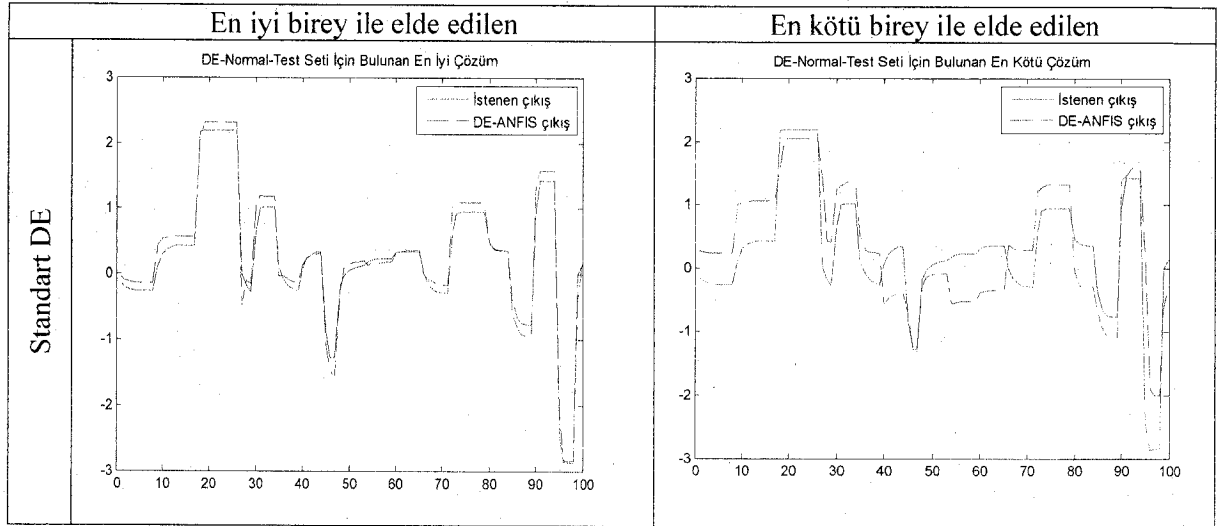
Şekil 50. DE öğrenmeli ANFIS ile ÖDS 4 için *TEST* fazı sistem tanıma sonuçları



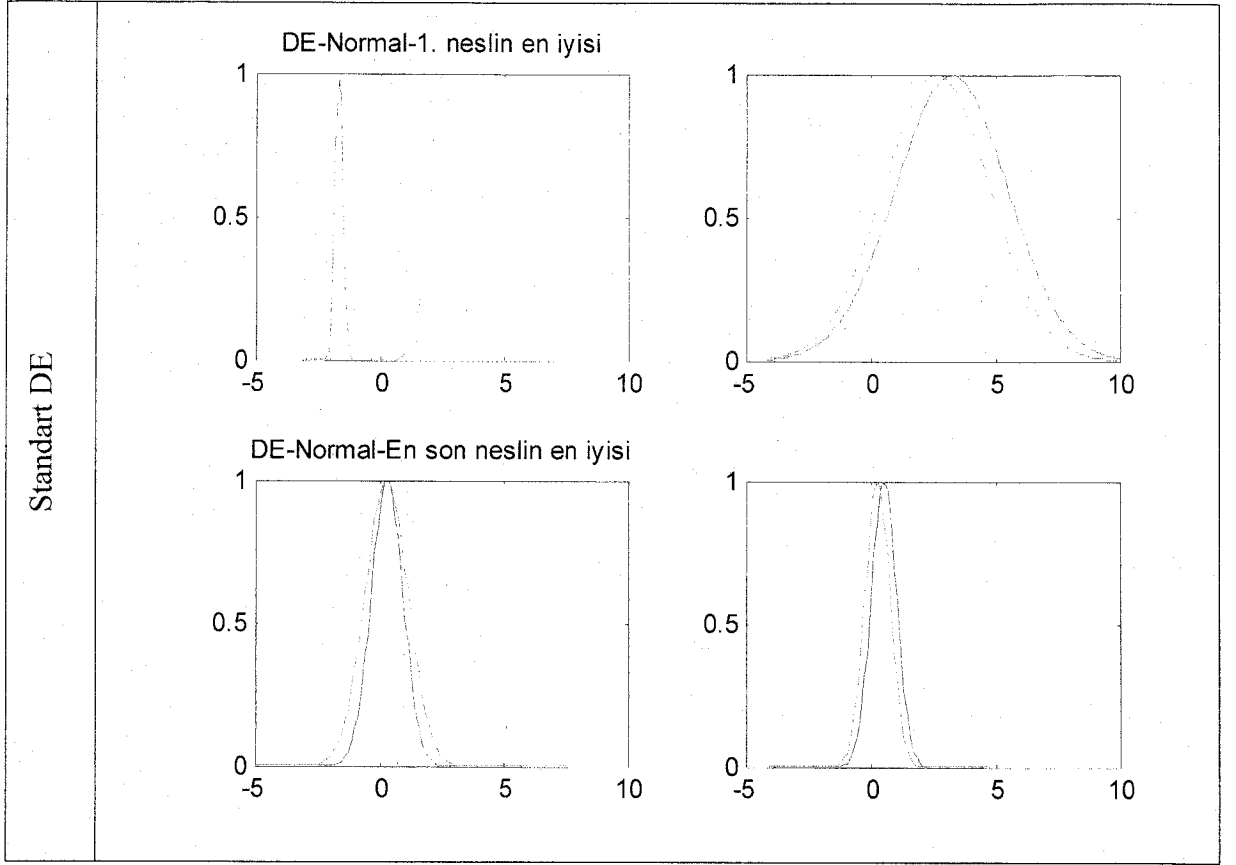
Şekil 51. DE ile eğitim fazında ÖDS 4 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları



Şekil 52. DE öğrenmeli ANFIS ile ÖDS 5 için *EĞİTİM* fazı sistem tanıma sonuçları



Şekil 53. DE öğrenmeli ANFIS ile ÖDS 5 için *TEST* fazı sistem tanıma sonuçları



Şekil 54. DE ile eğitim fazında ÖDS 5 için elde edilen ANFIS giriş ÜF'lerinin ilk ve son durumları

4.2. ABC ve DE algoritmaları ile elde edilen sonuçların değerlendirilmesi

Bir önceki alt bölümde sonuçları grafiksel verilen DE algoritması sonuçları, toplu olarak merkezi eğilim ölçüleri temel alınarak Tablo 10 ve Tablo 11'de verilmiştir. Tabloda koyu punto ile verilmiş değerler kategorisinde elde edilmiş en iyi sonucu göstermektedir.

Tablo 10'da verilen sonuçlar beş, Tablo 11'de verilen sonuçlar ise iki ana kategoride ele alınmıştır. Tablo 10'da verilen eğitim fazı sonuçları ABC algoritmasının nispeten başarılı olduğunu göstermektedir. Ancak DE algoritmasının performansının genel olarak ABC algoritmasının performansına yakın bir sonuç verdiği söylenebilir. Test fazı için yapılan karşılaştırma tablosunda (Tablo 11) ise DE ve ABC algoritmalarının performansları bakımından birbirlerine üstünlükleri konusunda açık bir fark görülememektedir.

Algoritmaların bir nesil için koşturma süresine bakıldığında ise ABC algoritmasının DE algoritmasına göre daha hızlı sonuç verdiği gözlenmiştir. Populasyondaki çeşitlilik bakımından nesildeki en kötülere, ortalama ve standart sapmalara göre ABC algoritmasının DE algoritmasına göre başarılı olduğu söylenebilir.

Tablo 10. Eğitim fazı için ABC ve DE algoritmaları kıyaslama tablosu

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	Ortalama	0.036994	0.045392	0.012165	0.050564	0.255944
	Std sapma	0.012073	0.032781	0.013409	0.019603	0.014218
	En iyi	0.013664	0.003924	0.004422	0.01568	0.236442
	En kötü	0.058886	0.145957	0.056209	0.092783	0.289824
	1 nesil süresi (s)	0.026451	0.02701	0.015753	0.024448	0.016379
	Başarım sayısı	4	3	1	4	4
	Ortalama Başarım Sayısı	16/25=0.64				
DE	Ortalama	0.040186	0.020487	0.004344	0.121729	0.293995
	Std sapma	0.005373	0.017221	0.001152	0.006401	0.151655
	En iyi	0.033473	0.011401	0.002454	0.112035	0.042477
	En kötü	0.060116	0.097258	0.007732	0.143402	0.56617
	1 nesil süresi (s)	0.038329	0.038061	0.034161	0.038154	0.036378
	Başarım sayısı	1	2	4	1	1
	Ortalama Başarım Sayısı	9/25=0.36				

Tablo 11. Test fazı için ABC ve DE algoritmaları kıyaslama tablosu

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	En iyi	0.027788	0.020664	0.025316	0.094835	0.203715
	En kötü	1.97251	0.801725	547.1167	5.72806	1322.705
	Başarım Sayısı	1	1	1	1	0
	Ortalama Başarım Sayısı	4/10=0.4				
DE	En iyi	0.03858	0.025721	0.040383	0.206293	0.037659
	En kötü	0.198069	0.405192	2.051066	0.854747	0.613042
	Başarım Sayısı	1	1	1	1	2
	Ortalama Başarım Sayısı	6/10=0.6				

5. Sonuç

Dördüncü bölümde detaylıca verilen sonuçlar toplu olarak merkezi eğilim ölçüleri temel alınarak Tablo 12 ve Tablo 13'de verilmiştir. Tabloda koyu punto ile verilmiş değerler kategorisinde elde edilmiş en iyi sonucu göstermektedir. Ortalama başarımları (OBS) baz alınarak en iyi başarımları gösteren algoritmaya ait kısımlar tablolarda renklendirilmiştir. Tablo 12'de verilen sonuçlar beş, Tablo 12'de verilen sonuçlar ise iki ana kategoride ele alınmıştır.

Tablo 12'de verilen eğitim fazı sonuçlarından, OBS baz alınarak, DE algoritmasının en başarılı algoritma olduğunu görülmektedir. Test fazı için yapılan karşılaştırma tablosunda (Tablo 13) ise yine DE algoritmasının başarımlarının iyi olduğu görülmektedir. Algoritmaların bir nesil için koşma süresine bakıldığında ise ABC algoritmasının DE algoritmasına göre daha hızlı sonuç verdiği gözlenmiştir. Diğer kategorilerde en iyi başarımları sahip DE, nesil başına koşma süresinde ise en kötü başarımları sahip olduğu tablolardan görülebilir. Hatta DE'nin nesil başına koşma süresi ABC, G-ABC-I ve G-ABC-II'nin koşma sürelerinin yaklaşık iki katıdır.

Bu proje çalışmalarından elde edilen sonuçlara göre genel olarak bulanık sistem modellemede DE'nin daha başarılı olduğu ancak ABC ve türevlerinin başarımlarının DE'nin başarımlarına çok yakın olduğu sonucuna varılmıştır. İncelemede ele alınan PSO algoritmasının başarımlarının ise ABC ve DE'nin başarımlarına göre daha kötü olduğu görülmüştür.

Bu proje kapsamında incelenen algoritmaların gerçek zamanlı uygulamalarında en önemli parametre nesil başına koşma süresidir. Bu bakımdan ABC ve türevlerinin tartışmasız üstün olduğu görülmüştür. Bu bakımdan, özellikle gömülü sistem tabanlı gerçek zamanlı uygulamalarda, hem kolay işlemler hem de DE'ye göre yarı yarıya daha az nesil başına koşma süresi olması sebebiyle ABC ve bu projede geliştirilen türevlerinin (G-ABC-I ve G-ABC-II) tercih edilmesinin gerektiği sonucuna varılmıştır.

6. Bilimsel Etkinlikler

Bu raporun ikinci bölümünde verilen çalışmalardan bir bildiri hazırlanmış ve *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU 2012)*'nda sözlü olarak 4 Temmuz 2012 tarihinde sunulmuştur. Bu bildirinin bir sureti EK-1'de verilmiştir.

Tablo 12. ABC, PSO ve DE algoritmalarının ÖDS modellemede *EĞİTİM* fazı kıyaslaması

Algoritma	Ölçü	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	Ortalama	0.036994	0.045392	0.012165	0.050564	0.255944
	Std sapma	0.012073	0.032781	0.013409	0.019603	0.014218
	En iyi	0.013664	0.003924	0.004422	0.01568	0.236442
	En kötü	0.058886	0.145957	0.056209	0.092783	0.289824
	1 nesil süresi (s)	0.026451	0.02701	0.015753	0.024448	0.016379
	Başarım sayısı	1	1	0	3	0
	Ortalama Başarım Sayısı	5/25=0.20				
G-ABC-I	Ortalama	0.036842	0.040892	0.016855	0.050983	0.246771
	Std sapma	0.011271	0.022851	0.013653	0.018154	0.00849
	En iyi	0.016055	0.009197	0.005407	0.015181	0.235473
	En kötü	0.071235	0.117495	0.092333	0.106934	0.271231
	1 nesil süresi (s)	0.024406	0.025045	0.015585	0.024542	0.015048
	Başarım sayısı	2	0	0	0	4
	Ortalama Başarım Sayısı	6/25=0.24				
G-ABC-II	Ortalama	0.038039	0.046537	0.018718	0.052317	0.253359
	Std sapma	0.012515	0.025984	0.012315	0.016816	0.014552
	En iyi	0.015961	0.012833	0.004421	0.0135	0.235528
	En kötü	0.060354	0.111644	0.066238	0.105594	0.30777
	1 nesil süresi (s)	0.025276	0.0247	0.015538	0.025028	0.015218
	Başarım sayısı	0	1	1	1	0
	Ortalama Başarım Sayısı	3/25=0.12				
PSO	Ortalama	0.079258	0.030886	0.212148	0.120247	0.413184
	Std sapma	0.063467	0.027827	0.568219	0.043797	0.044641
	En iyi	0.027028	0.008052	0.005427	0.03366	0.319359
	En kötü	0.437995	0.197535	3.759978	0.220542	0.512254
	1 nesil süresi (s)	0.047024	0.045804	0.02863	0.047661	0.031011
	Başarım sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0				
G-PSO-I	Ortalama	0.084378	0.023013	0.172979	0.114145	0.449172
	Std sapma	0.056409	0.011735	0.460139	0.048272	0.090489
	En iyi	0.028781	0.009869	0.003724	0.043578	0.350809
	En kötü	0.296594	0.050653	1.96447	0.311882	0.820956
	1 nesil süresi (s)	0.048889	0.045896	0.030915	0.048223	0.033067
	Başarım sayısı	0	2	0	0	0
	Ortalama Başarım Sayısı	2/25=0.08				
DE	Ortalama	0.039115	0.018553	0.00405	0.123121	0.281597
	Std sapma	0.004217	0.098136	0.001221	0.01032	0.139422
	En iyi	0.33775	0.010484	0.002383	0.112719	0.020042
	En kötü	0.054579	0.131649	0.00828	0.156764	0.551824
	1 nesil süresi (s)	0.040384	0.041445	0.034371	0.040374	0.32614
	Başarım sayısı	2	1	4	1	1
	Ortalama Başarım Sayısı	9/25=0.36				

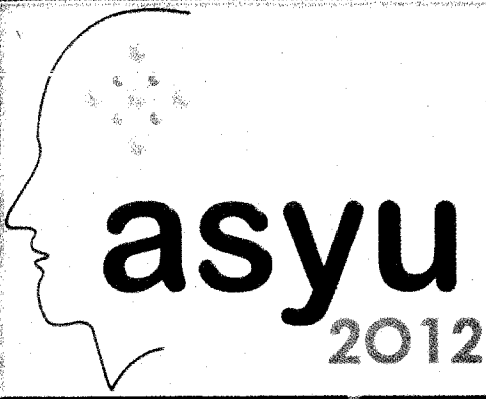
Tablo 13. ABC, PSO ve DE algoritmalarının ÖDS modellemede *TEST* fazı kıyaslaması

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	En iyi	0.027788	0.020664	0.025316	0.094835	0.203715
	En kötü	1.97251	0.801725	547.1167	5.72806	1322.705
	Başarım Sayısı	0	1	0	0	0
	Ortalama Başarım Sayısı	1/10=0.1				
G-ABC-I	En iyi	0.032866	0.035922	0.017745	0.120933	0.201923
	En kötü	1.869034	1.317354	10.55646	21.5831	344.9939
	Başarım Sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0/5=0				
G-ABC-II	En iyi	0.023195	0.028244	0.024321	0.08736	0.201685
	En kötü	2.494889	0.924081	5.162191	6.997366	154.9019
	Başarım Sayısı	1	0	0	0	0
	Ortalama Başarım Sayısı	1/10=0.1				
PSO	En iyi	0.054068	0.025113	0.023815	0.083096	0.324227
	En kötü	1.278393	1.094454	51.89824	0.732723	0.931145
	Başarım Sayısı	0	0	0	2	0
	Ortalama Başarım Sayısı	2/10=0.2				
G-PSO-I	En iyi	0.037084	0.022213	0.014848	0.119945	0.393983
	En kötü	1.066521	0.272192	18.25806	0.742309	5.139894
	Başarım Sayısı	0	1	1	0	0
	Ortalama Başarım Sayısı	2/10=0.2				
DE	En iyi	0.047424	0.024831	0.031532	0.204259	0.025181
	En kötü	0.195956	0.675152	2.356835	0.919691	0.640239
	Başarım Sayısı	1	0	1	0	2
	Ortalama Başarım Sayısı	4/10=0.4				

Kaynaklar

- [1] D. Karaboğa, «An idea based on honey bee swarm for numerical optimization,» Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [2] R. Babuska and S. Stramigioli, "Matlab and simulink for modeling and control, Lectuter Note," Delft University of Technology, Faculty of Information Technology and Systems, Control Laboratory, Delft, 1999.
- [3] C. Karakuzu, Ö. Yıldırım, U. Yüzgeç, M. Kesler, " ABC Algoritmasının PID Denetleç Parametrelerinin Optimizasyonunda Kıyaslamalı Başarımı" ASYU 2012 Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu Bildiriler Kitabı, Sf. 101-105, Trabzon, Türkiye, 3-4 Temmuz 2012.
- [4] R. J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system", IEEE Trans. on Systems, Man, and Cybernetics, vol. 23, No. 3, pp. 665-684, 1993.
- [5] K. S. Narendra, K. Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Transactions on Neural Networks, vol. 1, no 1, pp. 4-27, 1990.
- [6] R. Babuska, "Fuzzy System, Modeling and Identification", <http://www.dcsc.tudelft.nl/~babuska/transp/fuzzmod.pdf> (son erişim 19 Şubat 2012).
- [7] Y. Oussar, I. Rivals, L. Dreyfus, "Training wavelet networks for nonlinear dynamic input output modeling", Neurocomputing, cilt 20, syf 173-188, 1998.
- [8] P. S. Sastry, G. Santharam, K. P. Unnikrishnan, "Memory neuron networks for identification and control of dynamical systems", IEEE Transaction on Neural Networks, cilt 5, no 2, 1994.
- [9] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," IEEE Trans. Evol. Comput., vol. 6, no. 1, pp. 58-73, 2002.
- [10] Mehmet Ali Cavuslu, Cihan Karakuzu, Fuat Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning", Appl. Soft Comput. J. (2012), doi:10.1016/j.asoc.2012.03.022, In Press, Corrected Proof, Available online 17 March 2012.
- [11] K. Price, & R. Storn, "Differential evolution: A simple evolution strategy for fast optimization", Dr. Dobb's J. Software Tools, 22 (4), 18-24, 1997.
- [12] R. Storn, & K. Price, "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces", J. Global Optimization, 11, 341-359,1997.
- [13] K. Price, K., "An Introduction to Differential Evolution", D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 79-108, ISBN: 007-709506-5, New Ideas in Optimization.

EK-1: ASYU 2012'de sunulan bildiri



AKILLI SİSTEMLERDE YENİLİKLER VE UYGULAMALARI SEMPOZYUMU



2012

3-4 Temmuz 2012, Trabzon

<http://asyu.inista.org>

ABC Algoritmasının PID Denetleç Parametrelerinin Optimizasyonunda Kıyaslamalı Başarımı

Cihan KARAKUZU

Özlem YILDIRIM

Uğur YÜZGEÇ

Metin KESLER

Bilgisayar Mühendisliği
Bölümü

Mühendislik Fakültesi

Bilecik Şeyh Edebali

Üniversitesi, Gölümbe,

BİLECİK

cihan.karakuzu.@bilecik.edu.tr

Bilgisayar Mühendisliği
Anabilim

Dalı, F.B.E.

Bilecik Şeyh Edebali

Üniversitesi, Gölümbe,

BİLECİK

ozlemsarac2002@gmail.com

Elektrik-Elektronik

Mühendisliği Bölümü

Mühendislik Fakültesi

Bilecik Şeyh Edebali

Üniversitesi, Gölümbe,

BİLECİK

ugur.yuzgec@bilecik.edu.tr

Bilgisayar

Mühendisliği Bölümü

Mühendislik Fakültesi

Bilecik Şeyh Edebali

Üniversitesi, Gölümbe,

BİLECİK

metin.kesler@bilecik.edu.tr

Özet

Bu bildiri, sezgisel arama algoritmalarından biri olan ABC (Artificial Bee Colony) algoritmasının başarımı ile karmaşık problemlerin çözümünde yaygın olarak kullanılan gelen diğer popüler sezgisel arama algoritmalarının (GA, TACO, PSO, DE) başarımlarının kıyaslama sonuçları verilmiştir. Kıyaslama DC motor hız denetiminde PID denetleç parametrelerinin optimizasyonu problemi üzerinde yapılmıştır. Algoritmalar problem üzerinde aynı sürü büyüklüğü ve maksimum nesil sayısı kullanılarak 100'er kez koşurulmuş, elde edilen sonuçlar merkezi eğilim ölçüleri ile değerlendirilmiştir. Elde edilen sonuçlar, ABC algoritmasının özellikle bir nesil için gerekli koşma süresi bakımından üstünlüğünü göstermiştir.

1. Giriş

Mühendislik bilimlerinin uygulamalarında karşılaşılan problemlerin çoğunluğu doğrusal olmayan bir davranış sergiler. Bu tip problemlerin nümerik çözümlerinin zor ya da mümkün olmadığı durumlarda kullanılmak üzere doğal yaşamda gözlenen bazı olaylardan esinlenerek çeşitli sezgisel algoritmalar geliştirilmiştir. Özellikle çok parametrelili problemlerin optimizasyonu çok boyutlu uzayda arama gerektirdiğinden, bu tip problemlerin klasik optimizasyon yöntemleri ile çözüm sürecinde zorluklar kaçınılmazdır.

Bu zorlukları aşmada sezgisel algoritmaların gösterdiği başarımlar, bu algoritmaların bilimsel camiada bir çok alanda yaygın uygulama alanı bulmasına sebep olmuştur. Bu sezgisel yaklaşımlardan Genetik Algoritma (GA), Karınca Kolonisi Algoritması (ACO), Parçacık Sürü Optimizasyonu (PSO) ve Farksal Gelişim (Differential Evolution, DE) algoritması en yaygın kullanılanlar arasında sayılabilir.

ABC (Artificial Bee Colony) algoritması 2005 yılında bal arılarının davranışlarından esinlenerek Derviş Karaboğa tarafından tanımlanmış [1] yeni

bir sezgisel optimizasyon algoritmasıdır. Algoritmanın ön plana çıkan özelliği, işleyişinin PSO ve DE kadar basitliğinin yanı sıra sadece koloni büyüklüğü ve maksimum limit döngü gibi genel denetim parametrelerini kullanıyor olmasıdır.

Bu çalışmada ABC'nin PID parametrelerinin ayarlanması problemi üzerinde başarımı dört farklı sezgisel arama algoritması ile istatistiksel kıyaslanmış ve bu algoritmaların başarımları değerlendirilmiştir.

2. ABC Algoritması

ABC optimizasyon algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak arama uzayındaki olası çözümlerden en uygununu bulmaya çalışmaktadır. ABC algoritmasının temel işleyişi Şekil 1'de verilen kaba kodla özetlenebilir. Şekil 1'de özetlenen işlemleri sırası ile [2]'den tanımlayalım: Başlangıç yiyecek kaynaklarının üretilmesi (1) ile yapılır. Eşitlikte $i=1...N/2$, $j=1...D$, x_j^{min} ve x_j^{max} sırasıyla j . parametrenin alt ve üst sınır değeridir.

$$x_{ij} = x_j^{min} + rand(0,1) * (x_j^{max} - x_j^{min}) \quad (1)$$

(1) ile atanan işçi arılar için atanan k indisli bir komşu kullanılarak yeni bir yiyecek kaynağı üretilmesi (2) ile yapılır. Eşitlikte j [1 D] aralığında rasgele üretilen tam sayı ve θ [-1 1] aralığında rasgele üretilen sayıdır.

$$u_{ij} = x_{ij} + \theta(x_{ij} - x_{kj}) \quad (2)$$

Gözcü arıların seleksiyonda kullanacakları p_i olasılık değerlerinin hesaplanması (3) ile yapılır. Eşitlikte fit_i i . işçi arının uygunluk değeri olup N kolonideki arı sayısıdır.

$$p_i = fit_i / \sum_{m=1}^{N/2} fit_m \quad (3)$$

Gözcü arıların (2) ile yeni bir çözüm üretilip üretmeyecekleri [0 1] aralığında rasgele üretilen sayının (3) ile hesaplanan p_i 'den küçük ya da büyük olmasına bağlıdır. Algoritmada koloni büyüklüğü (N) genellikle 20-50 arası, yiyecek kaynağının terk edilmesi ile ilgili limit (LC) değer için ise $N \times D$ değeri önerilmektedir [2].

```

Arama uzayı boyutunu (D) belirle
Maksimum nesil sayısı (Gmax), sürü büyüklüğü (N), kaynağı bırakma limiti (LC) ata
Başlangıç yiyecek kaynaklarını rasgele ata (x)
N/2 işçi arı için başlangıç uygunluk değeri (fit) hesapla
ite=1
ite=Gmax oluncaya değin Tekrarla1
  i=1
  i=N/2 oluncaya değin Tekrarla2
    [1 N/2] aralığında i. işçi arı için rasgele komşu seç (k)
    i. işçi arı için k komşunu da kullanarak yeni yiyecek kaynağı üret (ui)
    ui ve xi arasında aç gözlü seçme işlemini uygula
    Eğer ui faydasız ise faili= faili+1
    i. işçi arı pi olasılık değerini hesapla
    pi olasılık değerine göre gözcü arı ata (vi) ve değerlendir
    vi ve xi arasında aç gözlü seçme işlemini uygula
    Eğer vi faydasız ise faili= faili+1
  i=i+1
Tekrarla2_son
Faydasız çözüm sayısı (fail) ≥ LC olan arılar yerine kaşif arı ata (yeni bir
rasgele çözüm üret)
Şimdiye kadar elde edilen en iyi çözümü (arıyı) sakla
ite=ite+1
Tekrarla1_son
Elde edilen en iyi çözümü (arıyı) problemin çözümü olarak ata

```

Şekil 1. ABC algoritmasının kaba kodu.

3. Kıyaslamada Kullanılan Algoritmalar

Bu bölümde ABC algoritmasının başarımını kıyaslamada kullanılan dört sezgisel arama algoritmasına ait temel bilgiler verilmiştir.

3.1. Genetik Algoritması (GA)

Genetik algoritmalar doğada ancak daha iyi ve daha kuvvetli olan bireylere yaşama şansının verildiği biyolojik olaylara benzetilerek John Holland [3] tarafından 1975'te geliştirilmiş sezgisel bir arama algoritmasıdır. Çözüm için "iyi"nin ne olduğunu belirleyen bir uygunluk (fitness) fonksiyonu ve yeni çözümler üretmek için doğal seleksiyon, çaprazlama, mutasyon ve inversion operatörleri kullanılır. GA kromozom adı verilen 0 ve 1'lerden oluşan dizi şeklinde bireylerden oluşan sürüde aşağıda verilen işlem basamakları ile en iyi çözümü arar.

1. Başlangıç sürüsünü oluştur
2. Sürünün her bir bireyi için uygunluk değeri hesapla
3. Kromozomları uygunluk değerlerine göre seç ve seçilmiş kromozomları eşleşme havuzuna kopyala ve çiftler halinde grupta
4. Çaprazlama: İki ebeveyn kromozomun arasında belirlenen parçalar takas et
5. Mutasyon: Genetik çeşitliliği sağlamak amacıyla yeni bireyin bir bitini değiştir
6. Döngü: Nesil sayısı kadar 2. adıma dön
7. Son: Sürüdeki en iyi uygunluğa sahip olan kromozomu çözüm olarak ata

3.2. Farksal Gelişim Algoritması (DE)

Farksal Gelişim Algoritması Price ve Storn [4] tarafından 1995 yılında geliştirilmiş işleyiş ve operatörleri itibarıyla genetik algoritmaya (GA)

dayanan popülasyon temelli sezgisel bir optimizasyon tekniğidir.

DE algoritmasının temel fikri sürüdeki iki birey arasındaki farkın üçüncü bir bireye ilave edilmesine dayanmaktadır. DE algoritmalarında 10 farklı strateji vardır. Bu çalışmada algoritmanın en yalın hali olan ve *DE/rand-to-best/1/bin* olarak adlandırılan strateji [5] kullanılmıştır. Bu adlandırmada, ilk kısım karıştırılacak bireyi, ikinci kısım ilk kısım ile çaprazlama yapılacak fark vektörlerinin sayısını son kısım ise çaprazlama tipini göstermektedir. Bu stratejiyi kullanan DE algoritmasının işleyişi kısaca aşağıda verilmiştir.

1. DE algoritmasının parametrelerinin belirlenmesi.
2. Başlangıç popülasyonunun rastgele oluşturulması
3. Her bir bireyin uygunluk değerinin hesaplanması ve en iyi bireyin tespit edilerek saklanması
4. Popülasyon içinden seçilen stratejiye uygun farklı bireylerin seçilmesi, mutasyon ve çaprazlama işlemleri ile yeni popülasyonun oluşturulması
5. Eski popülasyon ve yeni oluşturulan popülasyondaki bireylerin uygunluk değerlerine göre yeni popülasyon bireylerinin seçilmesi
6. Popülasyondaki en iyi bireyin güncellenmesi
7. Eğer maksimum nesil sayısına ulaşılmadı ise 4. adıma gidilmesi, aksi halde aramanın sonlandırılması
8. Son adımdaki en iyi bireylerin çözüm olarak kullanılması.

3.3. Parçacık Sürü Algoritması (PSO)

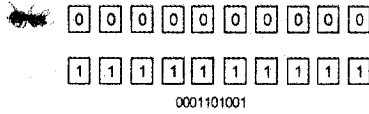
Parçacık sürü optimizasyonu algoritması 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiş, bireysel ve sosyal etkileşimi kullanarak bireylerini arama uzayında en anlamlı bölgeye yönlendiren sürü tabanlı sezgisel bir arama algoritmasıdır. Algoritma rasgele atanmış

bireylerle (parçacık) başlayarak her bir nesilde parçacıkların hızları ve pozisyonları güncellenir. Bu çalışmada parçacık hızlarının güncellenmesi için algoritmanın yalın hallerinden biri olan (4)'de verilen yöntem [6] kullanılmıştır.

$$v_i(n) = \xi \left[v_i(n-1) + \alpha_1 r_1 (p_{ib,i} - p_i(n)) + \alpha_2 r_2 (p_{gb} - p_i(n)) \right] \quad (4)$$

3.4. Tur Atan Karınca Koloni Algoritması (TACO)

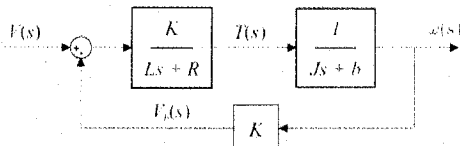
TACO algoritması Hiroyasu ve arkadaşları [7] tarafından geliştirilmiştir. Algoritmada bireyler ikili sayı dizisi ile temsil edilmekte olup her bir yapay karınca dizideki bir elemanın değerinin 1 ya da 0 olup olmadığına karar vermeye çalışır. Karar vermede karıncalar sadece koku maddesi (feromon) bilgisini kullanır. Bir karınca dizideki tüm ikili sayıların değeri için karar verdiğinde problem için bir çözüm üretmiş demektir. Bu çözüm, problem için değerlendirilir ve uygunluk fonksiyonu olarak adlandırılan bir fonksiyon aracılığıyla çözüme ilişkin bir uygunluk değeri elde edilir. Bu değere bağlı olarak, yapışacak bir koku maddesi miktarı ikili sayılar arasında oluşan yapay bütün alt yollara yapıştırılır [8]. Şekil 2 TACO'da bir karıncanın arama mantığı resmedilmiştir. Algoritma hakkında daha kapsamlı bilgi [8]'den elde edilebilir.



Şekil 2. TACO'da karıncanın arama şekli.

4. Kıyaslama Uygulaması

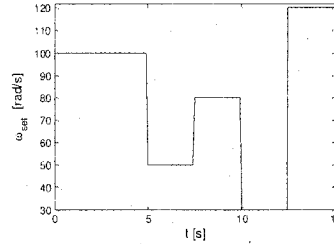
Bu çalışmada önceki bölümde kısaca tanımlanan GA, DE, PSO ve TACO algoritmalarıyla ABC algoritmasının başarım kıyaslaması için DC motoru hız denetimi için PID parametrelerinin (K_p , K_d ve K_i) belirlenmesi problemi kullanılmıştır. Şekil 3'de DC motor modeli öbek şeması verilmiştir. Bu çalışmada, gerçek bir DC motora ait ve Tablo 1'de verilen parametrelerle ayrı zamanda sayısal çözümleme ile sezgisel algoritmaların başarımları sınanmıştır. Algoritmalar koşutururken denetleç için ön ayar değerleri Şekil 4'de verildiği biçimde kullanılmıştır.



Şekil 3-DC motor modeli öbek şeması [9].

Tablo 1-DC motor parametreleri.

Motor gücü	$p=93.3 \text{ kW}$
Motor gerilimi	$v=210 \text{ V}$
Hız	1150 devir/dk
Rotor eylemsizlik	$J=2.83 \text{ kg.m}^2$
Mekanik sistem sürtünmesi	$b=1.89 \text{ Nms}$
Motor sargı endüktansı	$L=1.2 \text{ mH}$
Motor sargı direnci	$R=0.0125 \Omega$
Zıt emk sabiti	$K=1.78 \text{ N/A}$



Şekil 4-Açısal hız denetiminde için kullanılan ön ayar (set) değerleri değişimi.

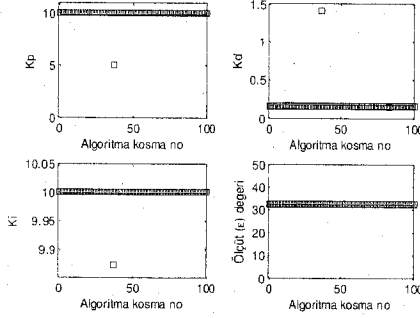
PID parametrelerinin optimizasyonu için sezgisel algoritmalarının bireyleri uygunluk değeri (5)'de verilen ortalama karesel hata (mse) ölçütü kullanılmıştır. Eşitlikte N bireylerin belirlediği değerlerle işleyen PID denetleç denetimi altında sayısal sistem çözümünde kullanılan toplam adım sayısını, ω_{set} motor açısal hız önayar (set) değeri, ω_i ise sistem çözümünde elde edilen açısal hız değerini ve k örnekleme indeksini göstermektedir.

$$e = \frac{1}{N} \sum_{k=1}^N (\omega_{set}(k) - \omega_i(k))^2 \quad (5)$$

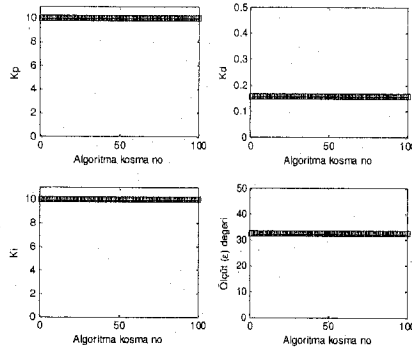
Kullanılan sezgisel algoritmalarda sürü büyüklüğü (birey sayısı) 30, en büyük nesil sayısı olarak 100 alınmıştır. Algoritmalarda kullanılan parametreler sırasıyla; ABC için kaynağı bırakma limit değeri $LC=90$, GA için 18 bit gen gösterimi, mutasyon olasılık 0.5, ve çaprazlama olasılık oranı ise 0.01; DE algoritması için ölçekleme faktörü 0.5 ve çaprazlama olasılık oranı ise 0.7; PSO için $\xi=0.7298$, $c_1=c_2=2.05$, parçacık hızları parametre alt /üst sınır değerlerinin $1/5$ 'i ile sınırlı; TACO için bireyler için 18 bit gösterim, buharlaşma oranı 0.1'dir. PID parametrelerinin sınır değerleri (6)'da verildiği biçimde kullanılmıştır.

$$0 \leq K_p \leq 10, \quad 0 \leq K_d \leq 3, \quad 0 \leq K_i \leq 10 \quad (6)$$

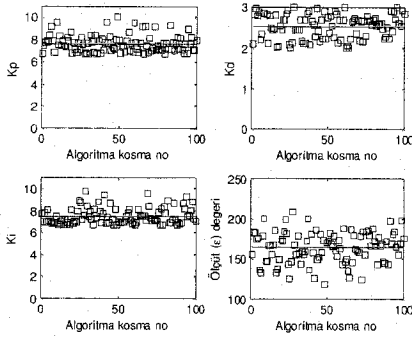
Her bir sezgisel algoritma MATLAB programlama dilinde kodlanarak Intel(R) Core(TM) i7, Q 740 @ 1.73 GHz CPU, 6 GB RAM donanımlı, 64 bit işletim sistemine sahip bir diz üstü bilgisayarda çalıştırılmıştır. Her bir algoritma, bir önceki bölümde tanımlanan problemin çözümünde başarımlarını irdelenek için 100'er kez koşuturulmuştur. Her bir ayrı çalışma sonucu bulunan PID parametreleri ve en son nesilde elde edilen en iyi ölçüt değeri, her bir algoritma için Şekil 5-9'da ayrı ayrı verilmiştir.



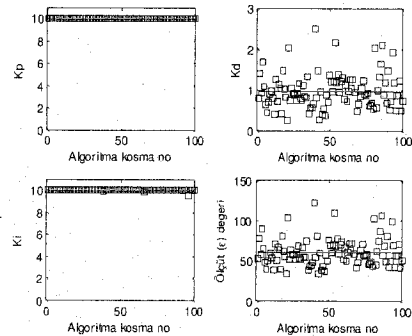
Şekil 5-ABC algoritmasının sonuçları.



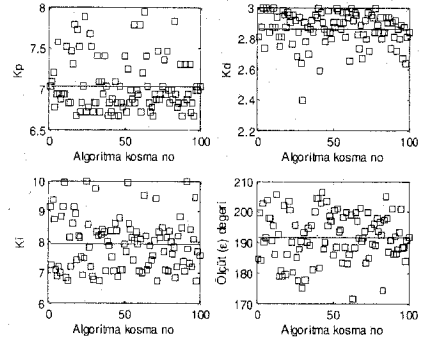
Şekil 6-DE algoritmasının sonuçları.



Şekil 7-GA algoritmasının sonuçları.



Şekil 8-PSO algoritmasının sonuçları.



Şekil 9-TACO algoritmasının sonuçları.

5. Sonuçlar ve Yorumlar

Her bir algoritmanın 100 kez koşurulması sonucu bulunan Şekil 5-9'da verilen sonuçlar Tablo 2'de merkezi eğilim ölçülerine dayalı olarak irdelenmiş ve tabloda her bir algoritma için aritmetik ortalaması verilen PID parametreleri ile denetim başarımı Şekil 10'da verilmiştir. Tablo 2'de toplu olarak verilen istatistikî sonuçlar baz alınarak, en iyi başarıyı gösteren algoritmaya ait veri koyu mavi yazı tipi ile, en kötü başarıyı gösteren algoritmaya ait veri altı çizilmiş kırmızı yazı tipi ile verilmiştir. Bu verilere dayalı olarak, DE algoritmasının nesil başına gerekli koşma zamanı hariç tüm kategorilerde en iyi başarıyı gösterdiği görülmektedir. ABC algoritmasının ise nesil başına gerekli koşma zamanında son derece çarpıcı üstünlüğünün yanı sıra diğer kategorilerde de DE algoritmasının en yakın rakibi olduğu değerlendirilebilir. DE ve ABC'nin ardından başarımların sıralaması olarak PSO, TACO ve GA verilebilir. Ancak, en önemli gösterge olarak görülebilecek nesil başına ortalama koşma zamanı kategorisinde sıralama ABC, DE, PSO, TACO ve GA şeklindedir. Bu sonuca dayalı olarak, gerçek zamanlı uygulamalarda ABC'nin kullanımının diğer dört rakibine göre belirgin olarak elverişli olduğu anlaşılmaktadır.

6. Teşekkür

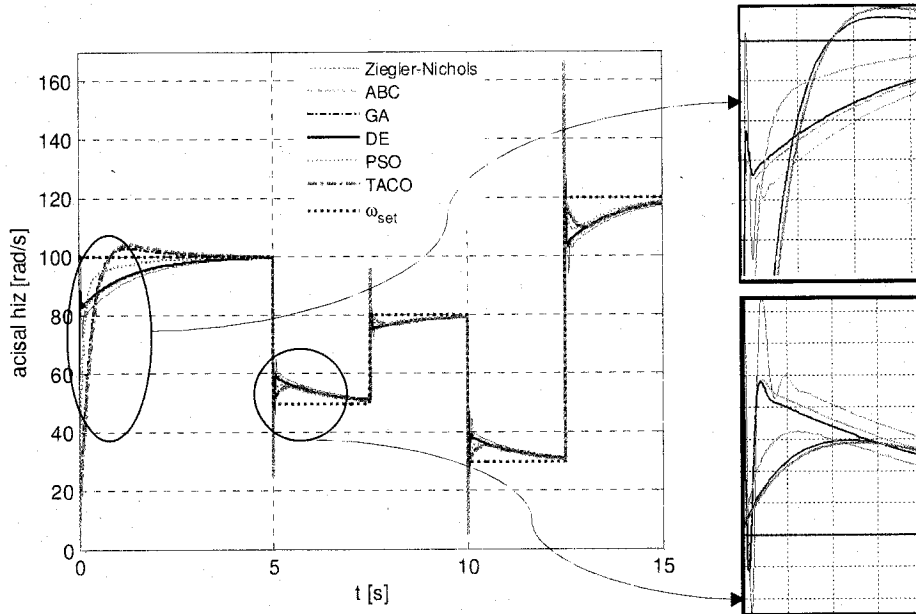
Bu çalışma Bilecik Üniversitesi 2010-02-BİL.01-002 nolu BAP projesi kapsamında desteklenmiştir.

7. Kaynaklar

- [1] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical Report, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

Tablo 2: Algoritmaların 100 kez koşturulma sonucu başarımlarını kıyaslama ölçütleri.

Kategori	Ölçütü	ABC	GA	DE	PSO	TACO
En iyi uygunluk değeri (Son nesilde bulunan)	Aritmetik ortalama	32.451355	164.068177	32.450369	60.3918143	<u>191.952581</u>
	Ort. sapma	0.001619	<u>15.770520</u>	5.46e-14	12.636871	6.510355
	Varyans	0.005299	<u>19.158736</u>	6.59e-14	16.825271	7.883210
Kp Değeri (Son nesilde bulunan)	Aritmetik ortalama	9.949624	7.626984	10	9.999825	7.039683
	Ort. sapma	0.099743	<u>0.588624</u>	6.93e-16	3.47e-04	0.256561
	Varyans	0.501230	<u>0.758087</u>	2.58e-15	0.001742	0.322914
Kd Değeri (Son nesilde bulunan)	Aritmetik ortalama	0.169251	2.521111	0.156927	0.968981	2.863651
	Ort. sapma	0.024792	0.250060	6.24e-09	<u>0.318168</u>	0.083505
	Varyans	0.124600	0.289377	7.49e-09	<u>0.423048</u>	0.108125
Ki Değeri (Son nesilde bulunan)	Aritmetik ortalama	9.998712	7.470370	10	9.9937182	7.941269
	Ort. sapma	0.002549	0.601682	3.91e-16	0.012203	<u>0.696921</u>
	Varyans	0.012811	0.735951	2.15e-15	0.046348	<u>0.849996</u>
1 nesil için gerekli süre [s]	Aritmetik ortalama	0.095165	<u>0.321317</u>	0.198012	0.215582	0.272611

**Şekil 10:** Algoritmalarla bulunan parametreler ile sistem denetim başarımları.

- [2] B. Akay, "Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi Algoritmasının Performans Analizi", Erciyes Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı Doktora Tezi, Kasım 2009.
- [3] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, United States, 1975.
- [4] R. Storn ve K. Price, "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", J. Global Optimization, vol. 11, 1997, s. 341-359.
- [5] B.V. Babu ve M. M. L. Jehan, "Differential Evolution for Multi-objective Optimization", Proceedings of 2003 Congress on Evolutionary Computation (CEC-2003), Canberra, Australia, December 8-12, 2003, s. 2696-2703.
- [6] M. Clerc ve J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space", IEEE Trans. Evol. Comput., vol. 6, no. 1, 2002, pp. 58-73.
- [7] T. Hiroyasu, M. Miki, Y. Ono and Y. Minami, "Ant Colony for Continuous Functions", The Science and Engineering Doshisha University XX(Y), 2000.
- [8] M. Kesler, C. Karakuzu ve U. Yüzgeç, "Güncel Sezgisel Arama Algoritmalarının Denetleyici Parametrelerinin Optimizasyonunda Başarım Kıyaslaması", 6. Uluslararası İleri Teknolojiler Sempozyumu (IATS'11) Bildirileri, Flazığ, Türkiye, 16-18 Mayıs 2011, vol. 4, s. 281-286. (<http://web.firat.edu.tr/iats/cd/subjects/Electrical&Electronics/EAE-59.pdf> linkinde internet ortamında basılmıştır.)
- [9] R. Babuska ve S. Stramigioli, Matlab and Simulink for Modeling and Control, Lecturer Note, Delft University of Technology, Faculty of Information Technology and Systems, Control Laboratory, 1999.